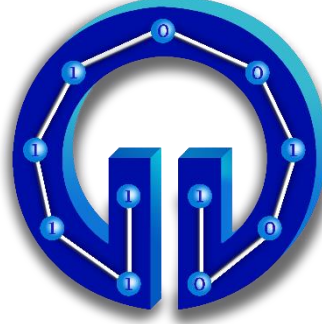


**KARADENİZ TEKNİK ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**İNSANSIZ HAVA ARAÇLARINDA FORMASYON KONTROL VE YÖRÜNGE  
PLANLAMA**

**BİTİRME PROJESİ**

**Zehra ÖZÇİRİŞ**

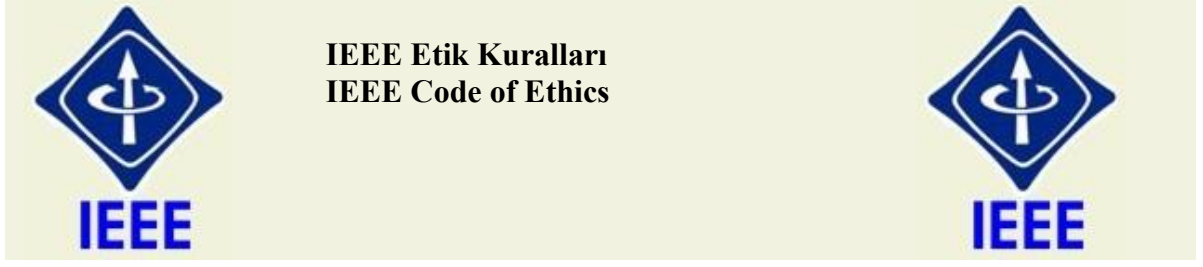
**2024-2025 BAHAR DÖNEMİ  
KARADENİZ TEKNİK ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**İNSANSIZ HAVA ARAÇLARINDA FORMASYON KONTROL VE YÖRÜNGE  
PLANLAMA**

**BİTİRME PROJESİ**

**Zehra ÖZÇİRİŞ**

**2024-2025 BAHAR DÖNEMİ**



Mesleğime karşı şahsi sorumluluğumu kabul ederek, hizmet ettiğim toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmaya söz verdiğimi ve aşağıdaki etik kurallarını kabul ettiğimi ifade ederim:

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;
3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;
4. Her türlü rüşveti reddetmek;
5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;
6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;
7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriye kabul etmek ve eleştiriye yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;
8. Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği, veya cinsiyet ifadesi üzerinden ayrımcılık yapma durumuna girişmemek;
9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;
10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.

## ÖNSÖZ

“Çoklu İnsansız Hava Araçlarında Formasyon Kontrol ve Yörünge Planlama” adlı bu çalışma, Karadeniz Teknik Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü’nde bahar dönemi bitirme projesi olarak hazırlanmıştır. Başta bitirme çalışmama danışmanlık yapan Dr. Öğr. Üyesi Selçuk CEVHER olmak üzere yardımcı dokunan diğer tüm hocalarıma, teknisyenlere ve tüm arkadaşlarıma teşekkür ederim. Bölüm olanaklarının bitirme çalışmalarında kullanılmasına izin verdiği için bölüm başkanlığına, desteklerinden dolayı Mühendislik Fakültesi Dekanlığına ve KTÜ Rektörlüğüne de teşekkür ederim.

Ayrıca eğitimim süresince bana her zaman ve her koşulda destek veren başta annem olmak üzere tüm aile fertlerime de teşekkür ederim.

Zehra ÖZÇİRİŞ  
Trabzon 2025

## İÇİNDEKİLER

	Sayfa No
IEEE ETİK KURALLARI	II
ÖNSÖZ	III
İÇİNDEKİLER	IV
ÖZET	V
1. GENEL BİLGİLER	1
1.1. Giriş	1
1.2. Problem Tanımı	2
2. YAPILAN ÇALIŞMALAR	2
2.1. Sistem Mimarisi	2
2.1.2. Formasyon Kontrol	2
2.1.2.1. Lider-Takipçi Algoritması	2
2.1.3. Yörünge Planlama	4
2.1.3.1. RRT Algoritması	4
2.1.3.2. RRT* Algoritması	5
2.2. Sistem İcrası	7
2.2.1. Başlangıç Adımı	7
2.2.2. Formasyon Kurma ve Yörünge Planlama Adımı	8
2.2.3. Yörünge Takibi ve Formasyon Koruma Adımı	9
3. SONUÇLAR	10
3.1. Entegre Sistem Başarımı	10
3.2. Formasyon Kontrolü ve Korunması	10
3.3. Dinamik Engelden Kaçınma Yeteneği	10
3.4. Gerçekçi Senaryo Simülasyonu	10
3.5. Problemin Bütünsel Çözümü	10
3.6. Optimizasyon Yaklaşımı	11
4. ÖNERİLER	11
4.1. Statik Engelli Ortamda Rota Planlama için Geliştirme Önerileri	11
4.2. Dinamik Formasyon Şekli ve Açısı için Geliştirme Önerileri	12
5. KAYNAKLAR	14
6. EKLER	15
STANDARTLAR ve KISITLAR FORMU	15

## ÖZET

İnsansız hava aracı teknolojisinde ve sürü zekası teknolojisinde devam eden gelişmelerle beraber İHA sürülerinin arama-kurtarma gibi görevlerde kullanımı da büyük ilgi görmeye başladı. İHA formasyonları için yörünge planlama görev icrasının anahtar kısmıdır.

Bu çalışmada İHA formasyonları için dağıtık haldeki sürünün formasyon oluşturup örnekleme tabanlı bir planlama algoritması olan RRT (Rapidly exploring Random Tree) kullanılarak mevcut kumandan hedef konuma yörünge planlaması sağlanarak formasyon kontrol ve yörünge planlama çalışmaları entegre edilip tek proje çatısı altında çözümlenmeye çalışılmıştır.

Sistem kurulurken belirli kısıtlar esas alınmıştır: rastgele konumlandırılmış otonom insansız hava araçlarının formasyon şeklini oluşturması ve bu şekli koruması, çarpışmalardan ve engellerden kaçınılması, hedef konuma yörünge planlanması ve ilerlenmesi.

