T.C.
MİLLÎ SAVUNMA ÜNİVERSİTESİ
ATATÜRK STRATEJİK ARAŞTIRMALAR VE LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRONİK MÜHENDİSLİĞİ PROGRAMI

İNSANSIZ HAVA ARAÇLARI İÇİN YOL PLANLAMA VE YOL İZLEME ALGORİTMALARI KULLANARAK GÜZERGAH OPTİMİZASYONU

YÜKSEK LİSANS TEZİ

HÜSNÜ UMUT OKUR
4192102

TEZ DANIŞMANI: Dr. Öğr. Üyesi Pınar Öztürk ÖZDEMİR
İKİNCİ TEZ DANIŞMANI: Dr. Musa Nurullah YAZAR

İSTANBUL
AĞUSTOS 2022

T.C.
MİLLÎ SAVUNMA ÜNİVERSİTESİ
ATATÜRK STRATEJİK ARAŞTIRMALAR VE LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRONİK MÜHENDİSLİĞİ PROGRAMI

İNSANSIZ HAVA ARAÇLARI İÇİN YOL PLANLAMA VE YOL İZLEME ALGORİTMALARI KULLANARAK GÜZERGAH OPTİMİZASYONU

YÜKSEK LİSANS TEZİ

HÜSNÜ UMUT OKUR
4192102

TEZ DANIŞMANI: Dr. Öğr. Üyesi Pınar Öztürk ÖZDEMİR
İKİNCİ TEZ DANIŞMANI: Dr. Musa Nurullah YAZAR

İSTANBUL
AĞUSTOS 2022

T.C.
MİLLÎ SAVUNMA ÜNİVERSİTESİ
ATATÜRK STRATEJİK ARAŞTIRMALAR VE LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRONİK MÜHENDİSLİĞİ PROGRAMI

İNSANSIZ HAVA ARAÇLARI İÇİN YOL PLANLAMA VE YOL İZLEME ALGORİTMALARI KULLANARAK GÜZERGAH OPTİMİZASYONU

YÜKSEK LİSANS TEZİ

HÜSNÜ UMUT OKUR
4192102

Tezin Enstitüye Verildiği Tarih : 22.08.2022
Tezin Savunulduğu Tarih : 28.07.2022

Tez Oy birliği ile başarılı bulunmuştur.

Unvan	Ad Soyad	İmza
Tez Danışmanı	Dr. Öğr. Üyesi Pınar Öztürk ÖZDEMİR	
İkinci Tez Danışmanı	Dr. Musa Nurullah YAZAR	
Jüri Üyeleri	Dr. Öğr. Üyesi Mehmet Kürşat YALÇIN	
	Dr. Öğr. Üyesi Sıddık Murat YEŞİLOĞLU	
	Dr. Öğr. Üyesi Deniz ÖZENLİ	

İSTANBUL
AĞUSTOS 2022

ÖZGÜNLÜK RAPORU

Tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler ve ç) Sonuç kısımlarından oluşan toplam 60 sayfalık kısmına ilişkin, 06/06/2022 tarihinde şahsım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan özgünlük raporuna göre, tezimin benzerlik oranı %9'dur.

Uygulanan filtrelemeler:

1- Kaynakça hariç
2- Alıntılar hariç/dâhil
3- 5 kelimeden daha az örtüşme içeren metin kısımları hariç

Millî Savunma Üniversitesi Atatürk Stratejik Araştırmalar ve Lisansüstü Eğitim Enstitüsü Lisansüstü Tez Çalışması Özgünlük Raporu Alınması ve Kullanılması Uygulama Usul ve Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Hüsnü Umut OKUR
22.08.2022

ETİK BEYAN

Millî Savunma Üniversitesi Enstitüleri Dönem Projesi ve Lisansüstü Tez Hazırlama Kılavuzu'nda yer alan kurallarına uygun olarak hazırladığım bu tez çalışmasında; tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi, tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu, tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi, kullanılan verilerde herhangi bir değişiklik yapmadığımı, bu tezde sunduğum çalışmanın özgün olduğunu, bildirir; aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Bu tezdeki düşünce, görüş, varsayım, sav veya tezler bana aittir; Millî Savunma Bakanlığı, Millî Savunma Üniversitesi ve Atatürk Stratejik Araştırmalar ve Lisansüstü Eğitim Enstitüsü sorumlu tutulamaz.

Hüsnü Umut OKUR
22/08/2022

Sevgili aileme ve eşime

ÖNSÖZ ve TEŞEKKÜR

Hayatta karşılaşılan zorluklara karşı geliştirilen çözümlerde rol alan mühendislerin, geliştirilen çözümü de geliştirme gayreti ve merakı içinde olması ile çok büyük bir buluş olarak nitelendirilebilecek olan bir planör uçağın şu anda insansız jet uçakları haline şahitlik ediyoruz. Hiçbir zaman elindeki ile yetinmeyen insanoğlu olarak bu tez çalışmasında en kısa yol planlama algoritması olan RRT algoritmasını nasıl daha verimli hale getirebiliriz düşüncesi ile Pure Pursuit algoritması ile hibrit yapıda kullanarak insansız hava araçları için güzergâh optimizasyonu algoritması geliştirdik. Tez çalışmam süresince hem akademik anlamda hem de günlük yaşamda her zaman vakit ayıran ve ortaya bir ürün koyma noktasında daima hazır olan Dr. Musa Nurullah Yazar'a, desteğini hiçbir zaman esirgemeyen eşim Fatma Betül Okur'a, çalışmalarımı devam ettirebilmem için çalışma alanlarından bolca faydalandığım Fatih Belediyesine ve bana yüksek lisansa başlama ve İnsansız Hava Araçları üzerinde çalışma motivasyonunu kazandıran Selçuk Bayraktar'a teşekkür ederim...

İstanbul; Ağustos 2022 Hüsnü Umut OKUR

İÇİNDEKİLER

Sayfa

Sayfa

TABLOLAR LİSTESİ

Sayfa

ŞEKİLLER LİSTESİ

Sayfa

Sayfa

SEMBOL LİSTESİ

$\boldsymbol{\delta}$: Direksiyon Açısı
d : İleri Bakma Mesafesi
e : Çapraz Yol Hatası
κ : Yolun Eğriliği
R : Yarıçap
θe : Baş Hatası
$\boldsymbol{e_{fa}}$: Yol Hatası

KISALTMALAR

DARPA **:** Defense Advanced Research Projects Agency (İleri Savunma Araştırma Projeleri Ajansı)
İHA **:** İnsansız Hava Aracı
PRM **:** Probabilistic Roadmap (Olasılıksal Yol Haritası)
PPM **:** Pure Pursuit Method (Saf Takip Metodu)
RRT **:** Rapidly Exploring Random Tree (Rastgele Hızlı Ağaç Keşfetme)

ÖZ

İnsansız Hava Araçları İçin Yol Planlama Ve Yol İzleme Algoritmaları Kullanarak Güzergah Optimizasyonu

Hüsnü Umut OKUR

Millî Savunma Üniversitesi, Atatürk Stratejik Araştırmalar ve Lisansüstü Eğitim Enstitüsü

İstanbul, Ağustos 2022

Bu çalışmada otonom araçlara yönelik yol planlama ve yol izleme algoritmasının hibrit olarak kullanımına dayalı bir yöntem önerilmiştir. Ortamın boyutu ve engellere bağlı olarak başlangıç noktasından hedef noktasına bağlanan en uygun yolu bulmak, hesaplama açısından maliyetli ve zaman alıcı olabilmektedir. Yol planlama algoritmaları temelde grid ve örneklemeye dayalı yöntemler olarak iki kategoride ele alınabilir. Grid tabanlı yöntemler küçük boyutlu ortamlarda en uygun yolu daha verimli bir şekilde bulabilmektedir ancak ortamın boyutu ve gridlerin çözünürlüğüne göre algoritmanın verimliliği düşmektedir. Örneklemeye dayalı algoritmalar ise büyük ortamlarda çok daha hızlı bir şekilde uygun yolu bulabilirken rastgele noktalarla oluşturulan yol çok sayıda keskin dönüş içermekte bu da yolun verimliliğini düşürmektedir. Örneklemeye dayalı algoritmanın yüksek iterasyonda kullanılması ile bu keskin dönüşlerin önüne geçilebilmekte ancak bu da yine hesaplama süresi açısından verimliliği düşürmektedir. Bu çalışmada örneklemeye dayalı en çok kullanılan yol planlama algoritmalarından biri olan RRT (Rapidly-Exploring Random Tree) yönteminin düşük iterasyonda kullanılmasıyla başlangıç noktasından hedef noktasına bağlanan uygun bir yolun bulunması ve çok sayıda keskin dönüş içeren bu yolun yine literatür de en çok kullanılan yol izleme algoritmalarından biri olan Pure-Pursuit metodu ile birlikte hibrit bir yapıda kullanılmasıyla düşük iterasyonlarda daha verimli bir yol elde edilmesi amaçlanmıştır. Hibrit yapı, RRT'den oluşturulan yol bilgisi ve engellere göre Pure-Pursuit yol izleme algoritmasının giriş parametresini uyarlayarak RRT ile elde edilen yolun daha uygun olmasını amaçlar. Simülasyonda kullanılan araç İnsansız Hava Aracı (İHA) olarak seçilmesine rağmen, yöntemi her türlü mobil robot için genişletmek ve genelleştirmek oldukça kolaydır. Gerçekleştirilen simülasyon sonuçları önerilen hibrit yapının düşük iterasyonlarda sağladığı yolun verimliliğini göstermektedir.

Anahtar Sözcükler: Yol Planlama, Yol İzleme, RRT, PPM, Engelden Kaçma, İHA

Bilim Kodu : 90526
Sayfa Sayısı : XVİ + 44
Tez Danışmanı : Dr.Öğr.Üyesi Pınar Öztürk Özdemir
İkinci Tez Danışmanı : Dr. Musa Nurullah Yazar

ABSTRACT

For Unmanned Aerial Vehicles, Path Optimization With Path Planning And Tracking Algorithms

Husnu Umut OKUR

National Defense University, Atatürk Strategic Studies And Graduate Institute

Istanbul, August 2022

In this paper, a method based on the hybrid use of path planning and path tracking algorithms is proposed for autonomous vehicles. Finding the optimal path between starting point and destination point can be computationally costly and time-consuming depending on the size of the environment and obstacles positions. The path planning algorithms basically can be classified into two categories: grid-based and sampling-based methods. Grid-based methods can find the optimal path more efficiently in small-sized environments, but the efficiency of the algorithm decreases due to the size of the environment and the increasing resolution of the grids. While sampling-based algorithms can find the sub-optimal path faster for big-sized environments, the path generated with random points contains many sharp turns which reduces the efficiency of the path. These sharp turns can be avoided by using the algorithm at a high number of iterations but in this case the efficiency drops in terms of computation time. In the proposed method, it is aimed to use the RRT path planning method to obtain a sub-optimal path between the starting point and destination point at the minimum number of iterations. Then, this sub-optimal path is improved by using Pure-Pursuit path tracking method in a hybrid structure where the input parameters of Pure-Pursuit method is tuned according to the flatness of the path and obstacle positions. Although the vehicle used in the simulation was chosen as an UAV, it is quite easy to expand and generalize the method for all kinds of mobile robots. The simulation results show the efficiency of the path obtained by the proposed hybrid structure at low iterations.

Keywords : Path Planning, Path Tracking, RRT, PPM, Obstacle Avoidance, UAV

Science Code : 90526

Pages : XVI + 44

Supervisor : Asst. Prof. Pinar Ozturk Ozdemir

The Second Supervisor : Ph.D. Musa Nurullah Yazar

1.GİRİŞ

Günümüzde insansız hava araçları savunma sanayii, lojistik, sinema, tarım vb. birçok alanda kullanılmakta ve insan hayatını kolaylaştırmaktadır. Kapalı ortamda veya açık ortamda otonom olarak kullanılan insansız hava araçları bir noktadan bir noktaya giderken noktalar arası yolu belirleyebilmesi ve bu yolu izlerken engellerden kaçınması gerekmektedir. Bu isterlerin karşılanabilmesi için otonom araçlarda kullanılmak üzere yol planlama ve yol izleme algoritmaları ortaya çıkmıştır. Bu algoritmaların birçoğu DARPA adı verilen Amerika da gerçekleşen bir yarışma da bulunmuştur veya hali hazırda kullanılan algoritmaların bir kısmı geliştirilerek yeni algoritmalar elde edilmiştir. Yol planlama algoritmaları başlangıç ve hedef noktaları verilen bir haritada engellere çarpmadan başlangıç noktasından hedef noktaya giden en kısa yolu bulmayı amaçlamaktadır. Otonom araç bir İHA olarak ele alındığında, yol planlama algoritmasının engellerden kaçınarak en uygun yolu oluşturabilmesi için engellerin konumunu tayin etmesi gerekmektedir, bunun için İHA'ların kamera, radar, lidar vb. sensörlerinden yararlanılır. Kamera ve sensör bilgilerine göre İHA'lar yuvarlanma, yunuslama ve dönme hareketleri ile manevra yaparak engellerden kaçabilmektedir. Yol planlama algoritmaları tarafından bulunan en kısa yolu İHA'lar yol izleme algoritmaları kullanarak izlemektedir. Yol izleme algoritmalarının temelde çalışma prensibi verilen yolu optimize ederek, otonom araca hedefe varana kadar güvenli bir hareket sağlamaktır. Yol izleme algoritması ile planlanan yoldaki rotayı en uygun şekilde izlemeye ve aracın hedefe daha doğru, daha kısa sürede, engellere çarpmadan hareket etmesini sağlamaktadır. Yol izleme algoritmasının kullanımının verdiği katkı ile yol planlama algoritmasının oluşturduğu yolun verimi ve doğruluğu artmaktadır. Yol planlama ve yol izleme algoritmalarının beraber kullanıldığı birçok çalışma bulunmaktadır. Bu örneklerde yol planlama algoritmasından çıkış olarak gelen hedef noktalar yol izleme algoritmasının girişini oluşturmakta ve yol izleme algoritması ise yine açısal hız, çizgisel hız ve dönüş açısı çıktılarını üreterek mobil aracın en doğru yolu izlemesini sağlamaktadır.

1.1.Tezin Amacı

Yol planlama algoritmaları, otonom araçlar için başlangıç noktasından başlayıp hedef noktasına ulaşan en kısa ve güvenli yolu bulmayı amaçlamaktadır. Yol izleme algoritmaları ise haritası bilgisi ve yol bilgisini kullanarak otonom araçların izlediği yolda engellerden kaçarak güvenli bir yol takibini, en hızlı ve verimli şekilde sağlamayı amaçlamaktadır. Yol planlama algoritması olarak RRT algoritması tercih edilmiştir. Bu algoritma yüksek iterasyonlarda daha verimli bir yol bulurken düşük iterasyonlarda bulduğu alt-optimal yolda fazla sayıda keskin dönüş bulunmaktadır. Yol izleme algoritması olan Pure-Pursuit metodu ise verilen yol bilgisi ve haritadaki engel bilgisine göre parametrelerini ayarlayarak izleyeceği yoldaki keskin dönüşleri kırpar, engellerden kaçar ve yolu optimize eder. Bu tezde ise yol planlama algoritması olan RRT algoritması ile Pure-Pursuit yol izleme metodunu hibrit yapıda kullanarak, düşük iterasyonda RRT algoritmasından elde edilen alt-optimal yolun ve harita bilgisinin Pure-Pursuit metodu tarafından kullanımı ile alt-optimal yoldaki keskin dönüşlerin kırpılması, İHA'nın yolu takip ederken engellerden kaçınması ve izlenecek olan alt-optimal yolun daha verimli hale getirilerek optimize edilmesi amaçlanmıştır.

1.2 Literatür Araştırması

Son yıllarda global bir planlayıcı olarak RRT algoritmasını, lokal bir planlayıcı olarakta Pure Pursuit algoritması ve reaktif kontrolü kullanan bir hibrit navigasyon çalışması gerçekleştirilmiştir. Bu kullanımda RRT algoritması ile güvenli bir rota oluşturulmakta ve araç Pure Pursuit metodunu kullanarak rotayı takip etmektedir ancak navigasyon bilinmiyorsa ve ortamda dinamik engeller varsa reaktif kontrol aktif hale geçmektedir. Bu çalışmada her iki hibrit yöntem de denenmiş ve dinamik ve statik ortamlarda kullanılmıştır. Pure Pursuit ve RRT, statik bir ortamda kullanılmış ve Reaktif kontrol ve RRT ise dinamik ortamda kullanılmıştır. Her iki kullanımda da engellere çarpmadan araç başarıyla hedef konuma gitmiştir (Elmokadem, T.; Savkin, A.V. A, 2021).

Son yıllarda global bir planlayıcının A* algoritmasını kullanması ve lokal bir planlayıcı olarak Uyarlanabilir Pencere Yaklaşımı (Adaptive Window Approach) algoritmasının kullanılması ile hibrit bir navigasyon çalışması gerçekleştirilmiştir. A* algoritması, bir mobil robotun mevcut konumundan hedef noktasına giden

güvenli yolu bulur, daha sonra Uyarlanabilir Pencere Yaklaşımı, A* algoritmasının oluşturduğu yoldan anahtar noktaları çıkarır. Anahtar noktalar yolun kıvrımlarıdır, bu nedenle anahtar noktaları kullanan Uyarlanabilir Pencere Yaklaşımı Algoritması engellere çarpmayı önler ve yoldaki keskin kenar sayılarını azaltır. Bu yöntemde araç, anahtar noktalar algoritmasındaki anahtar noktalar engellerden kaçınmak için kabul edilebilir hızı seçene kadar sabit bir hızla hareket eder. Pencere Yaklaşım Algoritması Uyarlanabilir(Adaptif) olmasaydı, araç küçük ölçekli alanlarda engellerden kaçamaz ve ayrıca araç büyük ölçekli alanlarda hareket etmek için en uygun yolu bulamazdı. Bu algoritmada uyarlanabilir(adaptif) olması, büyük ölçekli alanlarda olduğu kadar küçük ölçekli alanlarda da başarılı olmasını sağlar, ayrıca bu yöntem dinamik engeller mevcutken de iyi bir şekilde çalışır (Z. Chen, Y. Zhang, Y. Zhang, Y. Nie, J. Tang and S. Zhu, 2019).

Son yıllarda karmaşık ortamlarda hibrit navigasyon çalışmaları yapılmaktadır. Bu çalışmada, global bir planlayıcı A* algoritmasını ve lokal bir planlayıcı olarak Dinamik Pencere Yaklaşımı (Dynamic Window Approach) algoritması kullanılmıştır.

Bu hibrit yaklaşımda öncelikle A* global planlayıcı tarafından en kısa yol bulunduktan sonra lokal planlayıcıya hedef nokta ve harita verilir, bu bilgiler lokal planlayıcıya girdi olarak verilir, bu girdileri DWA kullanarak aracın çizgisel ve açısal hızını tanımlar. Araç, nesnelere çarpmadan alınan hızlarla hareket eder. Araç, global planlayıcıya mevcut konum bilgilerini verir. DWA, yerel harita üzerindeki statik ve dinamik engellerden kaçınarak ve rotayı takip ederek en uygun yörüngeyi planlar. Yerel planlayıcı ve lokal planlayıcı, yerel hedeflerden yararlanarak beraber çalışırlar.