Proje esas alınan kısıtlara uyarak lider-takipçi algoritması ve RRT algoritması tabanında tasarlanmıştır. Projede RRT ile yörünge planlaması yapıldıktan sonra dağıtık halde bulunan İHA sürüsü formasyon oluşturur. Oluşturulan formasyon korunarak RRT ile oluşturulan yörünge boyunca hedefe doğru ilerler. Yörünge üzerinde dinamik bir engelle karşılaşılması durumunda formasyon hareketini durudurur ve engel yörünge üzerinden çekildiği zaman hareketine devam eder.

**Anahtar Kelimeler:** Formasyon kontrol, yörünge planlama, lider-takipçi algoritması, RRT algoritması

## 1. GENEL BİLGİLER

### 1.1. Giriş

Son yıllarda hızla gelişen havacılık ve otomasyon teknolojileri, insansız hava araçlarının (İHA) kullanım alanlarının genişlemesine öncülük etmiştir. İlk olarak askeri uygulamalarda kullanılmaya başlayan İHA'lar, bugün arama-kurtarma, tarımsal izleme, trafik gözlemleme ve yangınla mücadele gibi pek çok sivil alanda vazgeçilmez bir araç haline gelmiştir. İHA teknolojisinin bu kadar hızlı bir şekilde benimsenmesinde, düşük maliyetli olmaları, kullanım kolaylığı sunmaları ve teknolojik ilerlemelerle performanslarının sürekli iyileştirilmesi önemli bir rol oynamıştır. Özellikle birden fazla aracın birlikte çalışmasını gerektiren sürü uygulamaları, bu alandaki yeniliklerin merkezinde yer almaktadır.

Sürü uçuşlarının en büyük zorluklarından biri, yörünge planlamasını sağlamak diğeri ise araçlar arasındaki formasyonun etkin bir şekilde oluşturulması ve korunmasıdır.

Formasyon uçuşları, yalnızca görsellik ve düzen sağlamakla kalmaz; aynı zamanda enerji verimliliğini artırır ve sistem güvenliğini sağlar. Yıllar içerisinde bu problemi çözmek için behavior-based stratejiler, artificial-potential yaklaşımları, virtual structure stratejileri ve flocking algoritmalar gibi pek çok yöntem geliştirilmiştir.

Yörünge planlama problemine çözüm olarak grafik tabanlı yöntemler, örnekleme tabanlı yöntemler ve son zamanlarda popülerlik kazanan öğrenme tabanlı yöntemler geliştirilmiştir.

NASA, yakıt tasarrufu sağlamak amacıyla otonom formasyon uçuşu üzerine araştırmalar yapmıştır [1]. Bu projede, ticari uçakların V şeklinde formasyon uçuşu yaparak aerodinamik avantajlardan yararlanmaları hedeflenmiştir. Deneyler, lider uçağın arkasında belirli bir mesafede uçan takipçi uçakların yakıt tüketiminde %10'a varan tasarruf sağlayabileceğini göstermiştir.

ABD Savunma İleri Araştırma Projeleri Ajansı (DARPA), düşman ortamlarında İHA sürülerinin otonom bir şekilde iş birliği yapabilmesi için CODE [2] programını geliştirmiştir. Bu program, İHA'ların merkezi bir kontrol olmaksızın birlikte çalışarak görevleri yerine getirmelerini sağlamayı amaçlamaktadır.

Samaniego ve diğer araştırmacılar örneklemeyle dayalı İHA yol planlama tekniklerine dair bir çalışma sunmuştur [3]. Aynı araştırmacılar ayrıca bir uzayda kesin hücre ayrıştırması (ECD-PRM) ve modifiye edilmiş adaptif hücre ayrıştırması (MACD) olmak üzere iki yeni algoritma tanıtmışlardır [4].

Kang ve diğer araştırmacılar A\*, RRT ve PSO algoritmalarını tartışarak sonuçlarını dayanıklılığa bağlı olarak karşılaştırmışlardır. RRT algoritmasının A\* ve PSO algoritmalarından daha iyi sonuç ürettiğini savunmuşlardır. Ayrıca RRT algoritmasının basit durumlarda daha iyi performans gösterdiğini, A\* ve PSO algoritmalarının ise düşük varyans durumlarda daha iyi performans gösterdiğini kanıtlamışlardır [5].

Bu çalışmada ise lider-takipçi algoritması ve RRT algoritması beraber kullanılarak bir İHA sürüsünün belli bir formasyonu oluşturup, o formasyonu koruyarak belirlediği yörünge boyunca güvenli, minimum maliyetli ve verimli bir uçuş gerçekleştirmesi amaçlanmıştır.

Formasyon şekli olarak aerodinamik avantajları ve görsel düzen sağlama kabiliyeti nedeniyle "V" formasyonu tercih edilmiştir. Bu formasyon tipi, özellikle uzun mesafeli uçuşlarda enerji tasarrufu sağlama ve araçlar arasındaki iletişimi kolaylaştırma gibi özellikleriyle ön plana çıkmaktadır.

Çalışmada, 5 İHA'dan oluşan bir sürü kullanılarak farklı senaryolar simüle edilmiştir. Her bir senaryo için sürünün takip edeceği yörüngeler analiz edilmiştir. Bu çalışmanın temel amacı, araç filolarının bilinen ve bilinmeyen çevrelerde güvenli, verimli ve düşük maliyetli bir şekilde uçuşmasını sağlayacak bir yaklaşım geliştirmektir.

## 1.2. Problem Tanımı

İHA sürülerinin uçuşlarında çözülmesi gereken en önemli problemlerden biri yörünge planlamasıdır. Uçuş gerçekleştirilen çevrede yörünge planlaması yapılırken bilinen ve bilinmeyen çevrelerde optimum yörünge oluşturulması sağlanmalıdır. Bunun yanı sıra gerçek dünyada karşılaşılan pek çok beklenmeyen dinamik değişken vardır. Bunların başında engeller ve araçların birbiriyle çarpışma olasılıkları gelir. Engelden kaçarken ortaya formasyon şeklinin korunamaması sorunu da çıkar.

Bu sorunlara ortak bir çözüm ancak yörünge planlama ve formasyon koruma algoritmalarının entegre edilerek tek proje çatısı altında çözümlenmesiyle sağlanabilir.

## 2. YAPILAN ÇALIŞMALAR

### 2.1. Sistem Mimarisi

Bu çalışma insansız hava araçları sürüleri için RRT algoritmasını ve lider-takipçi algoritmasını kullanarak yörünge planlama ve formasyon kontrol sağlayan genişletilmiş bir çözüm sunar. Sürü içinde belirlenen bir lider için yörünge planlaması gerçekleştirilir ve takipçi İHA'lar liderin yörüngesine bağlı kendi yörüngelerini takip eder. Aynı zamanda takipçi İHA'lar, lider İHA'ya göre kendilerini konumlandırarak bir 'V' formasyonu oluştururlar. Bu çalışmanın amacı gerçek senaryolara bağlı kalıp yörünge planlama problemini simüle ederek çözüm üretmektir.

#### 2.1.2. Formasyon Kontrol

Formasyon kontrolü, birden fazla aracın belirli bir düzende hareket etmesini sağlamak için kullanılan kontrol stratejileridir. Araçlar arasındaki mesafe, yön ve hız gibi parametreler senkronize edilir. Araçlar arası çarpışma ve engellerden kaçınma sağlanır. Amaç araç grubunun bir bütün olarak belirli bir görevi yerine getirmesidir.

##### 2.1.2.1. Lider-Takipçi Algoritması

Lider takipçi mekanizması kolay kontrol yapısı ve ölçeklenebilirliği ile en yaygın ve üzerine en çok çalışılmış yöntemdir[6]. Bu yaklaşımda gruptaki bir araç lider olarak, diğer araçlar ise takipçi olarak atanır. Lider global bilgiye tam erişim hakkına sahiptir ve liderin yörüngesi takipçiler için bir referans oluşturur. Tüm takipçiler lider ile aralarındaki mesafeyi ölçebilir ve bu mesafeye göre formasyonda kendilerini konumlandırabilirler. Bu mekanizmada takipçilerden lidere bir geribesleme yapılmaz ve bu bir dezavantajdır.



Şekil 1. Lider takipçi mekanizmasının gösterimi



Huang ve ark. [7], ucuz, sensör sınırlı ve hesaplama açısından sınırlı robotların, uzun mesafeler boyunca istenen bir oluşumda bir lider robotu takip ettiği heterojen bir mobil robot grubunun kontrolünü ve lokalizasyonunu sunar. Önerilen yöntem, liderin genel olarak takipçilerle görüş hattı temasını sürdürmesi gerektiği için sınırlıdır.

Algoritmanın sözde kodu devamında verilmiştir.

```
// Ana Algoritma
1. Lider Dron (3. dron) için RRT* ile yol planlaması yap
- Başlangıç noktasından hedef noktaya yol bul
- Engellerden kaçın
- Yolu kaydet

2. Her Takipçi Dron için:
FOR i = 1 to n DO
  IF i ≠ 1 THEN
    // Liderin yolundaki her nokta için
    FOR j = 1 to leader_path_length DO
      // Liderin yönünü hesapla
      IF j < leader_path_length THEN
        leader_heading = atan2(leader_path[j+1].y - leader_path[j].y,
        leader_path[j+1].x - leader_path[j].x)
      ELSE
        leader_heading = atan2(leader_path[j].y - leader_path[j-1].y,
        leader_path[j].x - leader_path[j-1].x)
      END IF

      // Takipçinin liderden uzaklığını hesapla
      dis = d * |l-i|

      // Formasyon açısını belirle
      IF i < 1 THEN
        ang = alpha      // V'nin sağ tarafı
      ELSE
        ang = -alpha     // V'nin sol tarafı
      END IF

      // Takipçinin istenen pozisyonunu hesapla
      pd = leader_path[j] + dis * [cos(leader_heading + ang),
      sin(leader_heading + ang),
      0]

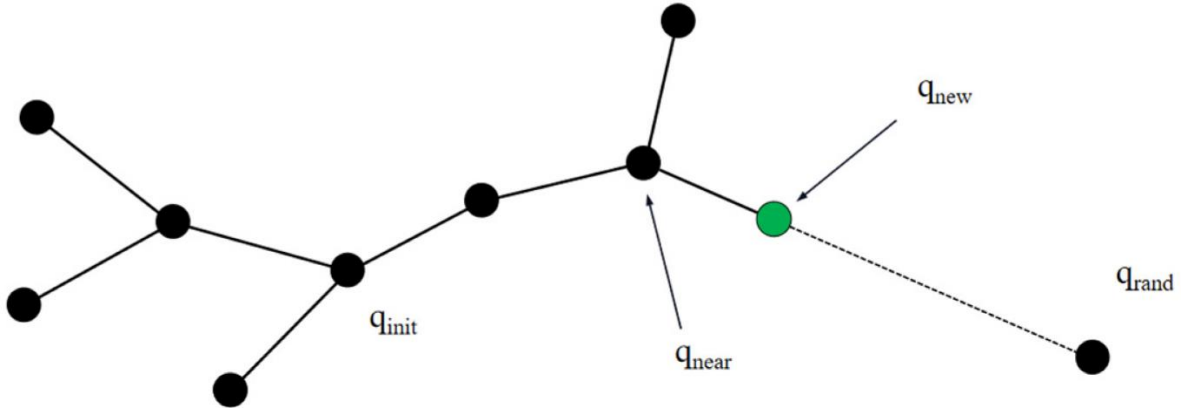
      // Takipçinin yolunu güncelle
      follower_path[j] = pd
    END FOR
  END IF
END FOR
```

### 2.1.3. Yörünge Planlama

Yörünge planlama, bir aracın ya da otonom sistemin başlangıç noktasından hedef noktaya ulaşması için uygun bir yol oluşturma sürecidir. Bu süreçte, robotun çevresel haritası, dinamik kısıtları ve belirli hedefleri dikkate alınarak en uygun rota planlanır.

#### 2.1.3.1. RRT Algoritması

RRT (Rapidly exploring Random Tree), başlangıç ve hedef noktalarının belirli olduğu, eşit rastgele örneklemeyle dayanan tek sorgulu bir ağaç yapısına sahip arama algoritmasıdır. RRT algoritması, bir başlangıç noktasından hedef noktayı arayan bir ağaç şeklinde yapı konfigürasyonlarına göre büyür; bu ağacın her bir düğümü, çalışma alanında bir noktayı temsil eder. Bu algoritma LaValle tarafından tanıtılmıştır [8]. RRT yöntemi, bir ağaç kökü olarak bir başlangıç noktası ( $q_{start}$ ) ile yol haritası konfigürasyonuna başlar. Sonra, algoritma, konfigürasyon alanındaki her bir sonraki iterasyon için rastgele bir nokta ( $q_{rand}$ ) seçer ve mevcut grafikten en yakın düğüm ( $q_{near}$ ) aranır.  $q_{near}$  ile  $q_{rand}$  arasında önceden tanımlanmış bir mesafe ( $\epsilon$ ), yani adım boyutu mesafesi ile yeni bir örnek ( $q_{new}$ ) üretilir. Yeni seçilen düğümde ( $q_{new}$ ) bir çarpışma kontrolü yapılır. Eğer bir çarpışma gerçekleşirse, yeni adım ( $q_{new}$ ) iptal edilir. Aksi takdirde, her yeni nokta için, yeni üretilen düğüm ile hedef nokta arasındaki mesafe dikkate alınacak şekilde arama ağacına eklenir. İşlem Şekil 2'de gösterilmiştir.