Lokal planlayıcı bölümünde, lokal yol DWA tarafından tahmin edilir. Lokal yol, arama uzayındaki her hız altında aracın kinematik modeli ile kolayca tahmin edilir. Optimal bir yol seçmek için, optimal yolu tahmin edilen yollar açısından değerlendirilir bunun için amaç fonksiyonu kullanılır. Sonuç olarak, bu hibrit yaklaşım, birçok dinamik ve statik engelden kaçarak yolu izlemeyi ve A* tarafından oluşturulan global yoldaki keskin kenarları azaltmayı başarmıştır (Zhong, X., Tian, J., Hu, H. et al. 2020).

2. YOL İZLEME VE YOL PLANLAMA ALGORİTMALARI

2.1 Yol Planlama Algoritmaları

Yol planlama algoritmalarının kullanım amacı mobil robotlarda, otonom araçlarda ve İHA'lar da belirli bir haritada, bir başlangıç noktasından ulaşılmak istenen hedef noktaya güvenli, hızlı ve kısa sürede götürecek yolu kendine özgü yaklaşım ve yöntemler ile elde etmektir. Yol planlama algoritmaları içerdikleri yöntemlere göre farklı avantaj ve dezavantajlar göstermektedir. Yol planlayıcılar temelde iki kategoriye ayrılmaktadır. Bunlar; Grid tabanlı ve Örneklemeye dayalı algoritmalardır. Grid tabanlı algoritmalar küçük haritalarda daha iyi ve doğru sonuçlar verirken örneklemeye dayalı algoritmalar büyük haritalarda çok daha verimli ve hızlı çalışarak optimale yakın yollar üretmektedir.

2.1.1 Grid Tabanlı Algoritmalar

Grid tabanlı algoritmalarda bir alan üzerine gridler(ızgaralar) yerleştirilir ve en kısa yolu bulmak için grid tabanlı algoritmalar kullanılmaktadır. Her grid'in o gride gitme yoluna bağlı bir maliyet hesabı bulunmaktadır. Bu maliyet hesabına göre en iyi yol planlanmaktadır. Grid tabanlı algoritmalar birçok ticari oyunlarda en kısa yolu bulabilmek için kullanılmaktadır.

2.1.1.1 A* Algoritması

En iyi ilk arama algoritmalarından birisidir. İlk olarak Peter Hart, Nils Nilsson and Bertram Raphael tarafından 1968 yılında yayınlanmıştır (Hart, P. E., Nilsson, N. J., & Raphael, B. 1968). Dijkstra algoritmasına benzer olarak A* algoritması da başlangıç düğümünden hedef düğüme en düşük maliyetli yol ağacını oluşturarak çalışır. A* algoritmasını oluşturan iki temel eleman vardır, bunlar: OPEN listesi ve CLOSE listesidir. OPEN listesi ilk olarak tercih edilmeyen düğümlerin kaydını tutar ve liste, başlangıç düğümü ile başlamaktadır. CLOSE listesi halihazırda keşfedilmiş düğümlerin kaydını tutar ve kaydın tutulduğu liste boş olarak başlamaktadır. En olası yolu araştırırken, düşük olasılıkları olduğu kanıtlanan yolları yok saymaktadır. Sezgisel bir algoritma olan A* algoritması bir düğümden başka bir düğüme geçmek

ve sonuç olarak en kısa yol mesafesini hesaplamak için kullanılan denklem **2**.1'de verilmektedir.

$$f(n) = g(n) + h(n) \quad \textbf{(2.1)}$$

- f(n)= Başlangıç düğümünden geçen yolun toplam tahmini maliyeti
- g(n)= Başlangıç düğümünden mevcut düğüme kadar gelmenin maliyeti
- h(n)= mevcut düğümden hedef düğüme olan maliyeti tahmin eden sezgisel fonksiyondur.

Algoritma her adımda en düşük maliyeti olan düğümü alarak ilerlemektedir. Gidilen düğüm ile komşu düğümlerin değerleri güncellemektedir. Bu sayede seçim yapıldıktan sonra ulaşılan düğümün maliyeti fazla çıktığında bir önceki düğüme dönüp yol daha kısa ise o yolu tercih eder. Hedefe ulaşıldığında ise algoritma sonlanmaktadır.

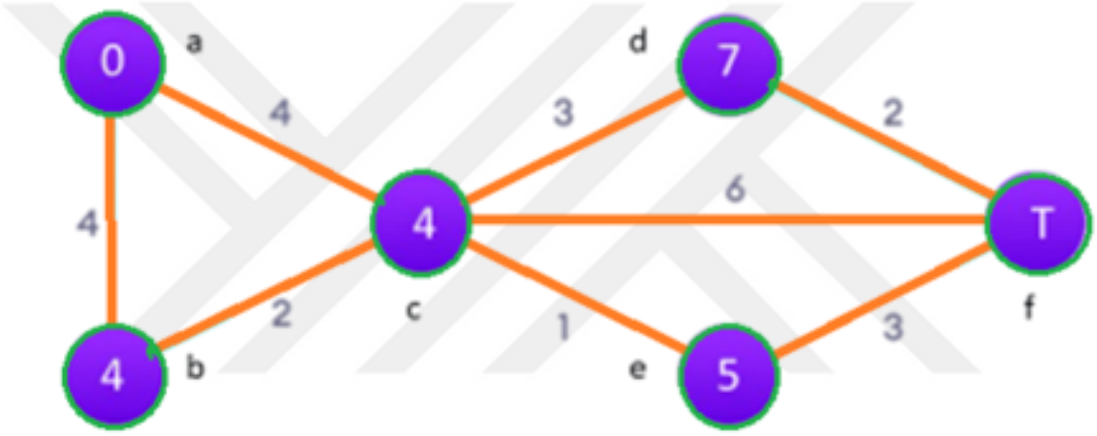
2.1.1.2 D* Algoritması

D* algoritması ilk olarak 1994 yılında Anthony Stentz tarafından tanıtılmıştır (Ferguson, D., & Stentz, A. 2007). D* adı "Dinamik A*" teriminden gelmektedir. Çünkü D* Algoritması çalışırken ark maliyetlerinin değişebilmesi dışında A* gibi davranır. Başlangıçta Dijkstra'nın algoritmasını kullanarak plan yapar ve hızlı yeniden planlama için ara verilerin akıllıca önbelleğe alınmasına olanak tanır. Optimal yolu bulabilmekte ve mutlaka bir en kısa yol oluşturmaktadır. A* algoritmasına göre maliyeti yüksek ve daha karmaşık ortamlarda daha verimli bir yol planlama yapabilmektedir. D* algoritması fonksiyonel olarak A* algoritmasına eşdeğerdir. A* algoritmasından farklı olarak CLOSE listesi ve OPEN listesinin yanında RAISE ve LOWER listelerine sahiptir. Algoritma bir başlangıç düğümü ile başlar ve ardından bitişik düğümün açık olup olmadığını kontrol eder. Bitişik düğüm açıksa h(n), g(n) ve f(n) formülüne göre en kısa yolu günceller ve geçmiş hesaplamaları mevcut düğüm hesaplamalarıyla karşılaştırır. h(n), g(n)'den büyükse, düğüm RAISE listesine eklenir, ancak h(n) g(n)'ye eşitse, düğüm LOWER listesine eklenir. Ardından, algoritma tarafından hedef düğüme ulaşılıp ulaşılmadığını görmek için düğüm karşılaştırılır. Hedef düğüme ulaşmamışsa, bir sonraki bitişik düğümün açık olup olmadığına bakılır. Düğüm açık değilse, CLOSE listesine eklenir ve bir sonraki düğüm kontrol edilir. Hedef düğüme ulaşana kadar düğümleri kontrol edilir.

Hedef düğüme ulaşıldığında, en düşük hesaplamalara sahip düğümler seçilen yolu oluşturur ve algoritma sona erer.

2.1.1.3 Dijkstra Algoritması

Dijkstra algoritması, bilgisayar bilimcisi Edsger W. Dijkstra tarafından 1956 yılında tasarlanmıştır (Dijkstra, E. W. 1959)**.** Dijkstra algoritması (Açgözlü) bir yaklaşım kullanarak en kısa yolu bulmaktadır. Bu yaklaşımı kullanarak her seferinde bir düğümden başka bir düğüme geçerken olası en iyi yerel çözümü göz önüne almaktadır. Algoritma Şekil 2.1 üzerinden anlatılacaktır:



Şekil 2.1: Dijkstra Uygulaması.

Şekil 2.1'de graf üzerinde, başlangıç noktasından, hedef noktaya (T noktasına) giden en kısa yol Dijkstra algoritması kullanılarak bulunmuştur.

Seçili başlangıç düğümü sıfır olarak kabul edilir. Başlangıç düğümünden itibaren T ile gösterilen hedef düğüme giden en kısa yol bulunur. Başlangıç düğümünden çıkan diğer düğümlere bakılır, hangisinin maliyeti daha az ise o düğüm seçilerek ilk yol oluşturulur. Burada iki düğüme gitme maliyetide eşit olduğu için iki düğümde seçilmektedir.

Daha sonra gidilen düğümün bağlantılı olduğu diğer düğümlere bakılır ve en az maliyet ile gidilebilen düğüm seçilir. Burada "a"→"c" düğümüne bakıldığında "4" maliyet iken "b" üzerinden "c" düğümüne gitmek 4+2=6 maliyet olacağından "a" düğümünden "c" düğümüne geçen yol tercih edilir.

Sonraki düğüm için karşımıza 3 seçenek çıkmaktadır. Halhazırda ziyaret edilen yolların maliyetleri güncellenmez. Bu üç seçenekten öncelikle en az maliyeti olan

seçilir ve toplam maliyet hesaplanır. “c” düğümünden “e” düğümü seçilir. 4+1=5. Şimdi ikinci en az maliyetli düğüme hesaplama yapılır. Bu durumda “c” düğümünden “d” düğümü hesaplanır, 4+3=7 maliyet eder. Daha sonra “c” düğümünden hedef düğüm seçilir ve 4+6= 10 hesabından hedef düğüme giden en kısa yol 10 olarak yazılır.

“e” düğümünden hedef düğüme gidilerek toplam maliyet hesaplanır, 5+3=8 maliyet eder ve hedef düğüme giden en kısa yol olan 10’dan küçük bir değer bulunduğu için hedefe giden en kısa yol maliyeti 8 olarak güncellenir. “d” düğümünden hedef düğümün maliyeti ise 7+2=9 maliyet eder. Dolayısıyla bir önceki “e” düğümünden hedefe gidilen toplam maliyet aştığı için bu yol takip edilmez. Bu durumda “e" düğümünden hedefe giden maliyetten daha az bir maliyet ortaya çıkmadığı için hedef düğüme giden yol maliyeti güncellenmez ve 8 olarak bulunmuş olur.

2.1.2 Örneklemeye Dayalı Algoritmalar

Açık alandaki navigasyon problemleri için spesifik yol planlama algoritmaları geliştirilmiştir. Bu algoritmalar genellikle bir maliyet fonksiyonuna göre işlemektedir. Bu yöntemler örneklemeye dayalı yöntemler ile kıyaslandığında grid tabanlı yöntemler, arama teknikleri kullanarak, düşük boyutlu alanları içeren problemler ile sınırlı kalmaktadır. Örneklemeye dayalı yol planlama algoritmaları büyük alanlarda etkisini ispatlamış yöntemlerdir. Bu yöntemler büyük alanlardaki problemleri ele almaktadır, bu yöntemlerin kullanılması ile çarpışmasız alt optimal yollar elde edilir.

2.1.2.1 RRT Algoritması

RRT algoritması 1998 yılında Steven M. LaValle ve James J. Kuffner Jr. tarafından geliştirilmiştir (LaValle, S. M. 1998). RRT algoritması büyük boyutlu alanları çok hızlı bir şekilde aramak için tasarlanmıştır. Büyük boyutlu alanları hızla taramak için rasgele ağaçlar oluşturur ve bu rastgele ağaçları oluşturmayı ise algoritmanın sahip olduğu kurallara göre düğüm atayarak gerçekleştirmektedir. Algoritma her bir adımda öncelikle rastgele bir düğüm seçer daha sonra bu düğümü, kendisine en yakında bulunan, daha önceden hesaplanmış bir düğüm ile bağlantı sağlayarak ağaç oluşturur. Ardından iki düğümü birbirine bağlayan bu yolun ortamda bulunan engellere çarpma durumu kontrol edilir. Yolun engellere bir teması olmadığı takdirde bu düğüm, düğüm ağacına eklenir. Yolun, engeller ile temas etmesi (çarpışması) durumunda, bu yolun bağlanmak için mümkün olmadığına karar verilir ve yeni bir

düğüm seçilerek algoritma tekrar çalıştırılır ve hesaplamalar tekrarlanır, bu durum seçilen yeni düğüm ile oluşturulan yol, herhangi bir engele temas etmeyene kadar devam eder. Bu yöntemle her iterasyonda atanan düğümler ve oluşturulan ağaçlar ile harita taranır ve hedef nokta bulunana kadar devam eder, hedef nokta bulunduğunda veya başlangıçta belirlenmiş olan iterasyon sayısı kadar, algoritma çalıştıktan sonra algoritma sonlanır.

RRT algoritması örneklemeye dayalı algoritmalarda en sık kullanılan algoritmalardan birisidir. En büyük avantajı büyük alanları çok hızlı bir şekilde keşfetmesi ve hedefi bulmasıdır. Algoritma rastgele ağaçlar oluşturarak yollar elde ettiği için hedefi ararken bulduğu yolu en kısa yol olarak garanti edememekte ve aynı parametreler ile aynı haritada çalıştırıldığında her defasında farklı bir yol gösterebilmektedir. Bunlar RRT algoritmasının olumsuz yanlarını oluşturmaktadır.

2.1.2.2 RRT* Algoritması

2011 yılında Sertac Karaman ve Emilio Frazzoli tarafından RRT algoritmasının geliştirilmesi ile RRT* metodu bulunmuştur (Karaman, S., & Frazzoli, E. 2011).

RRT (Rapidly Random Tree), boş alanda rastgele düğümler oluşturarak bir ağaç oluşturur. Başlangıç düğümünden başlar ve hedef düğüme (konuma) ulaşana kadar genişler. Her yinelemede ağaç, yeni bir rastgele düğüm üreterek genişler. Bu düğüm, engel olarak kabul edilen bir bölge içinde değilse, ağaçtan çevresindeki en yakın düğüm aranır. Bu rastgele düğüme, algoritmanın içerisinde belirlenmiş olan maksimum adım büyüklüğü dikkate alınarak en yakın düğümden ulaşılırsa, düğüm ağaca eklenir. Aksi takdirde, bir yönlendirme işlevi kullanarak yeni bir düğüm seçilir, böylece yeni düğümü en yakın düğüme bağlayarak ağaç genişletilir. Yeni düğüm ile en yakın düğüm arasında bağlantı sağlamak için her yinelemede yapılan bağlantı ile oluşturulacak ağacın, haritada bulunan engellere çarpma durumu kontrol edilir.

RRT*'nin temel ilkesi, RRT ile aynıdır, ancak algoritmaya yapılan iki önemli ekleme, önemli ölçüde farklı sonuçlar vermektedir. İlk özellik, r yarıçaplı bir daireyi dikkate alarak yeni düğüm için en iyi ana (root) düğümü bulur. Düğümler birleştirilerek ağaç yapısı oluşturulmadan önce bu dairenin içindeki en düşük maliyetli ana (root) düğümü control eder. En düşük maliyetli komşuyu yeni düğüme bağlar. Ancak ikinci özellik, ağacı minimum maliyetli yollarla korumak için ağacı aynı dairenin içinde yeniden düzenler. Bunun sonucu olarak RRT* algoritmasında

RRT algoritmasına göre daha farklı, yelpaze şeklinde bir ağaç yapısı gözlemlenmektedir. RRT* son derecede düz yollar oluşturmaktadır. Ek olarak, oluşturulan grafikler karakteristik olarak RRT'den farklıdır. Özellikle engellerin fazla bulunduğunda bir alanda en uygun yolu bulmak için RRT*'nin yapısı oldukça faydalıdır.

2.1.2.3 PRM (Probabilistic Roadmap)

Yeni bir düğüm oluşturulduktan sonra, PRM algoritması düğümleri bağlı bileşenlere kümeler. Düğümün ait olduğu bağlı bileşenin saklanması gerekir. Kümeleme yapmak için, rastgele oluşturulmuş düğümün sabit bir yarıçapındaki tüm düğümler toplanır. Bu komşu düğümler, artan mesafe (veya istenen metrik) ile sıralanır. Sıralı komşu düğümler arasında döngü yaparak, rastgele oluşturulan düğümün incelenen düğümle aynı kümede olup olmadığı kontrol edilir. Koşul sağlanırsa yeni düğüm kümeye eklenir. Bir düğüm birden çok kümeye ait olabilmektedir. Yarıçapın boyutu kullanıcı tarafından verilir ve verilen yarıçap değeri yol haritasının yapısını tanımlar. Oluşan kümelerin bazıları bu yarıçapa sahip dairesel bölgelerden olacaktır. Seçilen yarıçap, oluşturulan yol haritasının hızını ve performansını da etkiler. Daha büyük bir yarıçap, daha fazla komşunun ayrıştırılmasını ve küme ilişkilerinin belirlenmesini içermektedir. Kullanıcının belirlediği diğer bir parametre ise düğüm sayısıdır. PRM algoritması, istenen sayıda düğüm üretildiğinde sona erer. Çok fazla düğüm oluşturmak, yol haritasını geliştirmek için avantaj sağlarken hareketin tamamlanması için gereken süreyi geciktirerek bir dezavantaj oluşturmaktadır. Fazla düğüm oluştuğunda komşu bölgelerin incelenmesine daha fazla zaman ayrılacaktır. Düğüm sayısının çok az olması ise kırık bir grafiğe sebep olabilmektedir. Kümeler, özellikle engellerin yoğun olduğu bölgelerde, yol haritasının geri kalanından koparak, kopuk yollar oluşabilmektedir. Az sayıda düğüm ile oluşturulmuş bir yol haritasında oluşturulan yollar da daha düzensiz olmaktadır. Daha az düğüm sayısı ile algoritma çalıştığında, olası rotaların sayısı daha az olmaktadır. PRM algoritmasının dezavantajı, diğer örneklemeye dayalı algoritmalarda da olduğu üzere en uygun kısa yolu vermemesidir. Ayrıca çok fazla engel olan haritalarda engeller arasına düğüm atamayabilir bu yüzden doğru yolu bulması oldukça zorlaşmaktadır fakat bu dezavantaj düğüm sayısı artırılarak giderilebilmektedir. Algoritmanın avantajı ise büyük alanlarda çok verimli çalışması ve birçok çalışma alanı için kullanışlı olmasıdır.