Şekil 2. Bir RRT yörünge diyagramı

RRT algoritması UAV'nin başlangıçtan hedefe güvenli bir yol bulmasını sağlar. Çarpışmalardan kaçınmak için kinematik kısıtlamaları hesaba katar. Özellikle dinamik tehditlerin olduğu ortamlarda kullanılır.

Algoritmanın sözde kodu devamında verilmiştir.

```
//RRT Algoritması
WHILE dronlar hareket ediyor DO
  FOR her dron DO
    IF herhangi bir engele çarpma riski varsa THEN
      // Engel kaçınma modunu başlat
      avoidance_mode = true

      // Engel kaçınma noktası hesapla
```

```

obs_to_pos = current_pos - obstacle_pos
perp_vector = cross(obs_to_pos, [0,0,1])
avoidance_point = obstacle_pos + (obs_to_pos + perp_vector) * safety_distance

// Geçici kaçınma yolu oluştur
temp_path = [current_pos, avoidance_point, next_pos]

// Yolu güncelle
path = [path[1:i-1], temp_path, path[i+1:end]]
END IF
END FOR
END WHILE

```

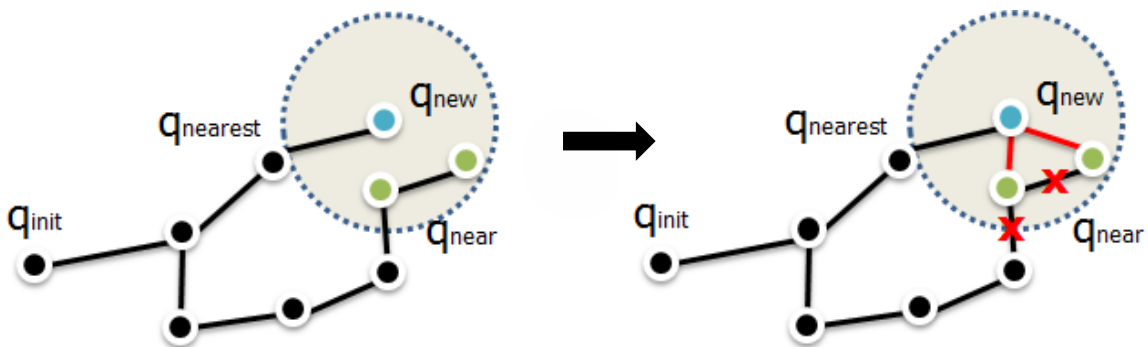
### 2.1.3.2. RRT\* Algoritması

RRT\* (Rapidly-exploring Random Tree Star), başlangıç ve hedef noktalarının belirli olduğu, eşit rastgele örneklemeye dayanan, en uygun yolu bulmayı hedefleyen tek sorgulu bir ağaç yapısına sahip arama algoritmasıdır. RRT\*'ın temel RRT'den farkı, daha iyi (daha kısa veya daha güvenli) yollar bulmak için ağacı optimize etmesidir. Bu algoritma, bir başlangıç noktasından hedef noktayı arayan bir ağaç şeklinde yapı konfigürasyonlarına göre büyür; bu ağacın her bir düğümü, çalışma alanında bir noktayı temsil eder.

RRT\* yöntemi, bir ağaç kökü olarak bir başlangıç noktası ( $q_{start}$ ) ile yol haritası konfigürasyonuna başlar. Ardından, algoritma, konfigürasyon alanındaki her bir sonraki iterasyon için rastgele bir nokta ( $q_{rand}$ ) seçer ve mevcut grafikten en yakın düğüm ( $q_{near}$ ) aranır.  $q_{near}$  ile  $q_{rand}$  arasında önceden tanımlanmış bir mesafe ( $\epsilon$ ), yani adım boyutu mesafesi ile yeni bir örnek ( $q_{new}$ ) üretilir.

Yeni seçilen düğümde ( $q_{new}$ ) bir engel çarpışma kontrolü yapılır. Eğer bir çarpışma gerçekleşirse, yeni adım ( $q_{new}$ ) iptal edilir. Aksi takdirde,  $q_{new}$ 'e belirli bir yarıçap içindeki komşu düğümler ( $Q_{near}$ ) aranır. Bu komşu düğümler arasından  $q_{new}$ 'e ulaşmak için en düşük maliyeti (örneğin, en kısa yolu) sunan düğüm,  $q_{new}$ 'in ebeveyni olarak seçilir. Daha sonra,  $q_{new}$  ağaca eklenir. Bu adım, RRT\*'ın en iyi ebeveyni seçme özelliğini oluşturur.

Ağaca yeni bir düğüm eklendikten sonra, RRT\* yeniden kablolama (rewiring) yapar. Bu,  $q_{new}$ 'in komşularından herhangi birinin,  $q_{new}$  aracılığıyla ağaçtaki mevcut ebeveynlerinden daha kısa bir yola sahip olup olmadığını kontrol etmeyi içerir. Eğer daha kısa bir yol bulunursa, ilgili komşu düğümün ebeveyni  $q_{new}$  olarak değiştirilir. Bu yeniden kablolama süreci, ağacın zamanla daha optimize edilmiş bir yapıya dönüşmesini sağlar ve daha verimli yolların bulunmasına yardımcı olur. Her yeni nokta için, yeni üretilen düğüm ile hedef nokta arasındaki mesafe dikkate alınacak şekilde arama ağacına eklenir. İşlem aşağıdaki gibi devam eder.



Şekil 3. Bir RRT\* yörünge diyagramı

RRT\* algoritması, İnsansız Hava Araçlarının (İHA) başlangıçtan hedefe güvenli ve optimum bir yol bulmasını sağlar. Çarpışmalardan kaçınmak için kinematik kısıtlamaları hesaba katar. Özellikle dinamik tehditlerin olduğu ortamlarda ve en kısa/güvenli yolu bulmanın kritik olduğu senaryolarda tercih edilir.

Algoritmanın sözde kodu devamında verilmiştir.

// RRT\* Algoritması

FUNCTION RRT\_Star\_Plan(start\_pos, goal\_pos, obstacles):

    nodes = {start\_pos}

    parent = {}

    cost = {start\_pos: 0}

    FOR iter = 1 TO max\_iter DO

        q\_rand = Generate\_Random\_Sample()

        q\_near = Find\_Nearest\_Node(q\_rand, nodes)

        q\_new = Steer(q\_near, q\_rand, step\_size)

        IF NOT Is\_Collision(q\_new, obstacles) THEN

            Q\_near = Find\_Nodes\_In\_Radius(q\_new, nodes, search\_radius) // RRT\*

özelligi

        // En iyi ebeveyni seç (ChooseParent)

        q\_min\_parent = q\_near

        min\_cost = cost[q\_near] + Calculate\_Edge\_Cost(q\_near, q\_new)

        FOR q\_neighbor IN Q\_near DO

            IF NOT Is\_Collision(q\_neighbor, q\_new) AND cost[q\_neighbor] +

Calculate\_Edge\_Cost(q\_neighbor, q\_new) < min\_cost THEN

                min\_cost = cost[q\_neighbor] +

Calculate\_Edge\_Cost(q\_neighbor, q\_new)

                q\_min\_parent = q\_neighbor

            END IF

        END FOR

        Add\_Node(q\_new, q\_min\_parent)

        cost[q\_new] = min\_cost

    // Yeniden Kablolama (Rewire)

    FOR q\_neighbor IN Q\_near DO

        IF q\_neighbor != q\_min\_parent AND NOT Is\_Collision(q\_new, q\_neighbor) AND cost[q\_new] + Calculate\_Edge\_Cost(q\_new, q\_neighbor) < cost[q\_neighbor] THEN

            Update\_Parent(q\_neighbor, q\_new)

            cost[q\_neighbor] = cost[q\_new] + Calculate\_Edge\_Cost(q\_new, q\_neighbor)

        END IF

    END FOR

```

    IF Distance(q_new, goal_pos) < goal_tolerance THEN
    // Hedefe ulaşıldı, yolu çıkar
    RETURN Reconstruct_Path(q_new, parent)
    END IF
  END IF
END FOR
RETURN [] // Yol bulunamadı

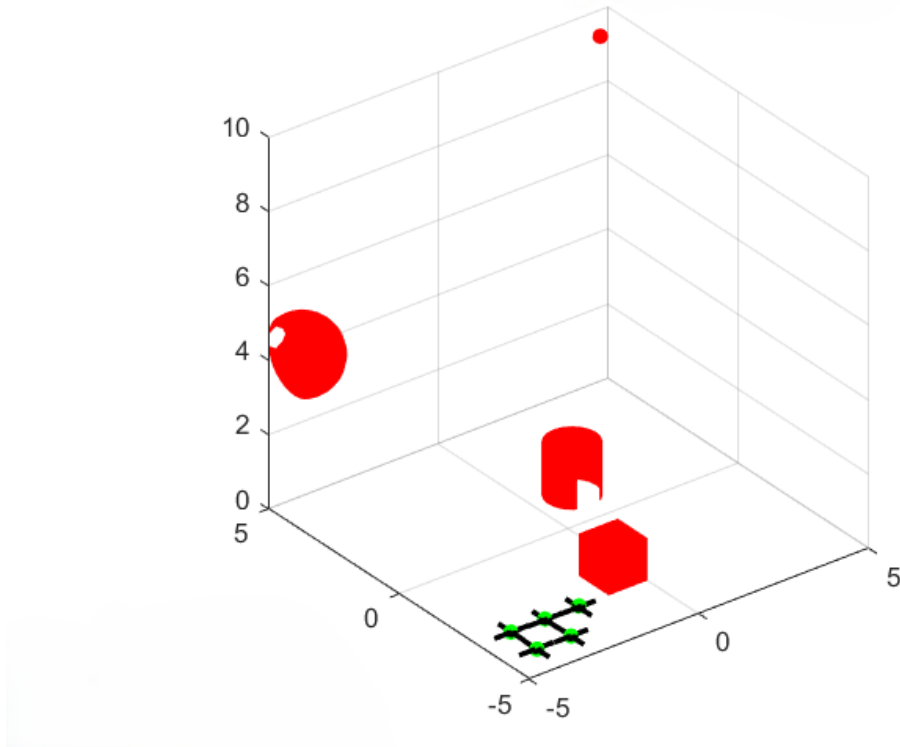
```

## 2.2. Sistem İcrası

Projenin çalışması temel olarak 3 adımdan oluşur. Bunlar başlangıç adımı, formasyon kurma ve yörünge planlama adımı, yörünge takibi ve formasyon koruma adımı olarak sıralanabilir.

### 2.2.1. Başlangıç Adımı

Bu adım, simülasyonun başlamadan önceki tüm ön hazırlıklarını ve sistem parametrelerinin tanımlanmasını içerir. Temelde, simülasyon ortamının ve İHA'ların hazır hale getirildiği kısımdır. Parametre tanımlamaları, grafik ortamının oluşturulması ve simülasyon için gerekli tüm statik ve dinamik nesnelerin tanımlanması bu adımda gerçekleşir. Görselleştirme ortamı hazır hale getirilir.