2.2 Yol izleme Algoritmaları

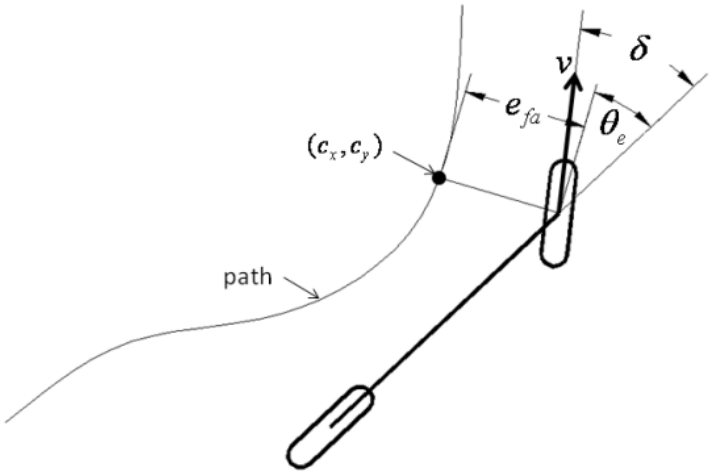
Otonom sürüşte çok önemli bir yere sahip olan yol izleme algoritmaları giriş bilgisi olarak verilen harita ve yol bilgilerini kullanarak engellere çarpmadan en kısa yolu en güvenli şekilde takip ederek hedefe ulaşmayı amaçlamaktadır. Yol izleme algoritmaları otonom aracın başlangıç noktasından hedef noktasına giderken sürüş güvenliğini sağlamaktadır. Yol izleme yöntemleri; Geometrik Algoritmalar, Kinematik Algoritmalar ve Dinamik Algoritmalar olarak üçe ayrılmaktadır. Bilinen yol izleme yöntemleri arasında, geometrik yol izleme algoritmaları, otonom araçlar için en popüler olanıdır. Geometrik yol izleme algoritmaları için Stanley Metodu, Pure Pursuit algoritması, Havucu takip etme algoritmasıdır.

2.2.1 Geometrik algoritmalar

Yol izleme yöntemlerinden en sık kullanılanı geometrik algoritmalardır. Bu yöntemler, araç ve yol arasındaki geometrik ilişkilerden yararlanarak yol izleme problemine kontrol kanunu çözümleri sunar. Bu teknikler genellikle aracın önündeki hatayı ölçmek için ileri bakma mesafesini (lookahead distance) kullanır ve basit dairesel yay hesaplamalarından, vida teorisini içeren daha karmaşık hesaplamalara kadar uzanabilir. Bu başlık altında, bu yöntemlerle en yaygın olarak kullanılan yöntemlerden ikisini açıklayacaktır: Pure Pursuit ve Stanley Metodu.

2.1.2.1 Stanley Metodu

Stanley Metodu 2004 yılında yapılan bir otonom araç yarışması olan DARPA yarışmasında Stanford Üniversitesinin geliştirmiş olduğu bir metottur. Birçok yönü ile Stanley Metodu Pure Pursuit algoritmasına benzemektedir. Stanley Metodunu Pure Pursuitten ayıran özellik, bu metodun referans konumu olarak bir mobil aracın ön akslarını(bir hava aracı için bu konum, daha farklı olmaktadır) kullanmasıdır. Stanley Metodu hem yol hatasına hem de çapraz yol hatasına bakar. Bu yöntemde çapraz yol hatası, aracın ön aksı ile yol üzerindeki en yakın nokta arasındaki mesafe olarak tanımlanmaktadır. Stanley yöntemi, bir mobil araç üzerinden anlatılacak olursa; **Şekil 2.2**'de görebileceğiniz gibi, θe yörünge yönü ile araç yönü arasındaki açıdır. Direksiyon açısı (Steering angle) δ olarak gösterilir.



Şekil 2.2: Stanley Metodu Gösterimi.

Stanley yönteminin üç sezgisel yönlendirme yasası vardır (Yan Ding 2020); Bunlardan ilki baş hatasını gidermek için kullanılan yasadır, formül **2.2**'de gösterilmiştir :

$$\boldsymbol{\delta = \theta e} \tag{2.2}$$

İkinci olarak çapraz yol hatasını gidermek için kullanılan yasadır, çapraz yol hatasını gidermek için araç ile yol arasındaki en yakın noktanın bulunması gerekmektedir. Bunun için kullanılan formül **2.3**'te gösterilmiştir:

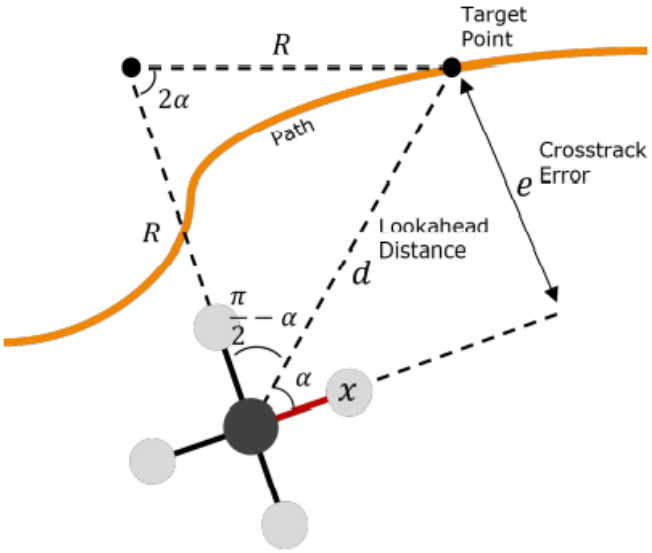
$$\boldsymbol{\delta(t) = tan^{-1}(\frac{e_{fa}(t)}{v_x(t)})} \tag{2.3}$$

Son olarak ise direksiyon açsının en fazla alabileceği değer sınırıdır, formül **2.4**'te gösterilmiştir:

$$\boldsymbol{\delta(t) = \theta e(t) + tan^{-1}(\frac{k e_{fa}(t)}{v_x(t)})} \tag{2.4}$$

2.2.1.2 Pure Pursuit Metodu

Pure Pursuit metodu (PPM), klasik ve en sık kullanılan bir Geometrik yol izleme algoritmasıdır. Bu algoritmada ileri bakma mesafesi, PPM'nin uygun yol izleme performansını elde etmesi için kritik bir parametredir.



Şekil 2.3: Pure Pursuit Metodu Gösterimi.

Şekil 2.3'te pure pursuit metodu kavramını göstermektedir. İleriye bakma mesafesi, yol üzerinde aracın bir miktar önünde olan bir hedef pozisyonunu belirlemek için kullanılır ve algoritma, yolun eğriliğini (κ) hesaplar ve dairesel yörüngeyi takip ederek çapraz yol hatasını **e** en aza indirir. Şekil 2.3'te sinüs yasasını kullanarak, yörüngenin eğriliğini (2.5)'teki gibi elde etmek oldukça kolaydır:

$$\frac{d}{sin2\alpha} = \frac{R}{\sin(\frac{\pi}{2}-\alpha)}$$

$$\frac{d}{2sin2\alpha cos\alpha} = \frac{R}{\cos(\alpha)}$$

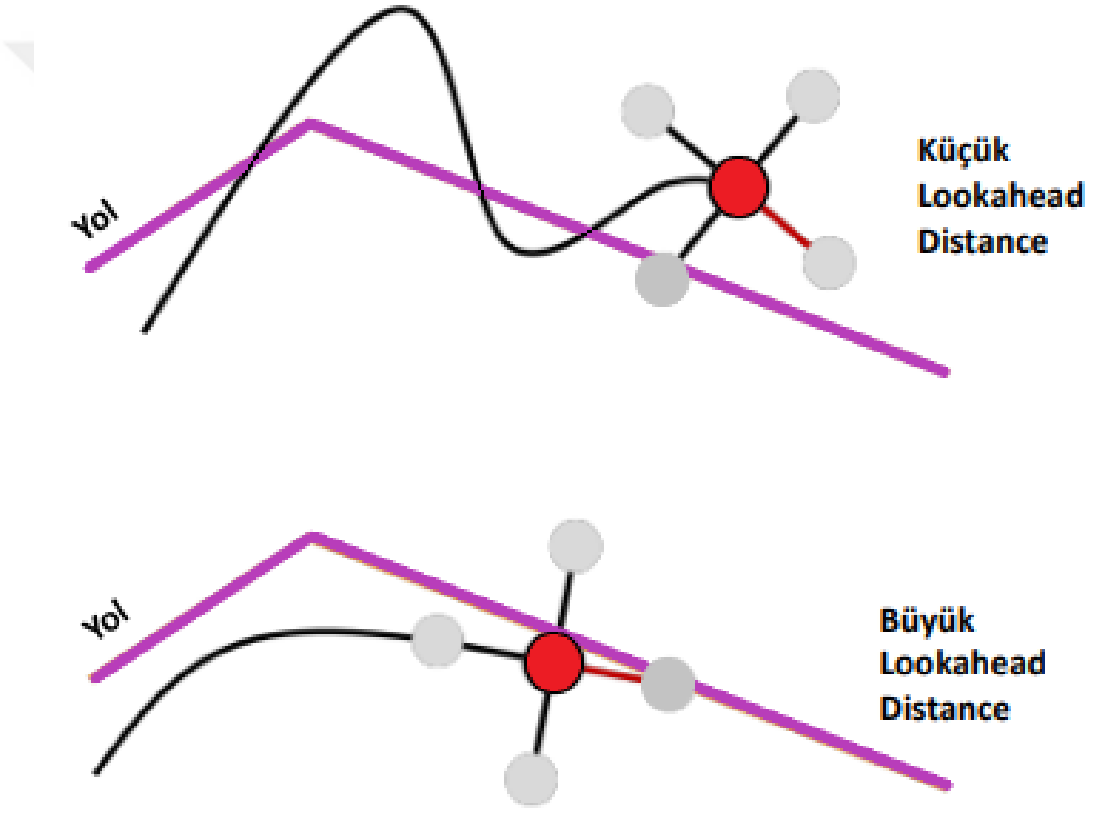
$$\frac{d}{sin\alpha} = 2R$$

$$\kappa = \frac{1}{R} = \frac{2sin\alpha}{d}$$

(2.5)

İleri bakma mesafesi parametresi, pure pursuit metodunun ayarlanarak algoritmanın çalışmasına etki eden ana parametredir. İleriye bakma mesafesi parametresi düşük olarak girildiğinde, aracın istenen yol boyunca aşırı hareket etmesine ve salınıma girmesine neden olan daha agrasif manevralara yol açmaktadır. İleri görüş mesafesi

parametresi daha büyük olarak girildiğinde, daha yumuşak manevralara yol açar ve aşmaları ve salınımları azaltır, ancak köşelerde daha büyük eğriliklere ve çapraz yol hatasına neden olur. Bu sebeple ileri bakma mesafesi parametresi pure pursuit algoritması için çok büyük bir önem taşımakta olup parametre değeri fazla ya da az olarak girildiğinde otonom aracın yolu uygun bir şekilde izlemesini oldukça zorlaştırmaktadır. Şekil 2.4'te (Coulter, R.C. 1992) hem daha küçük hem de daha büyük ileri bakma mesafesinin izleme performanslarını göstermektedir. İzleme algoritmasının performansını iyileştirmek için ileriye bakma mesafesi parametresinin adaptif olarak değiştirilmesi gerektiği açıkça görülmektedir. Son yıllarda ileri bakma mesafesinin adaptif olarak değiştirildiği birçok çalışma ve makale yayınlanmıştır.



Şekil 2.4: Farklı İleri Yol Mesafesi Parametresi ile Yol İzleme Sonuçları.

2.2.3 Kinematik Model

Kinematik model, bisiklet modeli gibi farklı modeller için kinematik hareket denklemlerini sunmaktadır. Ek olarak, hareket denklemlerini yeniden formüle etmek ve yol izleme problemine bir çözüm sağlamak için kontrol teorisinden önemli bir yöntem uygulanır. Bu teorinin uygulanması, kontrolörlerin tasarımı ve kararlılık analizi için iyi bilinen kontrol teorisi araçlarını kullanarak sağlanmaktadır.

Hazırlanan tez de İHA'nın bir kinematik modeli oluşturulmuş ve bu kinematik modelden faydalanılarak tanımlanan yol izlenmiştir. İnsansız hava aracının yol izlerken sahip olduğu kontrolör hava aracına baş açısını düzelterek yolu doğru yönde takip etmesine olanak sağlamıştır.

Kinematik model ile yol izleme takibi yapıldığında; keskin kenarları yumuşatma kinematik model tarafından yapılmamaktadır, otonom araç yüksek hızlara çıktığında yol takibinin verimi oldukça düşmektedir, fazla dönüş içeren yollarda yolu kaçırma ihtimali çok fazladır. (Snider, J. M. 2009)

2.2.4 Dinamik Model

Daha önce açıklanmış olan geometrik model ve kinematik modelde araç dinamiklerinin ihmal edilmesinin sebebi, otonom araçların hızı arttıkça ve yol eğriliği değiştikçe izleme performansının olumsuz olarak etkilenmesidir. Bir mobil aracın dinamikleri çok karmaşık olabilmektedir ve yüksek doğruluktaki modeller çok doğrusal değildir, süreksizdir ve hesaplama açısından maliyetlidir. Dinamik model, otonom araca dinamik etkiler ile yaklaşan ve doğrusal yol izleme kontrolörlerinin tasarımını sağlayan bir yanal dinamik model türetmektedir. Dinamik model kendi içerisinde farklı kontrolcüler kullanarak ve bu kontrolcüleri devamlı ayarlayarak çalışmaktadır.

3. RRT ALGORİTMASININ KULLANIM ALANI VE PERFORMANSI

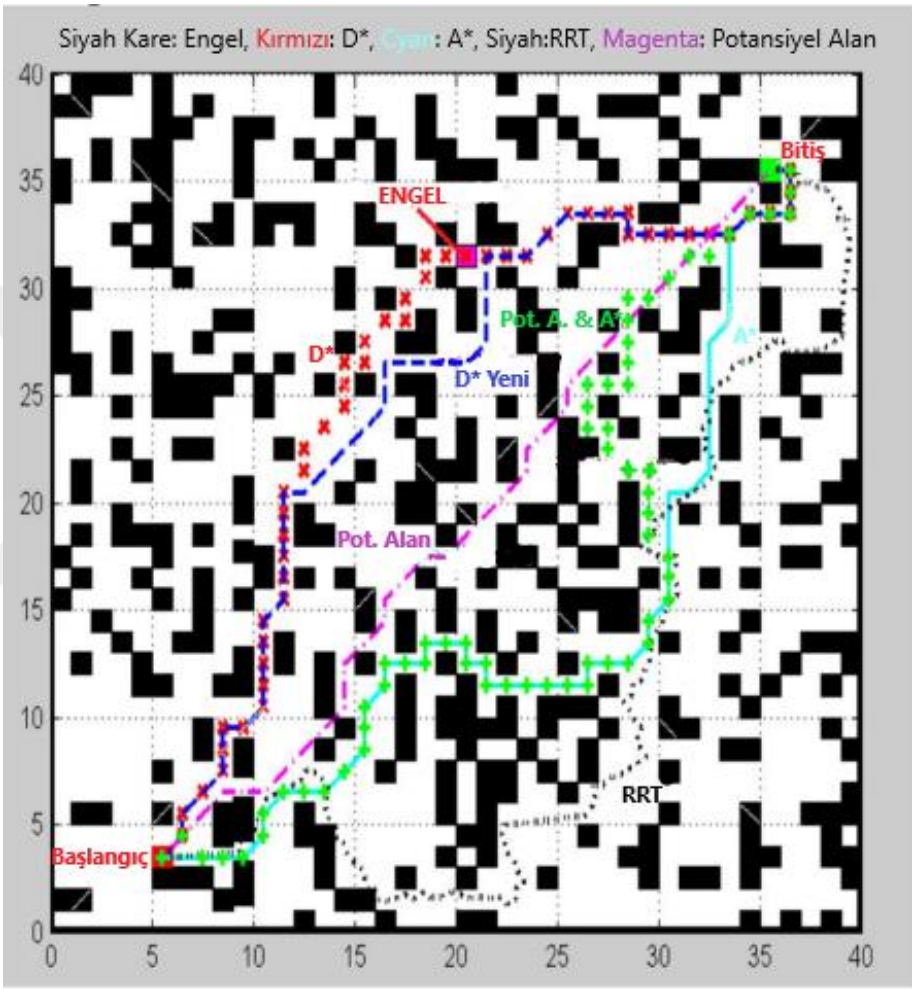
3.1 RRT Algoritmasının Kullanım Alanları

Bu tezde global planlayıcı olarak RRT algoritması kullanılmıştır. RRT algoritması çok eskiden bulunmuş bir algoritma olduğundan günümüze kadar birçok farklı uygulamada performansı test edilmiş ve kendisini ispatlamış bir örneklemeye dayalı yol planlama algoritmasıdır. Algoritma çok kısa sürede geniş ortamları hızlı bir şekilde keşfedebilme yeteneğine sahip olduğundan, büyük alana sahip ortamlardaki yol planlama problemlerini çözmek adına tercih edilmektedir. RRT algoritmasının kullanıldığı çalışmalar incelendiğinde, İHA'lar da mobil yer araçlarında, insansız deniz araçlarında kullanılarak çeşitli görevlerde kullanılmıştır ve kullanım alanının daha büyük alana da genişletilebildiği görülmüştür (Elmokadem, T., & Savkin, A. V. 2021). RRT algoritması kullanıldığı otonom araca ve otonom araç üzerinde kullanılan sensörlere göre, aracın sahip olduğu sensörlerden gelen veriler ile ortama dair engel ve harita bilgilerini elde edilir. Elde edilen engel ve harita bilgileri kullanılarak algoritma içerdiği kurallara göre hızlı bir şekilde rastgele ağaçlar atar ve algoritma iterasyon sayısı tamamlandığında veya hedefe ulaşıldığında sonlanır. Eğer girilen iterasyon sayısında hedef bulunmuş ise oluşturulan ağaçlar üzerinden hedefe giden en kısa yol seçilir. Elde edilen en kısa yol, daha sonra otonom aracın yolu takip etmesi için kullanılır. Otonom araç algoritmanın çok hızlı bir şekilde en kısa yolu elde etmesi ile kısa bir süre içerisinde en kısa yol bilgisini elde etmiş olur, araç bu yolun konumlarını ve harita bilgisini kullanarak yolu izler ve hedef noktaya ulaşır.

Klasik yöntemler ve sezgisel yöntemleri kullanan algoritmalar küçük alana sahip ortamlarda başarılı sonuçlar vermektedirler. Bu yöntemler kullanılırken, haritada belirlenen başlangıç noktasından hedef noktasına giden en kısa yolu bulma işlemi fazla zaman almakta ve işlemcide fazla hafıza kullanmaktadır. Önceki bölümlerde de bahsedildiği üzere RRT algoritması çok kısa bir sürede başlangıç noktasından hedef noktayı tespit etmektir. Bu özelliği diğer algoritmalara göre RRT algoritmasını öne çıkarmakta ve büyük alanlarda kullanım da, problemler için en önemli algoritmalardan birisi olarak gösterilmektedir.

3.2 RRT Algoritmasının Diğer Algoritmalara Göre Karşılaştırılması

Ortak bir harita kullanılarak yapılan bir çalışmada D*, A*, RRT ve Potansiyel Alan metodunun başlangıç noktasından hedef noktasına ulaşıncaya kadar geçen süre, algoritmanın kullandığı işlemci boyutu hesaplanmıştır. Bu sayede algoritmaların performanslarının analizi yapılmıştır. Yapılan çalışmada algoritmaların çıktıları Şekil 3.1'de gösterilmiştir (Gopikrishnan. S, B. V.S. Shravan 2011).



Şekil 3.1: Farklı Yol Planlama Algoritmalarının Karşılaştırılması.

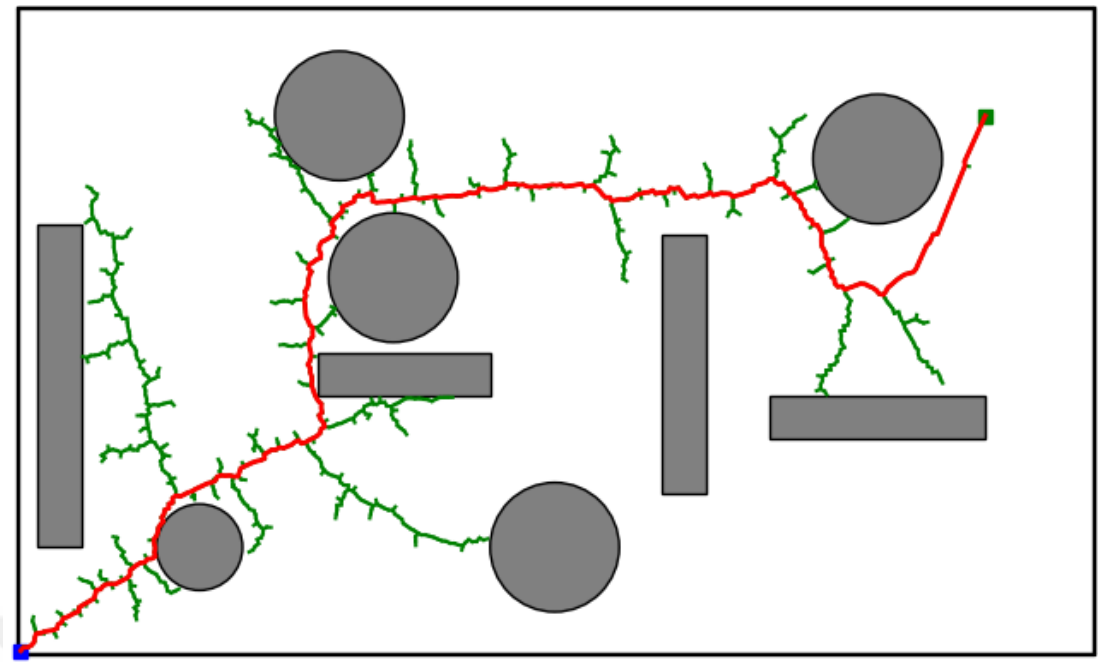
Tablo 3.1: Yol Planlama Algoritmaları Simülasyon Sonuçları.

Algoritma Adı	Tamamlanma Süresi(sn)	Algoritma Boyutu(Byte)
A*	6.3	9.459
D*	6	17.738
RRT	1	9.425
Potansiyel Alan Metodu	0.2	3.703

Tablo 3.1'deki simülasyon sonucundan algoritmaların izledikleri yolların uzunluk ve keskin kenar yoğunluğunu göz önünde bulundurduğumuzda, RRT algoritması diğer algoritmalara göre daha verimsiz bir sonuç üretimi gözükmektedir. Tablo 3.1'e baktığımızda ise RRT algoritmasının klasik ve sezgisel yöntemler kullanan algoritmalara kıyasla çok daha kısa sürede ve daha az işlemci belleği kullanarak hedef noktaya ulaştığı görülmektedir. Potansiyel alan metodu, sonuçlar Tablo 3.1'de ki veriler göz önünde bulundurulduğunda simülasyonda kullanılan tüm algoritmalara göre daha hızlı bir şekilde hedefe ulaşmış ve daha az işlemci belleği kullanılmış gözükmektedir ancak simülasyon sonucu dikkatli bir şekilde incelendiğinde aslında Potansiyel Alan Metodunun engellere çarpmadan bir yol izleyemediği dolayısıyla başarılı bir sonuç çıkaramadığı görülmektedir. Tüm sonuçlar dikkatli incelendiğinde RRT algoritması diğer algoritmalara göre zaman ve maliyet açısından bakıldığında daha verimli çalışmaktadır.

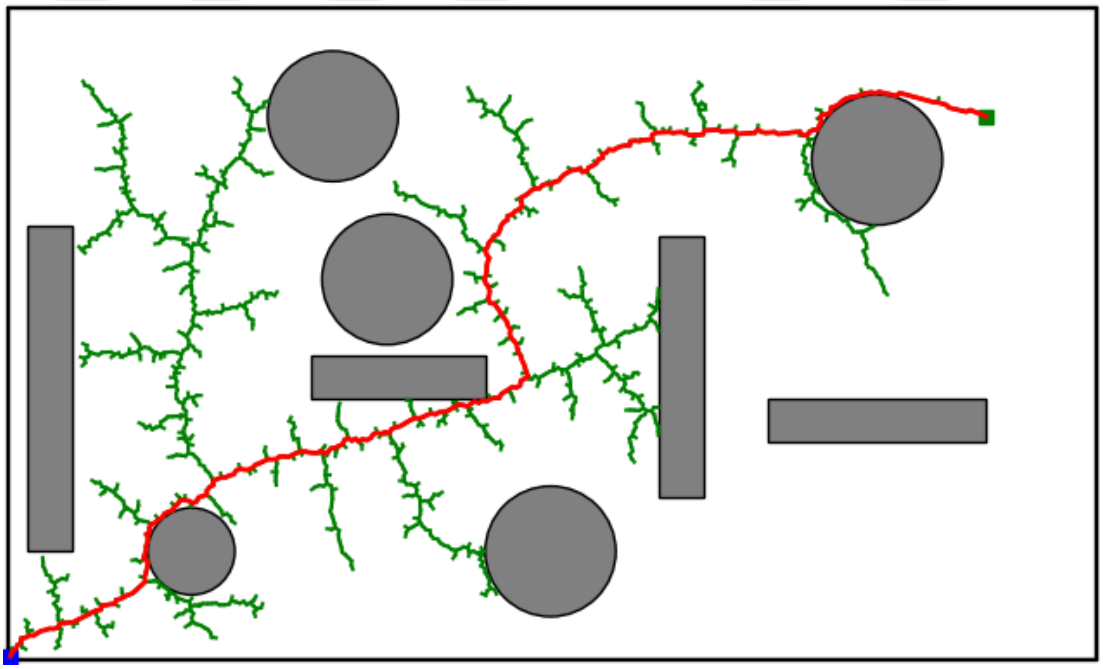
3.3 RRT Algoritmasının Performans Analizi

RRT algoritması, başlangıç ve hedef nokta koordinatlarının verildiği, 2 boyutta 300m x 500m'lik bir alanda, çeşitli engellerin olduğu bir haritada simüle edilmiştir. Bu simülasyon da amaç RRT algoritmasını iterasyon sayısı parametresinin, algoritmanın bulduğu en kısa yolun uzunluğuna, yoldaki keskin kenarlara ve simülasyonun tamamlanma süresine etkisine bağlı olarak performansı gözlemlenecektir.



Şekil 3.2: 2500 İterasyonda RRT Algoritmasının Çıktısı.

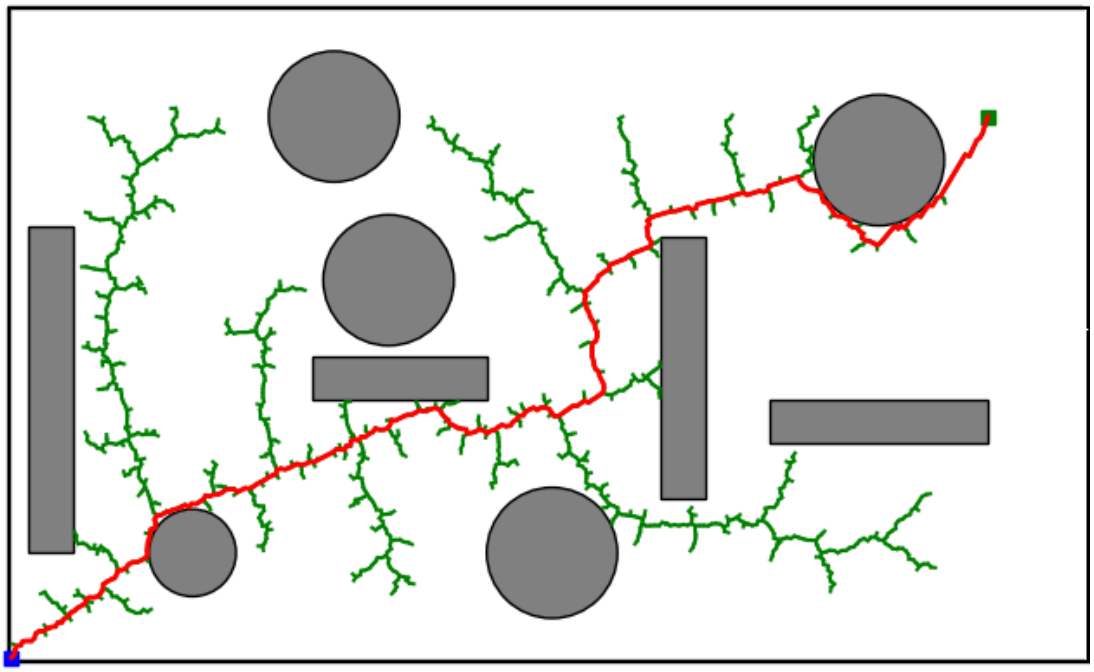
Şekil 3.2'de RRT algoritmasının iterasyon sayısı 2500 olarak ayarlanmıştır. Bu simülasyon da algoritmanın başlangıç noktasından hedef noktasını bulması 9.99 saniye sürmüştür. Kırmızı ile gösterilen yolun uzunluğu ise 183.4 metre olarak hesaplanmıştır.



Şekil 3.3: 5000 İterasyonda RRT Algoritmasının Çıktısı.

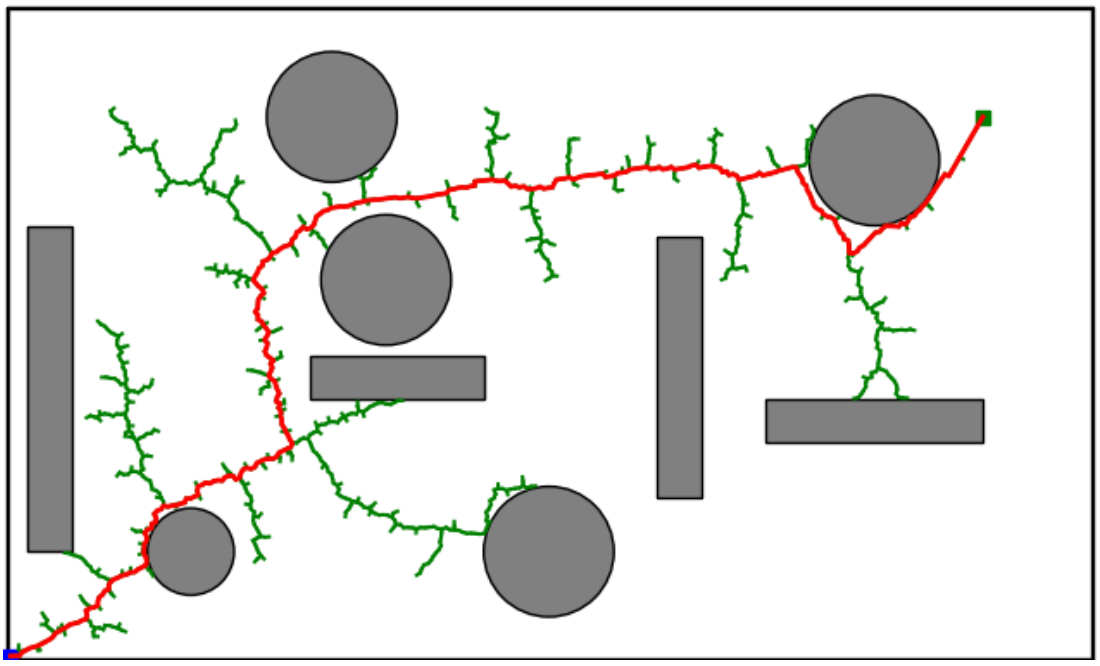
Şekil 3.3'te RRT algoritmasının iterasyon sayısı 5000 olarak ayarlanmıştır. Bu simülasyon da algoritmanın başlangıç noktasından hedef noktasını bulması 16.12

saniye sürmüştür. Kırmızı ile gösterilen yolun uzunluğu ise 194.6 metre olarak hesaplanmıştır.



Şekil 3.4: 10000 İterasyonda RRT Algoritmasının Çıktısı.

Şekil 3.4'te RRT algoritmasının iterasyon sayısı 1000 olarak ayarlanmıştır. Bu simülasyon da algoritmanın başlangıç noktasından hedef noktasını bulması 14.52 saniye sürmüştür. Kırmızı ile gösterilen yolun uzunluğu ise 189.2 metre olarak hesaplanmıştır.



Şekil 3.5: 15000 İterasyonda RRT Algoritmasının Çıktısı.

Şekil 3.5'te RRT algoritmasının iterasyon sayısı 15000 olarak ayarlanmıştır. Bu simülasyon da algoritmanın başlangıç noktasından hedef noktasını bulması 10.96 saniye sürmüştür. Kırmızı ile gösterilen yolun uzunluğu ise 184.5 metre olarak hesaplanmıştır.

3.3.1 Performans Analizi Sonuçları

Bölüm 3.3'te 4 farklı iterasyon değeri ayarlanmış RRT algoritmasının, sonuçları görülmektedir. Sonuçlara bakıldığında iterasyon sayısının fazla olması hedef noktanın bulunması için geçen süreyi her zaman düşürmemektedir. İterasyon sayısının fazla olduğu sonuçlarda keskin kenarlar iterasyon sayısının az olduğu diğer sonuçlara göre daha azdır. Algoritma çalıştırıldıktan sonra hedefi bulana kadar geçen süre iterasyon sayısı aynı olduğunda bile farklı çıkabilmektedir. Bu durum RRT algoritmasının rastgele ağaçlar oluşturarak bir yol elde etmesinden kaynaklanmaktadır. RRT algoritması diğer algoritmalara göre daha hızlıdır ve algoritma kendi içerisinde aynı harita boyutunda farklı iterasyonlarda birbirine yakın sürelerde sonuca ulaşır. Bu sebeple aynı ortamda iki RRT algoritması karşılaştırıldığında, oluşturulan yolun uzunluğu ve yol üzerindeki keskin kenar yoğunluğuna bakarak bir performans karşılaştırması yapılabilir. Bu şekilde bir performans karşılaştırılması yapıldığında genelde iterasyon sayısı yüksek olan uygulamada oluşturulan yol daha kısa ve keskin kenarlar daha az olmaktadır. Bu şekilde bir performans analizi yapıldığında genelde iterasyon sayısı yüksek değer verildiğinde daha verimli sonuçlar elde edilmiştir.

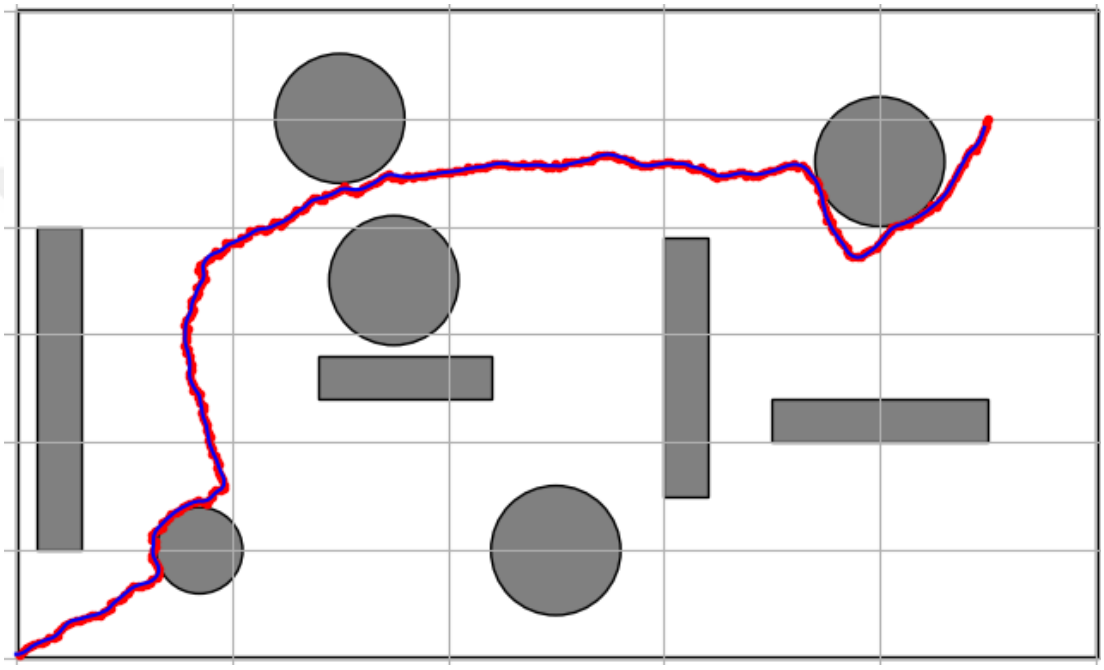
4. PURE PURSUIT METODUNUN KULLANIM ALANI VE PERFORMANSI

4.1 Pure Pursuit Metodunun Kullanım Alanları

Yol izleme algoritmaları otonom araçların planlanmış olan yolu doğru bir şekilde izlemesi için kullanılmaktadır. Birçok yol izleme yöntemi bulunmaktadır. Her yöntem kendine ait kurallara göre aracı yönlendirerek güvenli bir yol takibi sağlar ve aracı hedef konuma götürmeyi amaçlar. Bu yöntemlerden kararlılığı ve kolaylığı ile bilinen Pure Pursuit algoritması ilk olarak 1980'lerde Wallace tarafından keşfedilmiştir (Coulter, R.C. 1992). Pure Pursuit algoritması günümüzde otonom otomobillerde, mobil kara araçlarında, İHA'lar da ve diğer çeşitli platformlarda kullanılmaktadır. Pure Pursuit metodu güvenli bir takip sağlayabilmek için, girdi olarak verilen harita ve yol bilgisini kullanmaktadır. Elde ettiği bu bilgileri algoritma içinde gerçekleşen hesaplamaları yaparak otonom aracı bulunduğu konumdan önünde bulunan hedef konuma araca bir eğri çizdirerek gerçekleştirmektedir. Bu eğriyi oluşturup, otonom araç çizilen eğriyi takip ederken engellerden de kaçmaktadır. Pure Pursuit'in oluşturmuş olduğu iyileştirilmiş yol otonom araç tarafından kullanarak, güvenli bir hareket sağlamaktadır ayrıca iyileştirilmiş yol Pure Pursuit metodunun kullanımı ile eski haline göre daha kısa hale gelmekte ve yolun sahip olduğu keskin kenarları da çizilen eğriler sayesinde azalma göstermektedir. Önceki bölümlerde de bahsedildiği üzere Pure Pursuit algoritmasının en önemli parametresi ileri bakma mesafesi parametresidir. Bu parametre doğrudan metodun otonom araç üzerinde performansı etkilemektedir. Parametrenin yüksek bir değere ayarlanması, otonom aracın izlemesi gereken noktaları es geçmesine ve aracın yoldan çıkmasına sebep olabilmektedir. Parametrenin düşük bir değere ayarlanması ise otonom aracı salınıma sokabilmekte ve nihayetinde yoldan sapmasına sebebiyet verebilmektedir.