Şekil 4. Sistemin 1. Adımı

### 2.2.2. Formasyon Kurma ve Yörünge Planlama Adımı

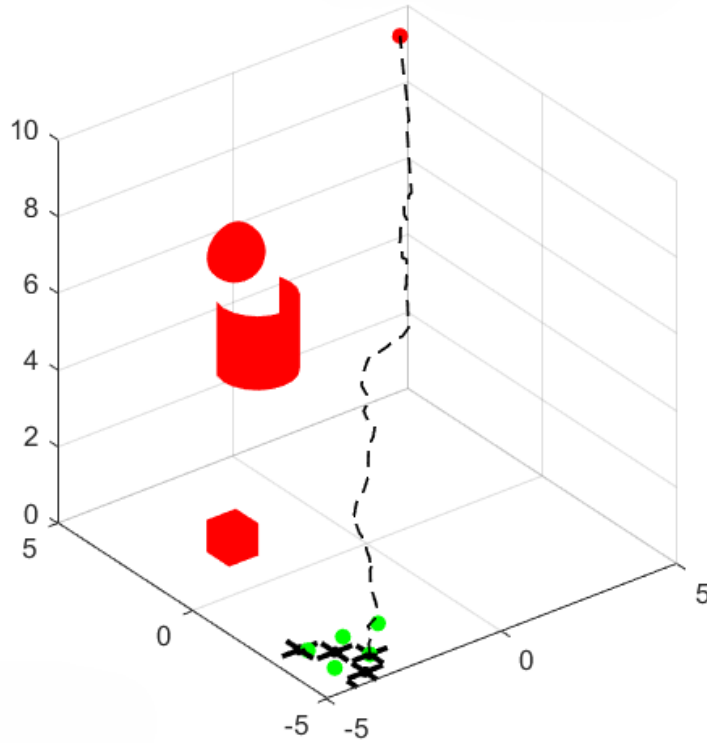
Bu adım, İHA'ların başlangıç konumlarından hedef formasyonlarına geçişini ve eş zamanlı olarak tüm İHA'lar için güvenli ve çarpışmasız yörüngelerin planlanmasını içerir. Bu iki süreç burada iç içe yürütülür.

Takipçi İHA'lar, başlangıç konumlarından lider İHA etrafındaki hedef formasyon pozisyonlarına doğru hareket eder. Lider İHA bu aşamada başlangıç konumunda sabit kalır. Takipçi İHA'lar, mesafe ve açı parametrelerine göre lidere göre göreceli hedef pozisyonlarını hesaplar. İHA'lar başlangıç konumlarından hedef formasyon pozisyonlarına doğru doğrusal olarak enterpolasyonla hareket ettirilir. Bu, formasyonun yavaşça ve kontrollü bir şekilde oluşmasını sağlar.

Formasyon kurma aşamasının belirli bir noktasından, lider İHA için RRT\* algoritması kullanılarak yol planlaması başlatılır. RRT\* algoritması, engellerden kaçınarak ve maliyeti optimize ederek lider İHA için başlangıç noktasından hedef noktasına kadar bir yol bulmaya çalışır. Eğer lider için geçerli bir yol bulunursa, Bu yol lider İHA'nın yolu olarak atanır. Diğer takipçi İHA'lar için yollar, liderin yoluna göre göreceli pozisyonlar korunarak hesaplanır. Her bir yol noktasında liderin başlığı hesaplanır ve bu başlık kullanılarak takipçilerin formasyon içindeki ideal pozisyonları belirlenir. Bu, formasyonun hareket halindeyken de korunmasını sağlar. Planlanan yol, kesikli çizgi ile grafiğe çizilir, böylece tüm İHA'ların gideceği rota önceden görselleştirilir.

Her formasyon adımı ve yol planlama kontrolü sonrasında, İHA'ların yeni konumları güncellenir ve çizilir. Bu, simülasyonun hareketli bir animasyonunu oluşturur.

Bu adım, İHA'ların formasyonlarını oluştururken aynı zamanda gelecekteki rotalarını dinamik engelleri de dikkate alarak belirlemelerini sağlar. Liderin yolu planlandıktan sonra, takipçilerin yolları formasyon geometrisine göre türetilir.



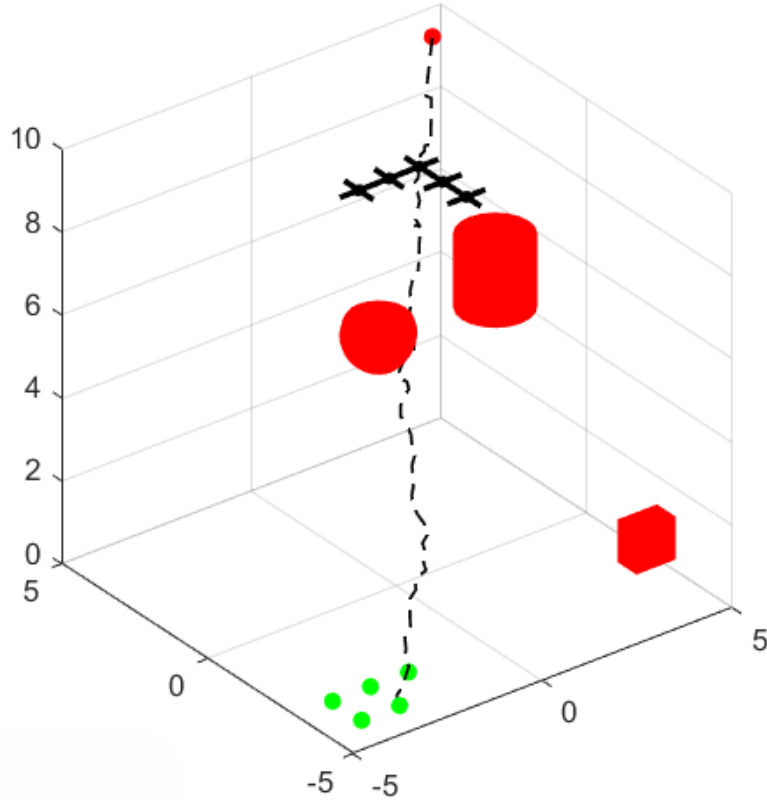
Şekil 5. Sistemin 2. Adımı

### 2.2.3. Yörünge Takibi ve Formasyon Koruma Adımı

Bu adım, formasyon kurulduktan ve yörüngeler planlandıktan sonra İHA'ların planlanan yörüngeleri takip etmesini ve formasyon yapısını korurken engellerden dinamik olarak kaçınmasını içerir.

İHA'lar planlanan en uzun yörüngenin sonuna ulaşana kadar hareket devam eder. Her İHA'nın bir sonraki adım pozisyonu için çarpışma kontrolü yapılır. İHA'nın bir sonraki adımı ile mevcut engeller arasındaki mesafe eşik değerden küçükse çarpışma algılanır. Eğer bir çarpışma algılanırsa ve İHA'lar zaten kaçınma modunda değilse, Her bir İHA için, çarpışma noktasına en yakın engele göre bir kaçınma yolu dinamik olarak hesaplanır. Bu yol, engelden uzaklaşacak ve ardından orijinal yörüngeye geri dönecek geçici bir segment içerir. İHA'nın mevcut konumu, kaçınma noktası ve orijinal yolun bir sonraki noktası arasına geçici bir yol eklenerek güncellenir. Eğer çarpışma yoksa ve İHA'lar kaçınma modundaysa, İHA'ların yolları, kaçınma öncesindeki orijinal yollarına geri döndürülür. Bu, İHA'ların engelden kaçtıktan sonra planlanan rotalarına devam etmelerini sağlar. Her İHA'nın mevcut konumu, planlanan yoldaki bir sonraki noktaya göre güncellenir. Ardından İHA modelleri grafikte bu yeni konumlarda yeniden çizilir.