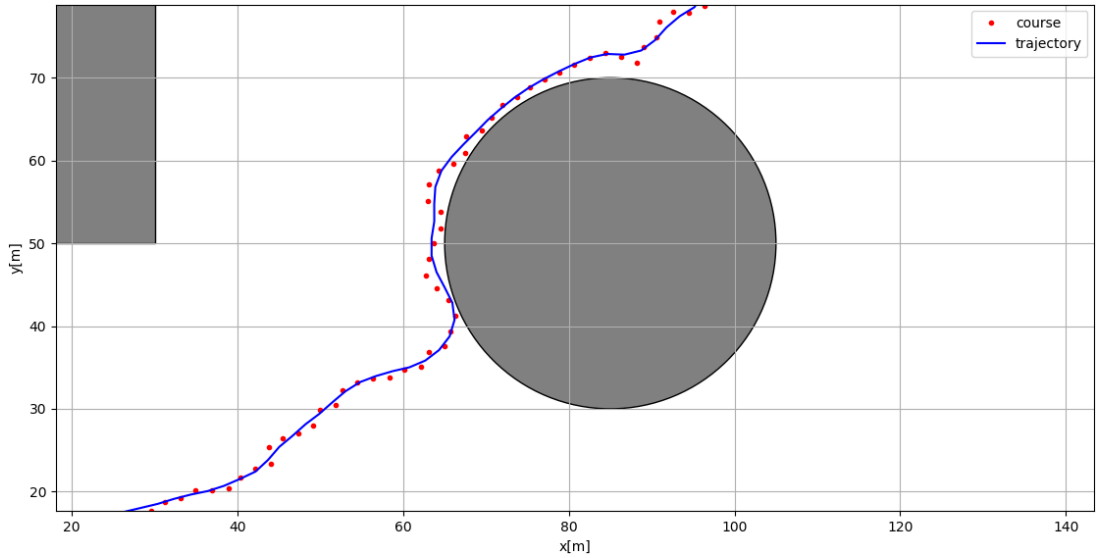
4.2 Pure Pursuit Metodunun Performans Analizi

Pure Pursuit metodu, başlangıç ve hedef nokta koordinatlarının verildiği, 2 boyutta 300m x 500m'lik alana sahip, çeşitli engellerin olduğu bir haritada simüle edilmiştir. Bu simülasyon da amaç RRT algoritmasını iterasyon sayısı parametresinin, algoritmanın bulduğu en kısa yolun uzunluğuna, yoldaki keskin kenarlara ve simülasyonun tamamlanma süresine bağlı olarak performansı gözlemlenecektir.

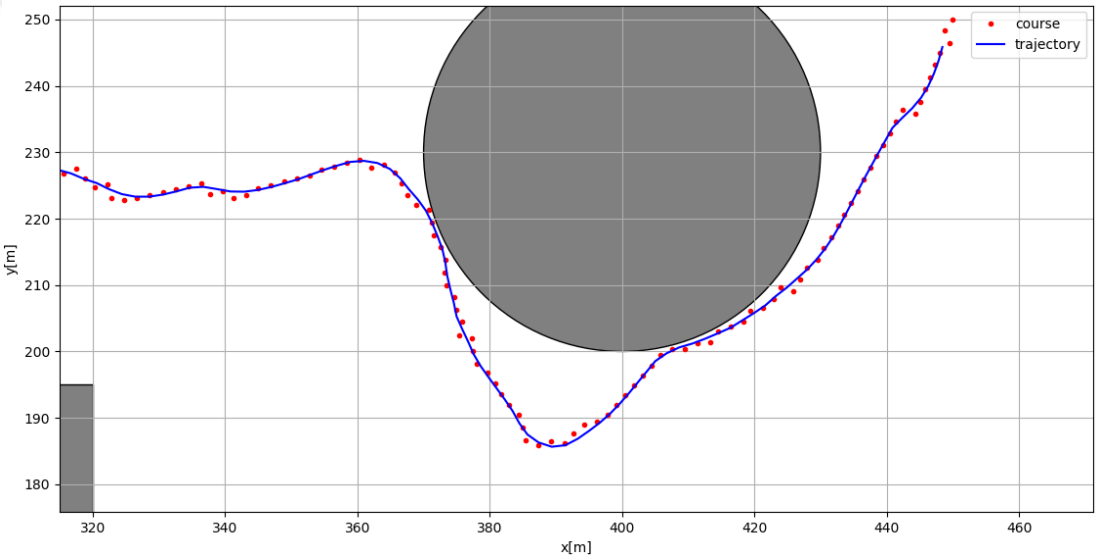


Şekil 4.1: Pure Pursuit Metodu İle RRT Algoritmasının Hibrit Kullanımı Simülasyon Sonucu.

Şekil 4.1'deki simülasyon sonucunda kırmızı renkteki yol RRT algoritması tarafından bulunan yolu lacivert renkteki yol ise Pure Pursuit metodu kullanılarak İHA'nın izlediği yolu göstermektedir. Simülasyonda ileri bakma mesafe parametresi bu ortam için en uygun olan parametre değeri 4.8 olarak ayarlanmıştır. Engellere yakın geçilen bölgeleri daha iyi inceleyebilmek adına engellerin olduğu bölgelere ait yakınlaştırılmış görüntüler Şekil 4.2'de gösterilmiştir:

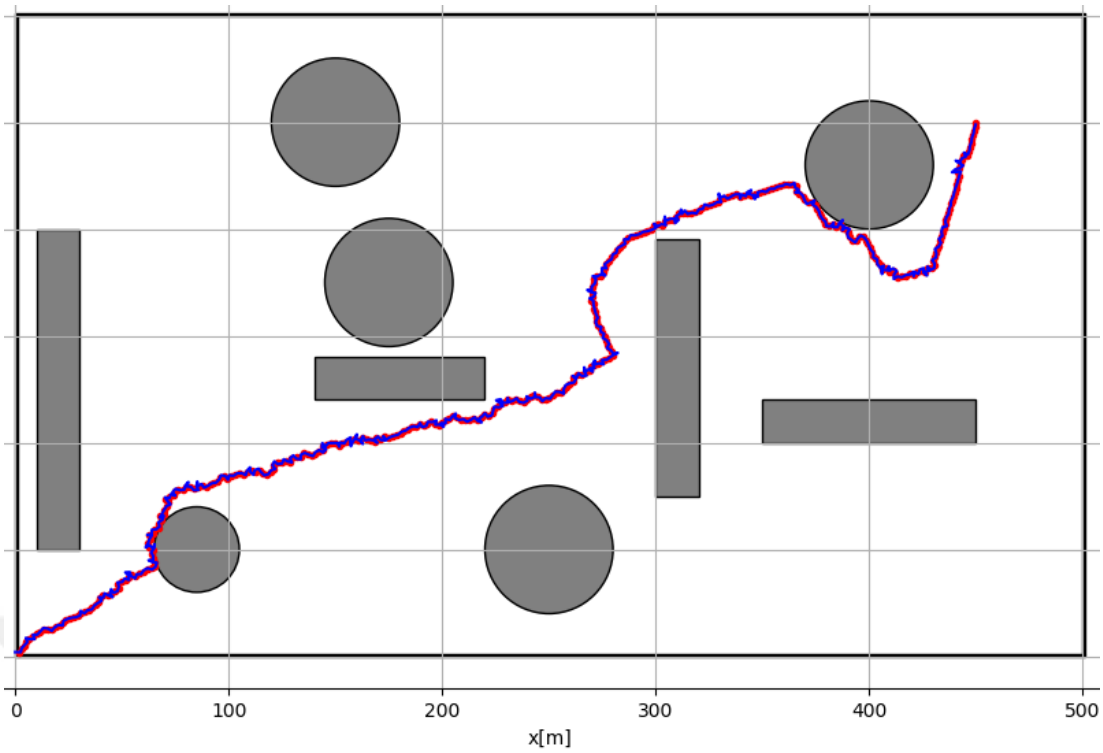


Şekil 4.2: Hibrit Yapıdaki Simülasyonun Yakınlaştırılmış Çıktısı-1.



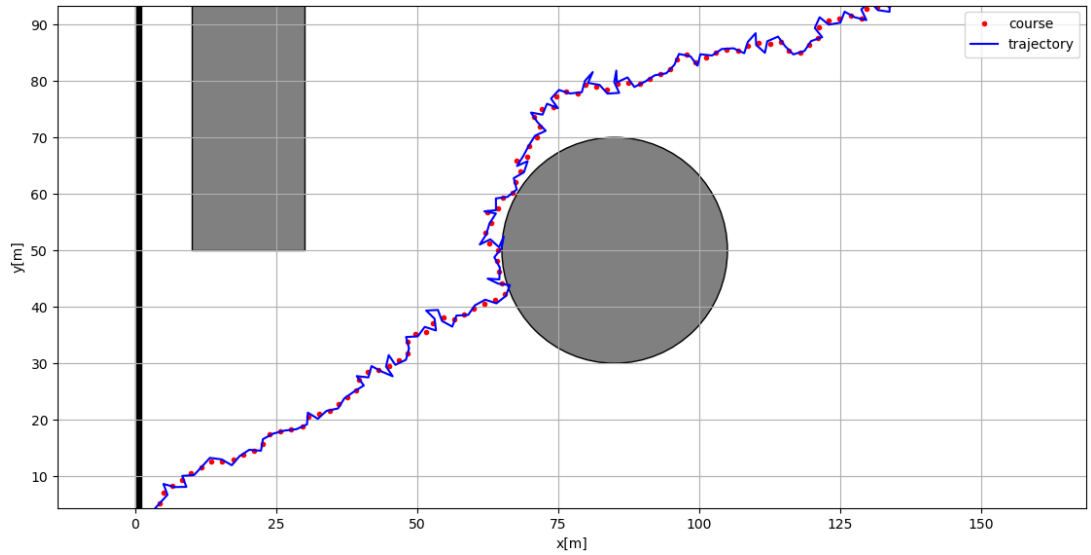
Şekil 4.3: Hibrit Yapıdaki Simülasyonun Yakınlaştırılmış Çıktısı-2.

Şekil 4.2 ve 4.3'te ki yakınlaştırılmış simülasyon sonuçlarından görüldüğü üzere İHA'nın hibrit yapıdaki metod kullanılarak izlediği yolda herhangi bir çarpışma olmamıştır. Ayrıca yol keskin kenarlardan arındırılarak optimize edilmiştir. Bu sayede güvenli bir hareket sağlanmış olup yol daha kısa hale gelmiştir.

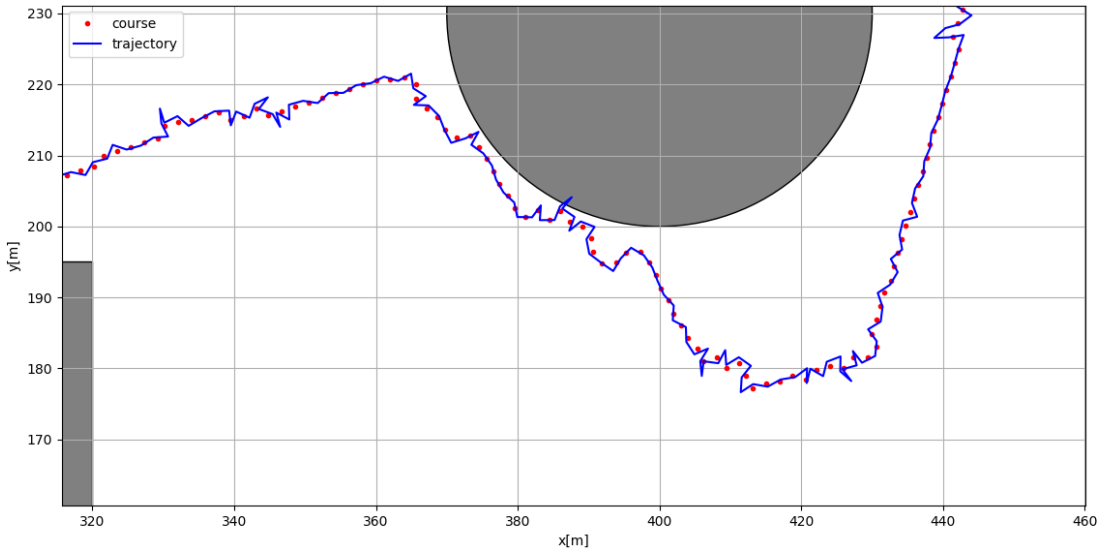


Şekil 4.4: Düşük İleri Bakma Mesafe Parametreli Hibrit Yapıdaki Simülasyon Sonucu.

Şekil 4.4'te ki simülasyon sonucunda kırmızı renkteki yol RRT algoritması tarafından bulunan yolu lacivert renkteki yol ise Pure Pursuit metodu kullanılarak İHA'nın izlediği yolu göstermektedir. Şekil 4.4'te sonucu gösterilen simülasyonda Pure Pursuit metoduna ait ileri bakma mesafesi parametresi düşük olarak ayarlanmıştır. Sonuç incelendiğinde yolda herhangi bir optimizasyon görülmemektedir. Çarpışma durumunun gerçekleşip gerçekleşmediğini anlamak ve metodun performansını görmek için Şekil 4.5'te görüntülerin yakınlaştırılmış hali verilmiştir.

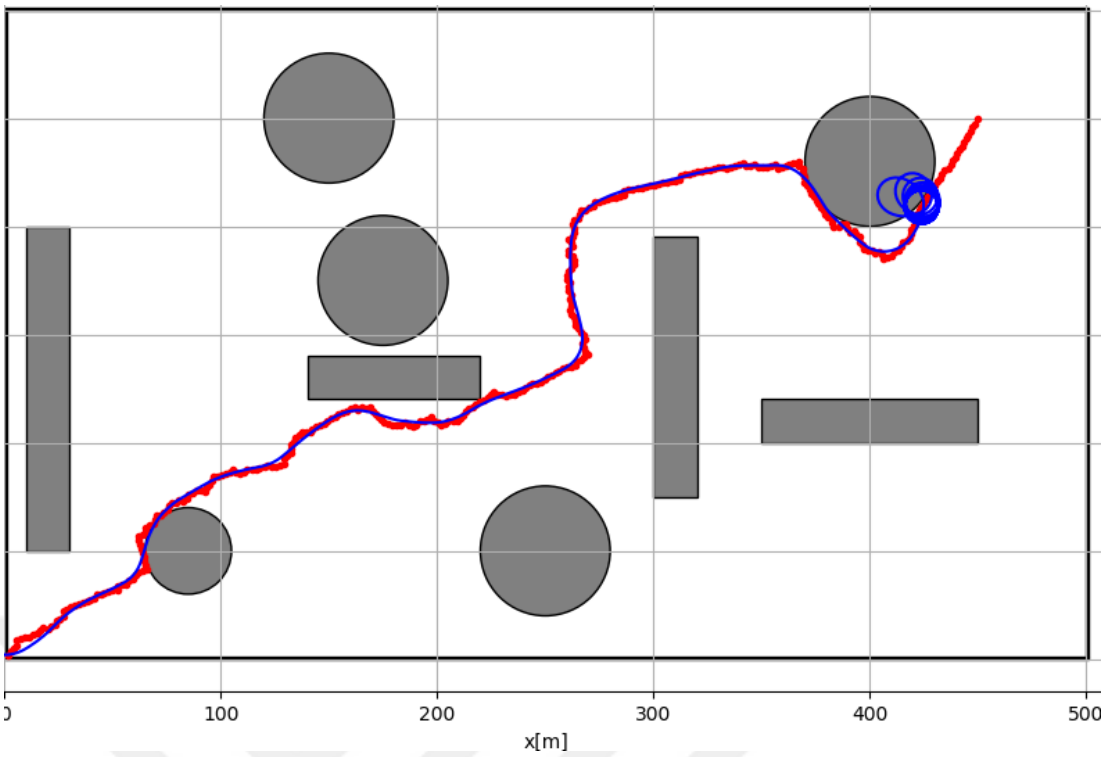


Şekil 4.5: Düşük İleri Bakma Mesafe Parametreli Hibrit Yapıdaki Yakınlaştırılmış Simülasyon Sonucu-1.



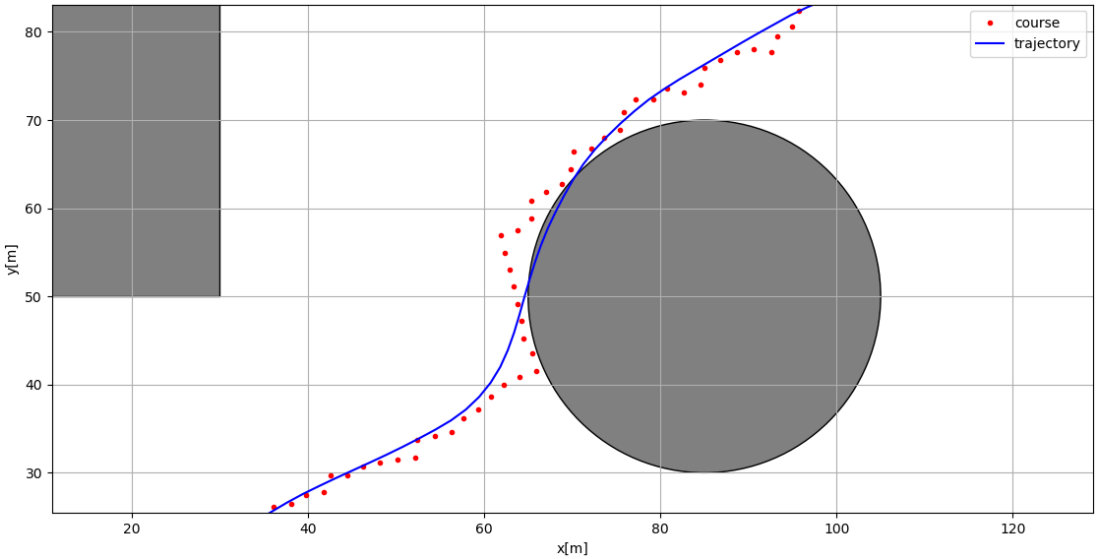
Şekil 4.6: Düşük İleri Bakma Mesafe Parametreli Hibrit Yapıdaki Yakınlaştırılmış Simülasyon Sonucu-2.