Bu adım, İHA'ların planlanan rotayı takip ederken, hareketli engeller gibi çevresel dinamik değişikliklere ve çarpışma riskleri gibi beklenmedik durumlara gerçek zamanlı olarak tepki vermesini ve formasyon disiplinini korumasını sağlar. Dinamik kaçınma mekanizması, sistemin otonom ve güvenli çalışması için hayati öneme sahiptir.



Şekil 6. Sistemin 3. Adımı

Temel olarak sistem önce statik bir formasyon oluşturmayı, ardından bu formasyonu koruyarak dinamik engellerden kaçınan akıllı bir yol planlaması yapmayı ve son olarak bu planlanan yolları takip ederken anlık çarpışma risklerine karşı proaktif kaçınma stratejileri uygulamayı hedeflemektedir. Bu, otonom çoklu İHA sistemlerinin karmaşık doğasını oldukça iyi bir şekilde simüle etmektedir.

### 3. SONUÇLAR

Bu bitirme projesi kapsamında, çoklu insansız hava araçlarının formasyon kontrolü ve yörünge planlaması entegre edilerek, lider-takipçi algoritması ve RRT algoritması temelinde kapsamlı bir çözüm sunulmuştur. Projede, İHA sürülerinin belirli bir formasyonu oluşturup koruyarak, dinamik ve bilinmeyen çevrelerde güvenli, minimum maliyetli ve verimli bir uçuş gerçekleştirmesi amaçlanmıştır.

Elde edilen temel sonuçlar aşağıdaki gibidir:

#### 3.1. Entegre Sistem Başarımı

Lider-takipçi algoritması ve RRT algoritmasının entegrasyonu sayesinde, dağıtık haldeki İHA sürüsünün formasyon oluşturma ve belirlenen yörünge boyunca hedefe güvenli bir şekilde ilerleme kabiliyeti başarılı bir şekilde simüle edilmiştir.

#### 3.2. Formasyon Kontrolü ve Korunması

Sistem, başlangıçta rastgele konumlanmış İHA'ların "V" formasyonu oluşturmalarını ve bu formasyonu uçuş süresince etkin bir şekilde korumasını sağlamıştır. Lider İHA'nın yolu referans alınarak, takipçi İHA'lar formasyon geometrisine uygun konumlarını koruyabilmiştir. Bu formasyonun, aerodinamik avantajlar ve enerji tasarrufu gibi özellikleriyle uzun mesafeli uçuşlarda potansiyel faydaları vurgulanmıştır.

#### 3.3. Dinamik Engelden Kaçınma Yeteneği

Proje, özellikle dinamik tehditlerin olduğu ortamlarda kullanılabilen RRT\* algoritmasının dinamik engel kaçınma mekanizmasını içermektedir. Yörünge üzerinde dinamik bir engelle karşılaşıldığında, formasyonun hareketini durdurarak veya dinamik olarak kaçınma yolu hesaplayarak güvenliği sağladığı gösterilmiştir. Bu, sistemin otonom ve güvenli çalışması için hayati öneme sahiptir.

#### 3.4. Gerçekçi Senaryo Simülasyonu

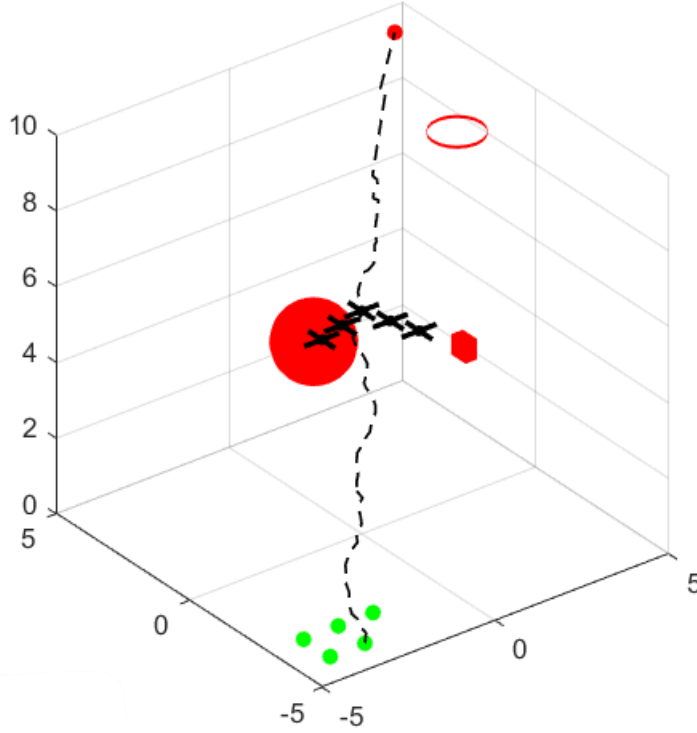
Çalışmada, 5 İHA'dan oluşan bir sürü kullanılarak farklı senaryolar simüle edilmiş ve gerçek dünya koşullarını yansıtan bir yörünge planlama problemi çözülmeye çalışılmıştır. Özellikle engeller ve araçların birbiriyle çarpışma olasılıkları gibi önemli gerçekçi kısıtlar dikkate alınmıştır.

#### 3.5. Problemin Bütünsel Çözümü

İHA sürülerinin uçuşlarındaki temel problemlerden olan yörünge planlaması ve formasyonun etkin bir şekilde oluşturulması ve korunması sorunlarına ortak bir çözüm sunulmuştur. Sistem, formasyon koruma ve yörünge planlama algoritmalarını tek bir proje



çatısı altında çözümleyerek İHA filolarının bilinmeyen çevrelerde güvenli ve verimli uçuşunu sağlama hedefine ulaşmıştır.