Yakınlaştırılmış simülasyon sonuçlarından görüldüğü üzere İHA engellere çarpmıştır. Hibrit kullanımın yolu optimize ederek keskin kenarları azaltması beklenirken RRT algoritmasının oluşturmuş olduğu yoldan daha uzun ve daha fazla sayıda keskin kenara sahip bir yol izlenmiştir.

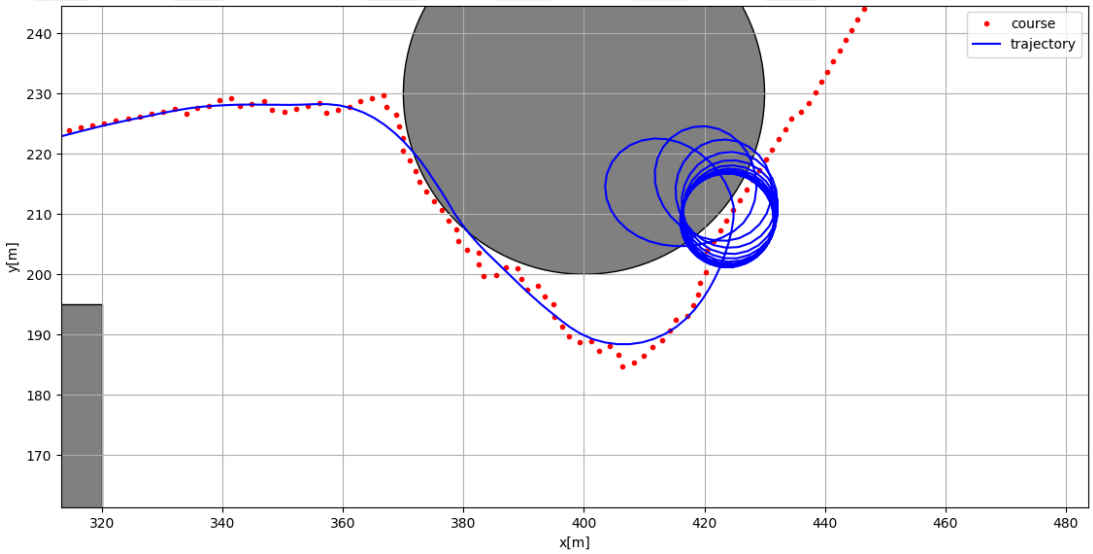


Şekil 4.7: Yüksek İleri Bakma Mesafe Parametreli Hibrit Yapıdaki Simülasyon Sonucu.

Şekil 4.6'da ki simülasyon sonucunda kırmızı renkteki yol RRT algoritması tarafından bulunan yolu lacivert renkteki yol ise Pure Pursuit metodu kullanılarak İHA'nın izlediği yolu göstermektedir. Şekil 4.7'de sonucu gösterilen simülasyonda Pure Pursuit metoduna ait ileri bakma mesafesi parametresi yüksek olarak ayarlanmıştır. Simülasyon sonucu incelendiğinde İHA, RRT algoritmasının oluşturmuş olduğu yolu sonuna kadar izlemekte başarısız olmuş ve hedefe ulaşamamıştır. Simülasyon sonuçlarının daha ayrıntılı incelenebilmesi için yakınlaştırılmış görüntüler Şekil 4.8 ve Şekil 4.9'da verilmiştir.



Şekil 4.8: Düşük İleri Bakma Mesafe Parametreli Hibrit Yapıdaki Yakınlaştırılmış Simülasyon Sonucu-1.



Şekil 4.9: Düşük İleri Bakma Mesafe Parametreli Hibrit Yapıdaki Yakınlaştırılmış Simülasyon Sonucu-2.

Yakınlaştırılmış simülasyon sonuçları incelendiğinde İnsansız Hava Aracı başarılı bir şekilde RRT algoritmasının oluşturmuş olduğu takip edememiştir. İzlenen yolda engellere temas edilmiştir, takip edilmesi gereken bazı yol noktaları es geçilmiştir ve hedefe çok yakın bir noktada İHA çember çizmeye başlamış ve döngüye girerek hedefe ulaşamamıştır.

4.2.1 Pure Pursuit Metodu Performans Sonuçları

Pure Pursuit metodunun en önemli parametresi önceki bölümlerde de bahsedildiği üzere ileri bakma mesafesi parametresidir. Bölüm 4.2'de yapılmış 3 farklı ileri bakma mesafesi parametre değerine göre elde edilen sonuçlardan, parametre değeri

normale göre düşük olduğunda aracın güvenli bir şekilde yolu izleyemediği, parametreye bağlı olarak çok kısa ileri yollar oluşturduğundan çok fazla zikzak yaptığı ve başarılı olamadığı görülmüştür.

Yüksek ileri bakma mesafesi parametresi girilen simülasyon sonucuna bakıldığında ise İHA'nın RRT algoritması tarafından oluşturulan yolu izlerken bazı yol noktalarını kaçırdığını, engellere temas etmeden geçmekte başarısız olduğunu ve en son olarakta izleyeceği yoldan saparak hedefe ulaşamadığı görülmüştür.

5. HİBRİT YAPIDA RRT ALGORİTMASI VE SAF TAKİP METODUNUN KULLANIMI

RRT Algoritması ve Pure Pursuit metodunun hibrit yapıda kullanımının performansını görmek için Python programlama dilinde yazılan kod ile farklı simülasyonlar yapılmıştır. Simülasyonlar iki boyutlu ortamda yapılmıştır fakat İHA için yapılan bu simülasyonlar İHA'nın irtifasının sabit olarak tutulup x ve y ekseninde hareket etmektedir. Bu şekilde simülasyonların aslında 3 boyutlu bir ortamda yapıldığını varsaymak mümkündür. Simülasyonlar farklı boyut ve özelliklere sahip haritalarda aynı parametre ve kontrolcüye sahip İHA modeli ile gerçekleştirilmiştir. Bu simülasyonlar;

- RRT ve Pure Pursuit'in hibrit yapıda kullanımı
- RRT ve Pure Pursuit'in farklı ortamlarda hibrit yapıda kullanımı
- RRT ve Kinematik modelin hibrit yapıda kullanımı

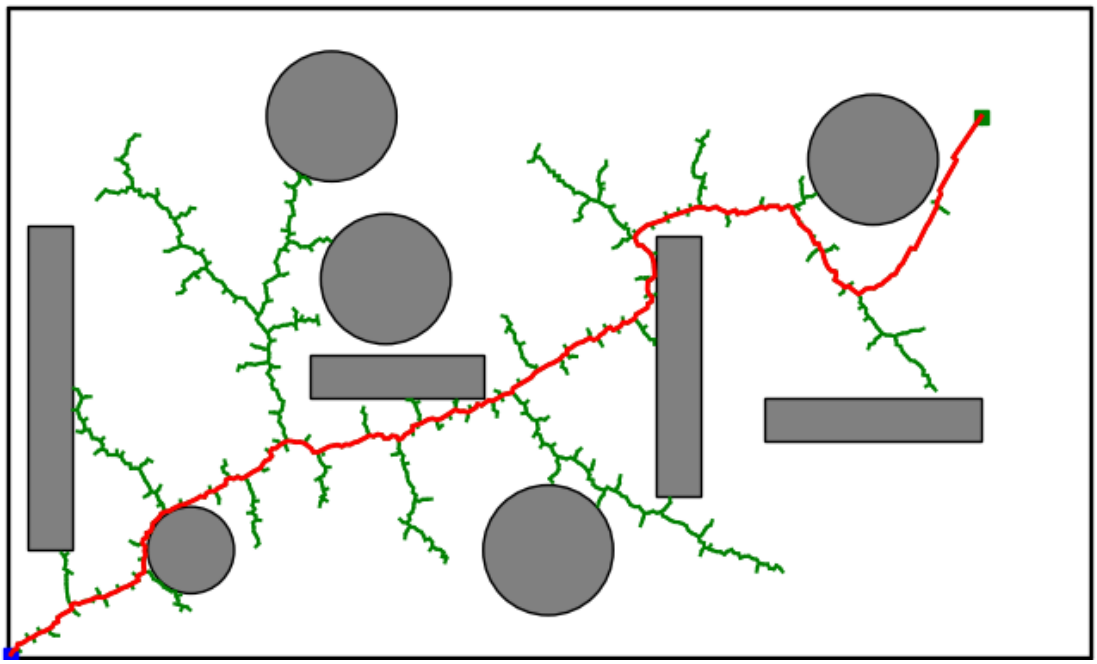
Olmak üzere ilk olarak RRT algoritması ve Pure Pursuit metodunun hibrit yapıda kullanımı test edilmiş ve pure pursuit metodunun RRT algoritması tarafından elde edilen yolu izlerken, engellere çarpma durumu, yol optimizasyonu ve yoldan sapma miktarı gibi hibrit yapıda kullanımın performansını gösteren parametrelere dikkat edilmiştir.

RRT algoritması büyük alanlarda daha verimli çalışmakta ve daha güvenli, kısayollar üretmektedir. Küçük alanlarda engel sayısı fazla olduğunda iki düğüm arası uzaklık parametresinden dolayı birbirine yakın engeller olduğunda algoritma hata yapabilmektedir. Bu yüzden ikinci olarak Hibrit yapının farklı ortamlarda başarısının test edilmesi amaçlanmıştır.

Üçüncü olarak ise RRT algoritması tarafından üretilen kısa yolu herhangi bir izleme veya optimizasyon algoritması içermeyen sadece kinematik model ve bir kontrolcüye sahip olan bir kinematik model ile izleyerek hibrit yapıların sonuçlarının karşılaştırılması amaçlanmıştır.

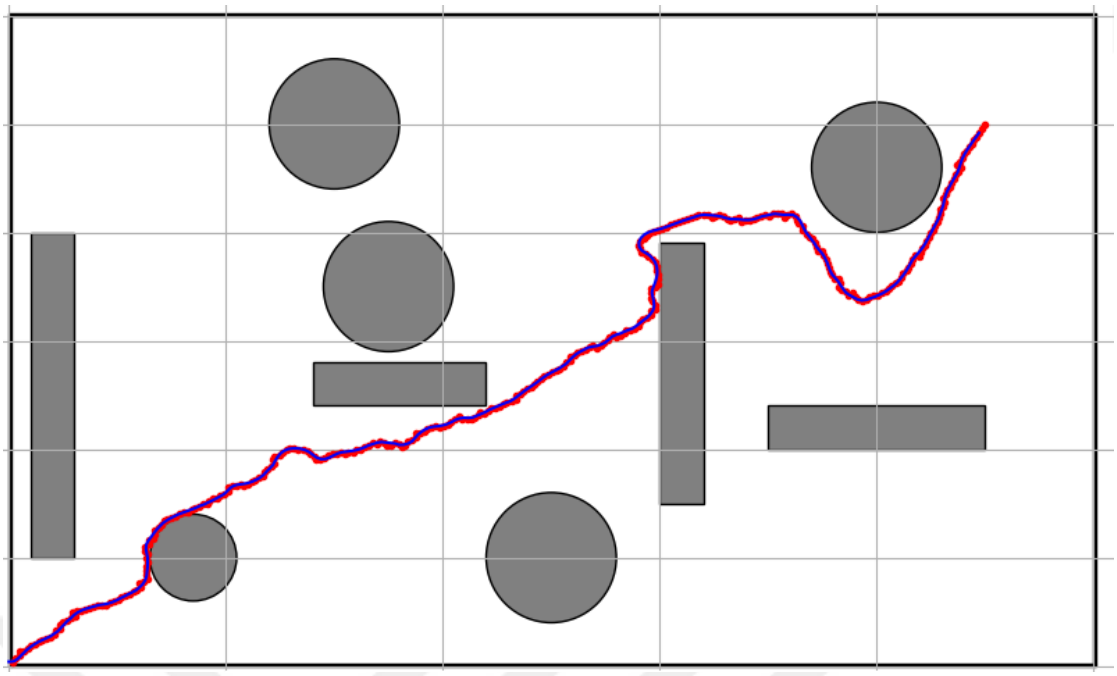
5.1 RRT Algoritması ile Pure-Pursuit Metodunun Hibrit Yapıda Kullanımın Simülasyon Sonuçları

Hibrit yapıda RRT algoritması ile Pure Pursuit metodunun kullanılma amacı RRT algoritmasının düşük iterasyonlarda, verilen haritada başlangıç noktası ile hedef nokta arasında elde ettiği en kısa yolun fazla sayıda keskin kenarlar içermesi, güvenli bir sürüş için tehlikeli olması ve yol uzunluğunun fazla olması gibi dezavantajları vardır. Yüksek iterasyon değerine sahip bir RRT algoritması daha az keskin kenar bulundurur ve az sayıda keskin kenar İHA'nın bu yolu izlerken daha güvenli gitmesine olanak sağlar, ayrıca yüksek iterasyon ile çalıştırılan algoritma sonucunda oluşan yol düşük iterasyon ile çalışan algoritma sonucundaki yola göre daha kısa olabilmektedir. Bu sebeple hibrit yapı RRT algoritmasından düşük iterasyonda elde edilen yolu Pure Pursuit metodu ile izlerken Pure Pursuit metodunun yolu optimize etmesi ve sürüşü güvenli hale getirmesi amaçlanmıştır. Bu sayede RRT algoritması az iterasyon sayısı ile daha düşük maliyette tutulurken yüksek iterasyonda çalıştırılan bir RRT algoritması kadar başarılı sonıuçlar elde etmektedir.



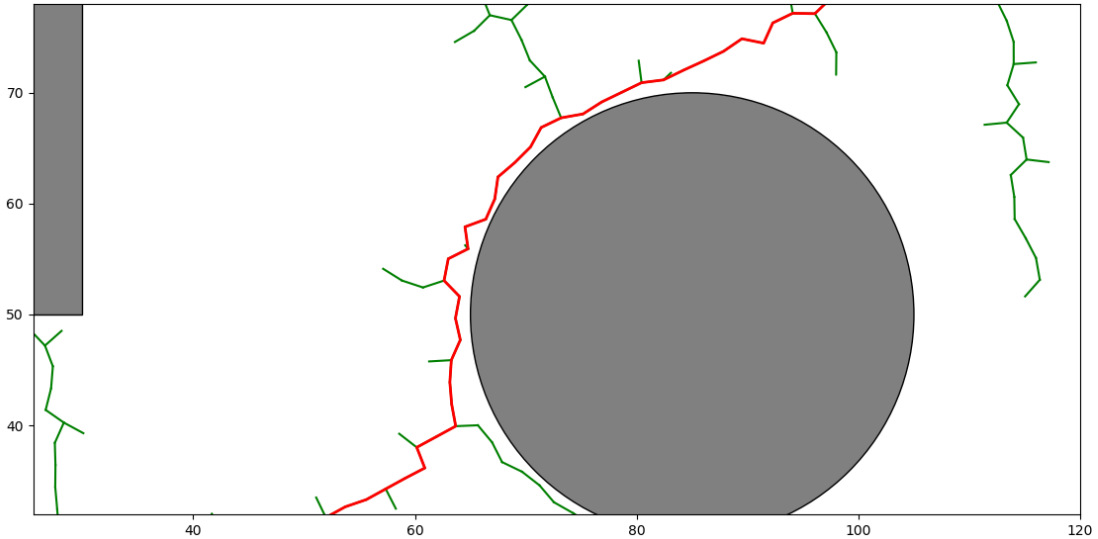
Şekil 5.1: 2000 İterasyon ile RRT Algoritmasının Simülasyon Çıktısı.

Bu şekilde 300m x 500**m'lik** bir haritada RRT algoritması 2000 iterasyon değeri ile çalıştırılmıştır. Başlangıç noktasından hedef noktasına giden en kısa yol kırmızı ile boyanmıştır. Bu yol birçok keskin kenar içermektedir. Bu yolun Pure Pursuit metodu ile izlenerek hedefe ulaşılmış olan simülasyon sonucu Şekil 5.1'de görülmektedir.



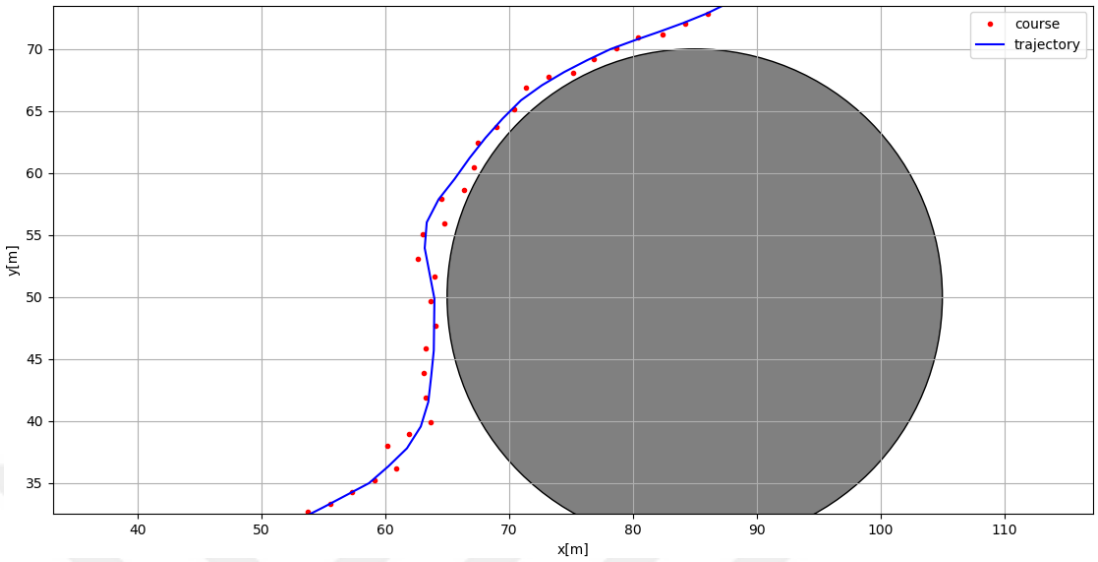
Şekil 5.2: 2000 İterasyonda RRT Çıktısının Pure Pursuit Metodu ile Hibrit Kullanım Simülasyon Sonucu.

Şekil 5.2'de Pure Pursuit metodu tarafından izlenerek İHA'yı hedefe ulaştırmış çıktı görülmektedir. Kırmızı renkte görünen yol RRT algoritması tarafından oluşturulan yoldur. Lacivert renkteki yol ise Pure Pursuit metodu kullanılarak optimize edilen yolun çıktısı olarak görülmektedir. Lacivert renkteki çıktıya bakıldığında aracın RRT tarafından oluşturulan yolu takip ettiği, yoldan sapma veya engele çarpma olmadığı görülmektedir. Yolun optimize edilmiş halini daha net anlayabilmek için İHA'nın engellere yakın geçtiği yerler Şekil 5.3'te yakınlaştırılarak gösterilmiştir.

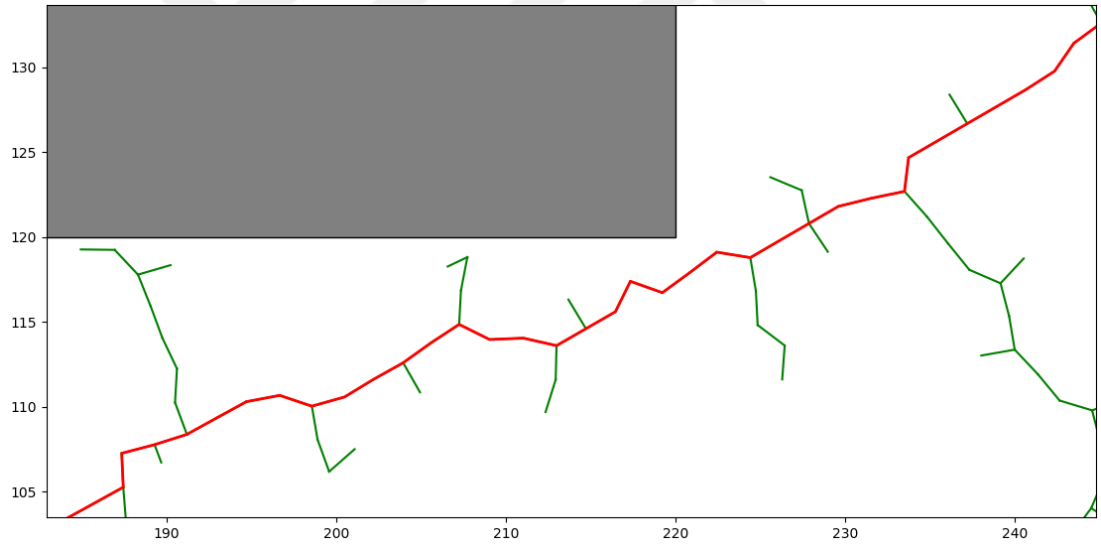


Şekil 5.3: 2000 İterasyon ile RRT Algoritmasının Yakınlaştırılmış Simülasyon Çıktısı-1.

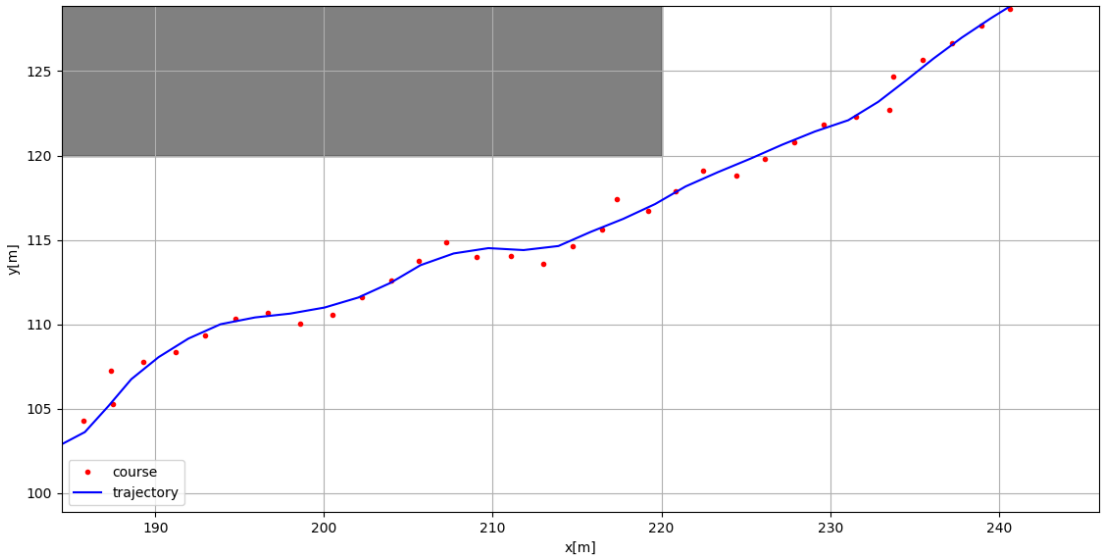
Şekil 5.3'te 2000 iterasyonda RRT algoritmasının çıktısının yakınlaştırılmış hali verilmiştir. Görüldüğü üzere yol engele çok yakın ve çok fazla keskin kenar içermektedir.



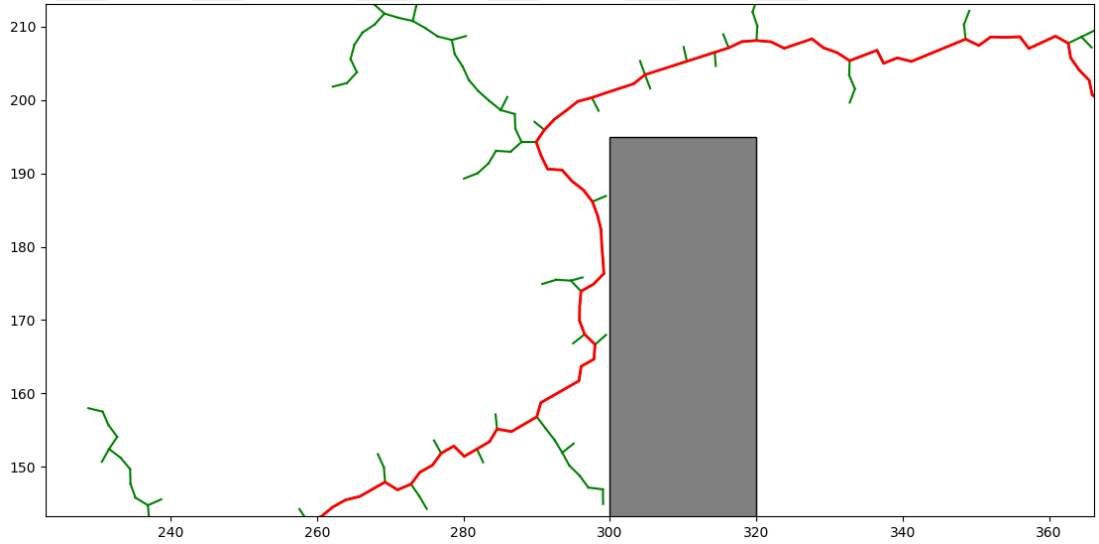
Şekil 5.4: 2000 İterasyonda Hibrit Yapıda Yakınlaştırılmış Simülasyon Çıktısı-1.



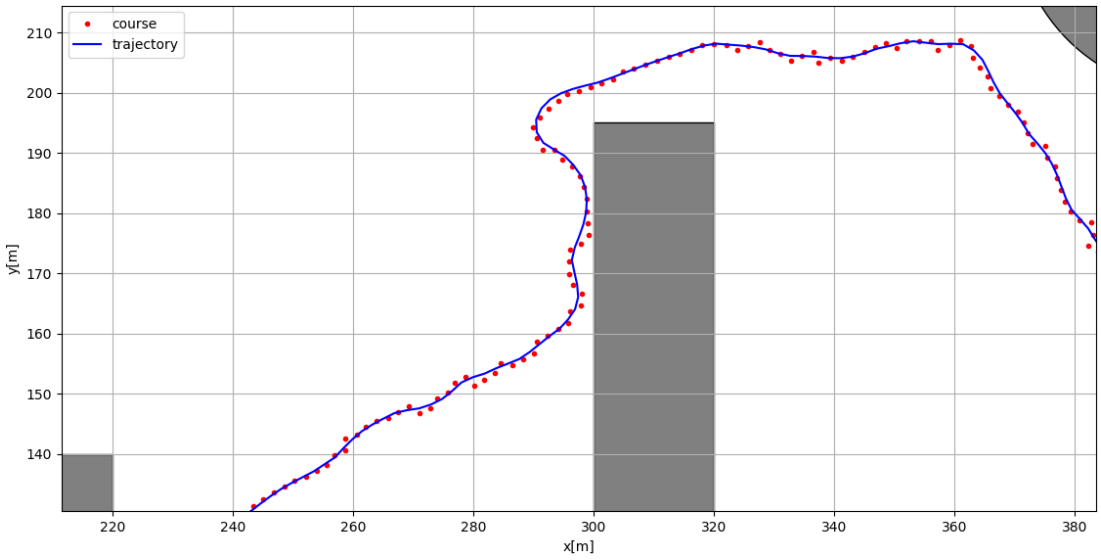
Şekil 5.5: 2000 İterasyon ile RRT Algoritmasının Yakınlaştırılmış Simülasyon Çıktısı-2.



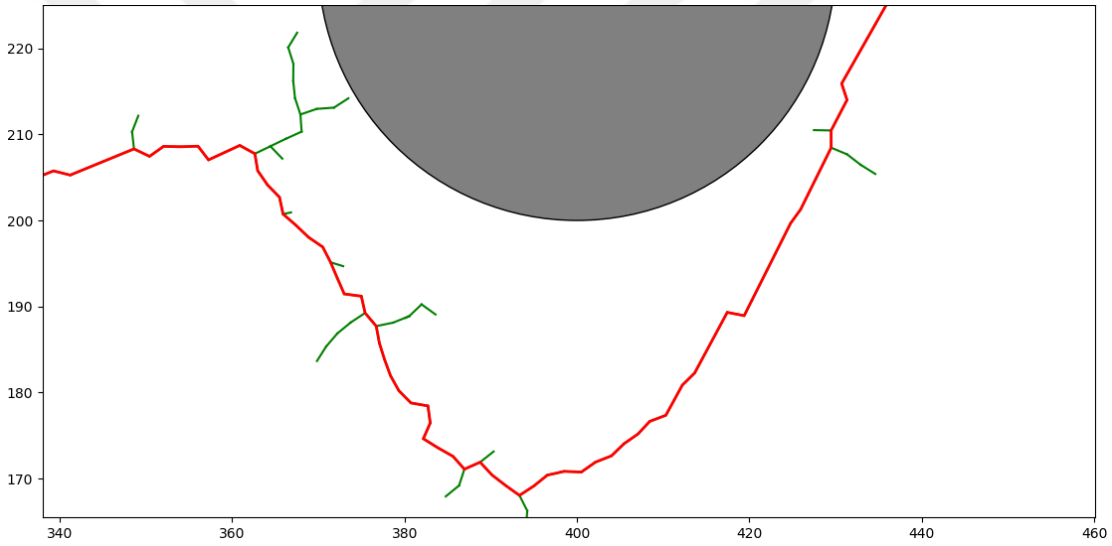
Şekil 5.6: 2000 İterasyonda Hibrit Yapıda Yakınlaştırılmış Simülasyon Çıktısı-2.



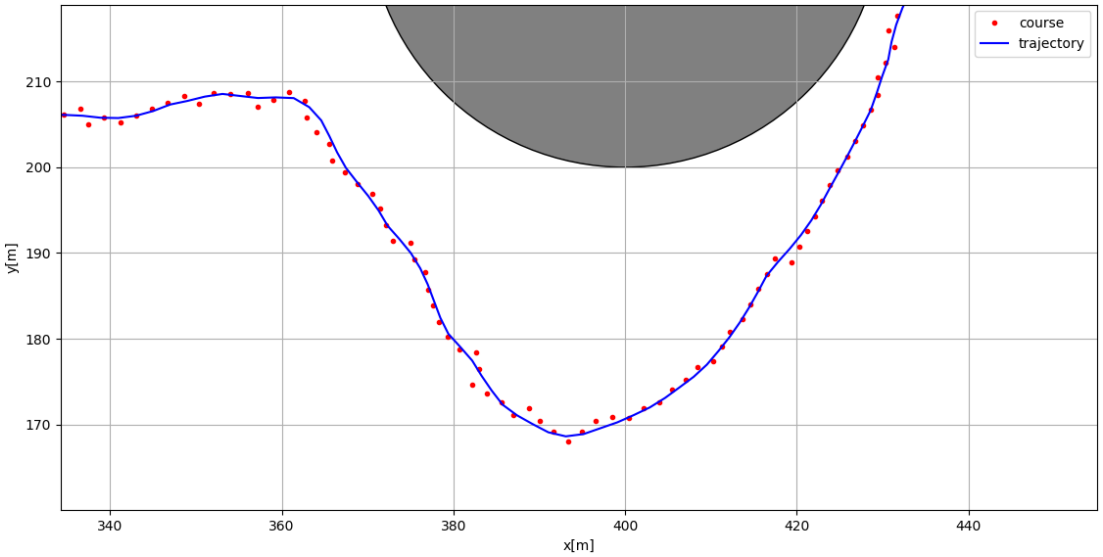
Şekil 5.7: 2000 İterasyon ile RRT Algoritmasının Yakınlaştırılmış Simülasyon Çıktısı-3.



Şekil 5.8: 2000 İterasyonda Hibrit Yapıda Yakınlaştırılmış Simülasyon Çıktısı-3.



Şekil 5.9: 2000 İterasyon ile RRT Algoritmasının Yakınlaştırılmış Simülasyon Çıktısı-4.

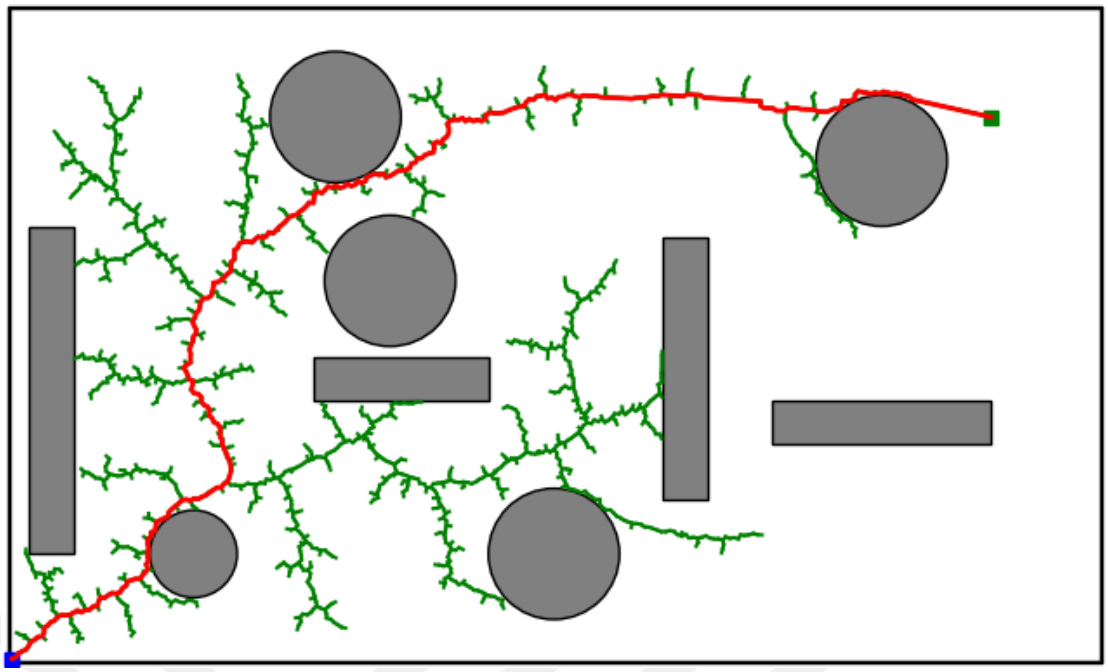


Şekil 5.10: 2000 İterasyonda Hibrit Yapıda Yakınlaştırılmış Simülasyon Çıktısı-4.

Şekil 5.1 ile Şekil 5.10 arasındaki şekillerde görüldüğü üzere 2000 iterasyondaki RRT algoritmasının çıktısnının Pure Pursuit metodu ile izlenmiş hali verilmiştir. Kırmızı noktalar RRT algoritması tarafından oluşturulan yoldaki birbirine bağlı ağaçların düğümleridir bunlar (waypoint) yol noktaları olarak gösterilmiştir. Lacivert renk ile gösterilen çizgisi ise Pure Pursuit çıktısıdır. Şekil 5.9'da görüldüğü üzere Pure Pursuit metodu RRT tarafından oluşturulan yolu optimize ederek engele çarpmadan güvenli bir hareket sağlamıştır ve keskin kenarları yumuşatarak yolu da daha kısa hale getirmiştir.

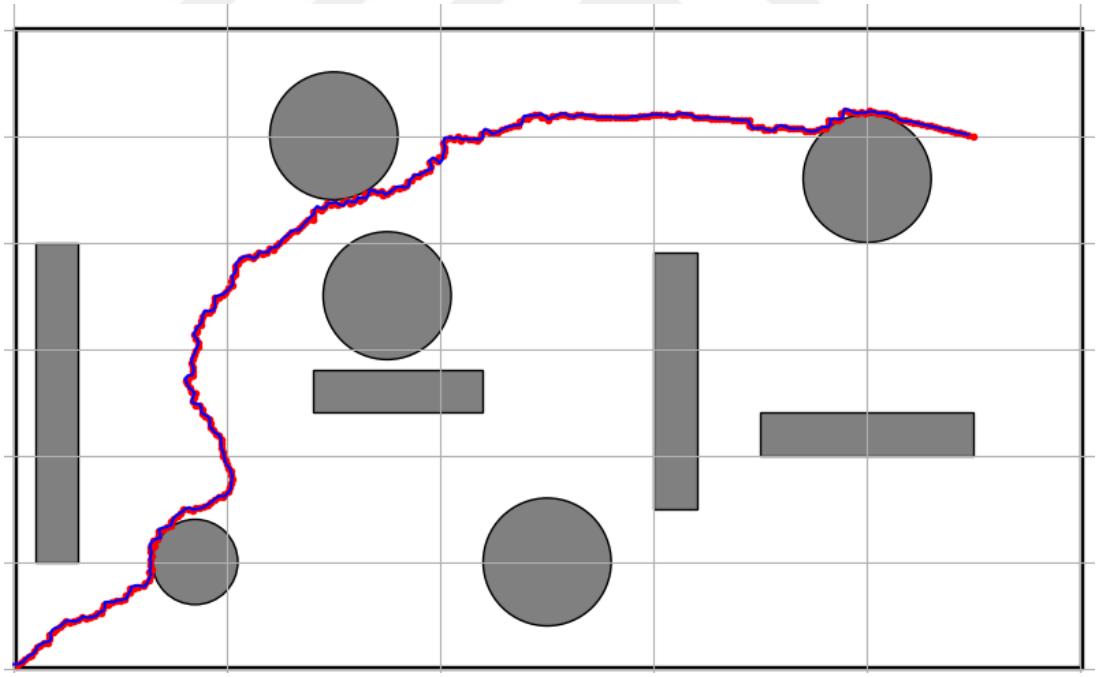
5.2 RRT Algoritması ile Pure Pursuit Metodunun Hibrit Kullanımı İle RRT Algoritmasının Kinematik Model İle Hibrit Kullanımının Karşılaştırılması

RRT Algoritması ile Pure Pursuit metodunun hibrit kullanımı ile elde edilen sonuçlar Bölüm 5.1'de görülmektedir. Bu bölümde yine RRT algoritması kullanılarak elde edilen yol herhangi bir optimizasyon veya yol iyileştirmesi yapmayan bir yol izleme metodu kullanılarak elde edilen sonuçlar karşılaştırılacaktır. Hibrit yapıda sadece İHA'nın kinematik denklemleri ve kontrolcüsü kullanılarak RRT tarafından oluşturulan yol takip edilmiştir.



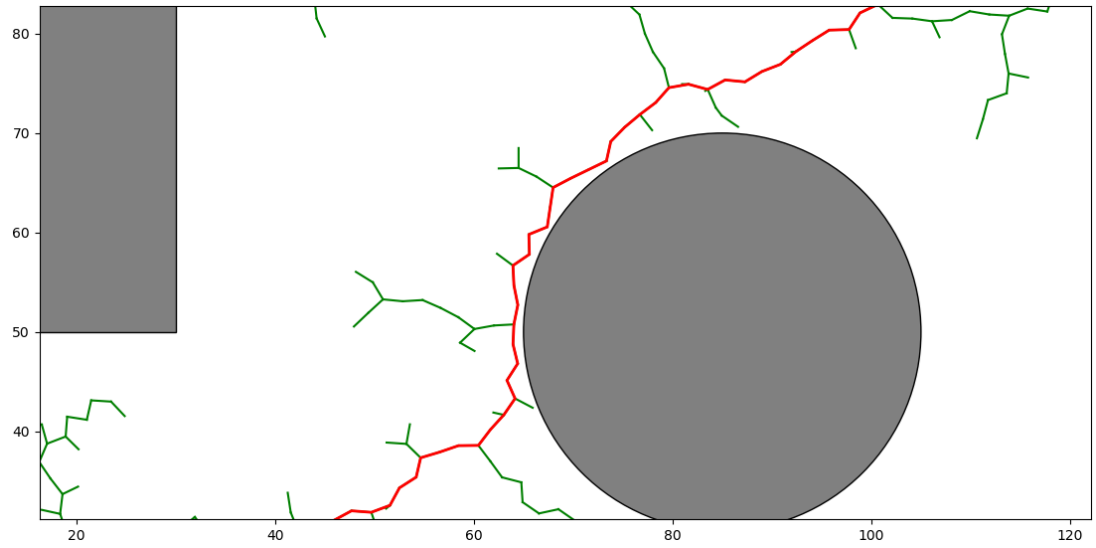
Şekil 5.11: 10000 İterasyon ile RRT Algoritmasının Simülasyon Çıktısı.

Şekil 5.11'de RRT algoritmasının 10000 iterasyonda çalıştırılması sonuncunda ki simülasyon sonucu görülmektedir.



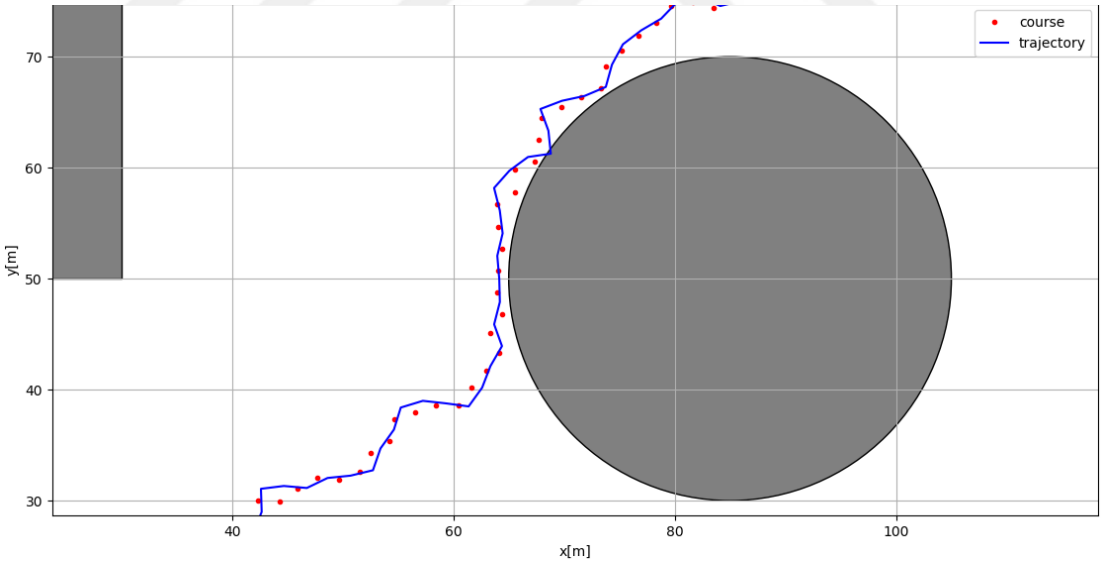
Şekil 5.12: 10000 İterasyondaki RRT Çıktısının Kinematik Model İle Takibinin Simülasyon Çıktısı.

Şekil 5.12'de lacivert ile boyanmış yol RRT Algoritması tarafından oluşturulan yolun izlenmesi ile elde edilmiştir. Kırmızı ile boyanmış yol ise RRT algoritması tarafından oluşturulan yoldur. Simülasyon sonucu incelendiğinde İHA'nın güvenli bir hareket sağladığı görülmektedir.



Şekil 5.13: 10000 İterasyon ile RRT Algoritmasının Yakınlaştırılmış Simülasyon Çıktısı-1.

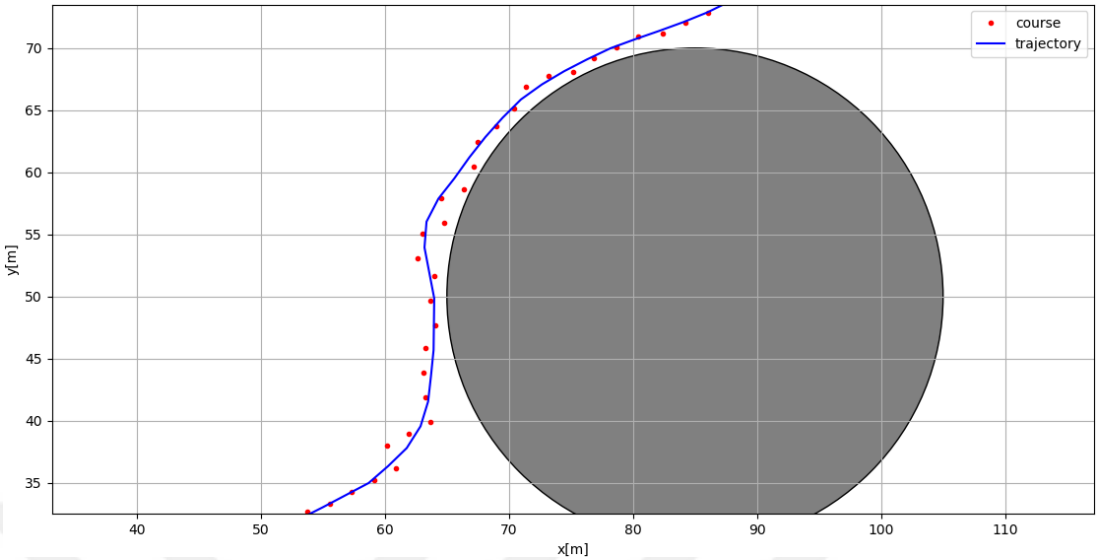
Şekil 5.13'de 10000 iterasyonda RRT algoritmasının çıktısının yakınlaştırılmış hali verilmiştir. İterasyon sayısı bir önceki bölümdeki simülasyona göre daha fazla olsada RRT algoritmasının sahip olduğu yapıdan dolayı rastgele atanan ağaçlar keskin kenarlara yol açmaktadır. Bu çıktı da da görüldüğü üzere yol engele çok yakın ve çok fazla keskin kenar içermektedir.



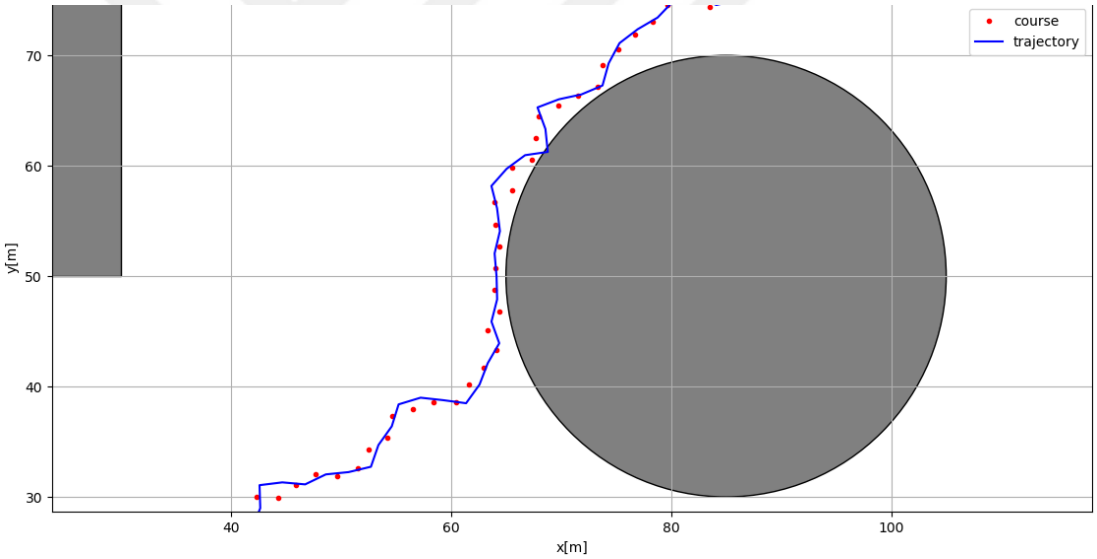
Şekil 5.14: 10000 İterasyonda RRT İle Pure Pursuit Algoritmalarının Hibrit Yapıda Yakınlaştırılmış Simülasyon Çıktısı-1.

Şekil 5.14'te lacivert ile boyanmış yol RRT Algoritması tarafından oluşturulan yolun hibrit yapıda kullanılan kinematik modelin izlenmesi ile elde edilmiştir. Kırmızı ile boyanmış yol ise RRT algoritması tarafından oluşturulan yoldur. Herhangi bir optimizasyon yöntemi olmayan bu metot ile yolun izlenmesi sonucunda yolda herhangi bir iyileştirme ve kısaltılma görülmemiş, keskin kenarların sayısı

azalmamış ve bazı yerlerde izlenen yol engele çok yakın olduğu için güvenli bir hareket sağlanamamıştır.



Şekil 5.15: RRT ile Pure Pursuit Algoritmalarının Hibrit Kullanımı Simülasyon Sonucu-1.



Şekil 5.16: RRT Algoritması ile Kinematik Modelin Hibrit Kullanımı Simülasyon Sonucu-2.

İki hibrit yapı kullanımı da karşılaştırıldığında RRT ile Pure Pursuit metodunun hibrit olarak kullanımı çok daha başarılı bir sonuç vermiştir. RRT ile Pure Pursuit kullanımı sonucunda, yol kısaltılmış, optimize edilmiş ve güvenli bir hareket planlaması yapılmıştır.

6. SONUÇLAR

Hibrit yapıda RRT algoritması ile Pure Pursuit metodunun kullanımı Bölüm 5'te simüle edilmiştir. Simülasyon sonuçları incelendiğinde Pure Pursuit metodunun doğru ileri bakma mesafesi parametresi girildiğinde yolu optimize ettiği görülmüştür. Yol izleme algoritması olarak Pure Pursuit metodu kullanıldığında kinematik modele göre daha kısa bir yol elde edebilmekte ve İHA daha hızlı hedef noktaya ulaşmaktadır. Pure Pursuit metodunun yoldaki kenar sayılarını azaltması güvenli bir hareket sağlamaktadır. Dar alanlarda ve engellerin yoğun bulunduğu ortamlarda RRT algoritması düzgün bir yol oluşturamadığından verimli bir yol takibi sağlanamamaktadır. Statik engellerin bulunduğu ortamlarda ilgili parametrelere uygun değerlerin verilmesi sonucunda RRT algoritması ve Pure Pursuit metodunun hibrit kullanımı başarılı bir sonuç vermektedir.

Pure Pursuit metoduna ait ileri bakma mesafesi parametresi ileride uyarlanabilir (adaptif) hale getirilmesi gelecekteki çalışmalar için düşünülmektedir. Uyarlanabilir ileri bakma mesafesi parametresi İHA'nın başlangıçta elde ettiği harita ve engel bilgilerini göre ayarlanarak İHA haritayı keşfettikçe haritadaki engel durumuna göre ileri bakma mesafesi parametresinin güncellenmesi öngörülmüştür.

KAYNAKÇA

Elmokadem, T., & Savkin, A. V. (2021). A Hybrid Approach for Autonomous Collision-Free UAV Navigation in 3D Partially Unknown Dynamic Environments. Drones, 5(3), 57.

Chen, Z., Zhang, Y., Zhang, Y., Nie, Y., Tang, J., & Zhu, S. (2019). A hybrid path planning algorithm for unmanned surface vehicles in complex environment with dynamic obstacles. *IEEE Access*, (*7)*: 126439-126449.

Zhong, X., Tian, J., Hu, H., & Peng, X. (2020). Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment. *Journal of Intelligent & Robotic Systems*, *99*(1), 65-77.

Coulter, R. C. (1992). Implementation of the pure pursuit path tracking algorithm. *Carnegie-Mellon UNIV Pittsburgh PA Robotics INST*.

Boğar, E. (2016). *Tek ve çok amaçlı robot yol planlama problemi için hibrit bir optimizasyon yöntemi* (Yüksek Lisans Tezi). Pamukkale Üniversitesi Fen Bilimleri Enstitüsü.

Čížek, P., Masri, D., & Faigl, J. (2017, September). Foothold placement planning with a hexapod crawling robot. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4096-4101). IEEE.

Saska, M., Baca, T., Thomas, J., Chudoba, J., Preucil, L., Krajnik, T., ... & Kumar, V. (2017). System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization. *Autonomous Robots*, *41*(4), 919-944.

Reeves, M. C. (2019). *An Analysis of Path Planning Algorithms Focusing on A* and D* (Doctoral dissertation). University of Dayton, Ohio.

Atalı, G. (2019). *Dağıtık mobil robotlar için yeni bir otonom yol planlama ve engel tespit sisteminin tasarımı* (Doktora Tezi). Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Sakarya.

Yap, P. (2002, May). Grid-based path-finding. In *Conference of the canadian society for computational studies of intelligence* (pp. 44-55). Springer, Berlin, Heidelberg.

Yang, K., & Sukkarieh, S. (2010). An analytical continuous-curvature path-smoothing algorithm. *IEEE Transactions on Robotics*, *26*(3), 561-568.

Huang, Z., Chu, D., Wu, C., & He, Y. (2018). Path planning and cooperative control for automated vehicle platoon using hybrid automata. *IEEE Transactions on Intelligent Transportation Systems*, *20*(3), 959-974.

Yang, S. M., & Lin, Y. A. (2021). Development of an improved rapidly exploring random trees algorithm for static obstacle avoidance in autonomous vehicles. *Sensors*, *21*(6), 2244.

Zhong, X., Tian, J., Hu, H., & Peng, X. (2020). Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment. *Journal of Intelligent & Robotic Systems*, *99*(1), 65-77.

Chai, Y., & Hassani, V. (2019). Hybrid collision avoidance with moving obstacles. *IFAC-PapersOnLine*, *52*(21), 302-307.

LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning.

Kiani, F., Seyyedabbasi, A., Aliyev, R., Gulle, M. U., Basyildiz, H., & Shah, M. A. (2021). Adapted-RRT: novel hybrid method to solve three-dimensional path planning problem using sampling and metaheuristic-based algorithms. *Neural Computing and Applications*, *33*(22), 15569-15599.

Jacobs, S. A., Stradford, N., Rodriguez, C., Thomas, S., & Amato, N. M. (2013, May). A scalable distributed RRT for motion planning. In *2013 IEEE International Conference on Robotics and Automation* (pp. 5088-5095). IEEE.

Yafei, L., Anping, W., Qingyang, C., & Yujie, W. (2020, September). An improved UAV path planning method based on RRT-APF hybrid strategy. In *2020 5th*

International Conference on Automation, Control and Robotics Engineering (CACRE) (pp. 81-86). IEEE.

Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, *30*(7), 846-894.

Giesbrecht, J., Mackay, D., Collier, J., & Verret, S. (2005). *Path tracking for unmanned ground vehicle navigation: Implementation and adaptation of the pure pursuit algorithm*. DEFENCE RESEARCH AND DEVELOPMENT SUFFIELD (ALBERTA).

Chen, Z., Zhang, Y., Zhang, Y., Nie, Y., Tang, J., & Zhu, S. (2019). A hybrid path planning algorithm for unmanned surface vehicles in complex environment with dynamic obstacles. *IEEE Access*, *7*, 126439-126449.

Özdemir, A. *Mobil Robot Navigasyon Sistemi Geliştirilmesi* (Doktora Tezi), İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul

Ge, S. S., & Cui, Y. J. (2000). New potential functions for mobile robot path planning. *IEEE Transactions on robotics and automation*, *16*(5), 615-620.

Snider, J. M. (2009). Automatic steering methods for autonomous automobile path tracking. *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08*.

Yan Ding (2020), Three Methods of Vehicle Lateral Control: Pure Pursuit, Stanley and MPC https://dingyan89.medium.com/three-methods-of-vehicle-lateral-control-pure-pursuit-stanley-and-mpc-db8cc1d32081 adresinden alınmıştır.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, *1*(1), 269-271.

Ferguson, D., & Stentz, A. (2007). Field D*: An interpolation-based path planner and replanner. In *Robotics research* (pp. 239-253). Springer, Berlin, Heidelberg.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, *4*(2), 100-107.

Gopikrishnan. S, B. V.S. Shravan (2011). Path Planning Algorithms: A Comparative Study. a Indian Institute of Space Science & Technology, Thiruvananthapuram-695547, India b ISRO Satellite Centre, Bangalore-560017, India

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : OKUR, Hüsnü Umut

Uyruğu : T.C

Eğitim

Derece	**Üniversite ve Bölüm**	**Mezuniyet tarihi**
Lisans	Erciyes Üniversitesi	2019

Yabancı Dil

İngilizce

Tezden Türetilen Yayınları/Sunumları

8. ULUSLARARASI MARMARA FEN BİLİMLERİ KONGRESİ IMASCON 2022 BAHAR