**Şekil 7.** Sistemin Bütünsel Çözümünün Temsili

### 3.6. Optimizasyon Yaklaşımı

Her ne kadar sistem dinamik engeller için tasarlanmış olsa da, RRT\*'ın en kısa/güvenli yolu bulma yeteneği sayesinde yörünge maliyet optimizasyonu hedeflenmiştir. Ayrıca, bilinen çevrelerde optimum yörünge seçilerek yakıt ve zamandan tasarruf edilmesi gibi ekonomik kısıtlar da göz önünde bulundurulmuştur.

## 4. ÖNERİLER

Sistemin 2 ana geliştirilebilir kısmı engelden kaçma ve formasyon esnekliği konularıdır. Sistem mevcut haliyle dinamik engellerin olduğu çevrede güvenli bir uçuş sağlayabilse de engelden kaçınma görevini tam olarak gerçekleştiremediği için statik engellerin olduğu ortamda güvenli bir uçuş sağlayamaz. Sistemde formasyon da çok katı kurallarla sabittir. Formasyon şekli ve açısı değişmez. Ama gerçek hayat senaryolarında formasyonun daha esnek olarak korunması gerektiği durumlar vardır.

### 4.1. Statik Engelli Ortamda Rota Planlama için Geliştirme Önerileri

Mevcut multi-drone simülasyon sistemdeki dinamik engeller yerine statik engellerin bulunduğu bir ortamda da verimli bir şekilde çalışır hale getirmek için bazı önemli geliştirmeler yapılabilir. Dinamik engellerle başa çıkmak için tasarlanmış mevcut RRT\*

algoritması, statik engeller için de uyarlanabilir, ancak statik engellerin doğası gereği daha gelişmiş ve önceden hesaplanmış rotalar elde etmek mümkündür.

Mevcut RRT\* algoritması dinamik ortamlar için uygun olsa da, statik engelli ortamlarda genellikle daha optimize edilmiş ve global olarak daha iyi çözümler sunan algoritmalar tercih edilebilir.

Kavraki'nin de üzerinde çalıştığı gibi [8] olasılıksal yol haritası (PRM) algoritması sisteme entegre edilebilir. Ortamda rastgele noktalar örnekleyerek ve bu noktalar arasında engelsiz bağlantılar kurarak bir grafik oluşturur. Daha sonra başlangıç ve hedef noktaları bu grafiğe bağlanır ve en kısa yol Dijkstra veya A\* gibi algoritmalarla bulunur. RRT\*'a benzer şekilde olasılıksal bir yaklaşımdır ancak genellikle daha yoğun bir grafik oluşturarak daha iyi çözümler bulabilir. Özellikle çok sayıda statik engelin olduğu kompleks ortamlarda etkilidir.

Lozano-Pérez'in de üzerinde çalıştığı gibi [9] görünürlük grafi algoritması mevcut sisteme entegre edilebilir. Görünürlük grafi, tüm engellerin köşelerini ve başlangıç/hedef noktalarını düğüm olarak kabul eder ve bu düğümler arasında engelsiz doğru hatları kenar olarak ekleyerek bir grafik oluşturur. Bu grafik üzerinde A\* gibi bir arama algoritması çalıştırılarak global olarak en kısa yol bulunur. Bu yöntem, engellerin şekilleri bilindiğinde en iyi yolları garantiler.

Daha optimize edilmiş algoritmalar kullanmanın yanı sıra formasyon lideri ve takipçiler için ayrı yörünge planlama stratejileri kullanmak da bir çözüm olabilir.

Khatib bir çalışmada [10] engelleri itici alanlar, hedefleri çekici alanlar olarak tanımlar ve İHA'nın bu alanların toplam gradient'ini takip ederek ilerlemesini sağlar. Buna benzer bir yaklaşımla lider İHA için global bir yol planlaması yapan ve takipçiler için lideri takip eden ancak yerel engellerden kaçınabilen bir mekanizma oluşturulabilir. Mevcut sistemde liderin yolu bulunduktan sonra takipçilerin yolları lidere göre offsetli olarak belirleniyor. Statik engeller söz konusu olduğunda, bu offsetli yolların her zaman engelsiz olacağının garantisi yoktur. Lider İHA için A\* veya PRM gibi bir algoritma ile en ideal yol bulunur. Takipçi İHA'lar, liderin yolunu takip ederken, kendi yakın çevrelerindeki statik engelleri algılayıp, liderin yörüngesinden çok sapmadan anlık olarak kaçınma manevraları yapabilirler. Bu, liderin yolunu bozmadan formasyonun devamlılığını sağlar. Bu yerel kaçınma için Potential Field (Potansiyel Alan) metodları kullanılabilir.

#### 4.2. Dinamik Formasyon Şekli ve Açısı için Geliştirme Önerileri

İHA'ların formasyon açısını ve şeklini çalışma anında dinamik olarak değiştirebilmesi, sistemin adaptasyon kabiliyetini ve çeşitli görev senaryolarına uygunluğunu önemli ölçüde artırır. Bu, yalnızca çevresel koşullara değil, aynı zamanda görevin gerekliliklerine göre de formasyonun optimize edilmesine olanak tanır. Ancak mevcut sistemde bu formasyon esnekliği sağlanamaz. Bu sorunu ortadan kaldırmak için formasyon optimizasyonu geliştirmeleri yapılabilir.

İHA'lar dar bir koridordan veya geçitten geçerken, V-formasyonu yerine tek bir çizgi halinde dizilmeleri daha uygun olabilir. Bu, çarpışma riskini azaltır ve geçişi kolaylaştırır. Geniş bir alanı taramak veya gözlemlemek gerektiğinde, daha geniş bir alpha açısı veya farklı bir küme formasyonu seçilebilir. Eğer belirli bir bölgedeki engel yoğunluğu artarsa, İHA'lar birbirlerine daha yakınlaşarak veya daha esnek bir formasyona geçerek manevra kabiliyetlerini artırabilirler.

Mevcut sistemde formasyon parametreleri, açı ve uzaklıklar, sabittir. Bunları çalışma anında değiştirilebilir hale getirmek için, sistemin belirli kriterlere göre karar verebilmesini sağlayacak bir formasyon adaptasyon katmanı eklenmesi gerekir. Sistemin çevresel koşulları algılayarak formasyon şekline karar vermesi temel adımdır. Simülasyon ortamındaki engel bilgilerini kullanarak çevresel koşulları değerlendiren ve buna göre en uygun formasyon

tipini (V-şekli, çizgi, küme vb.) veya açısını seçen bir modül eklenebilir. Örneğin Kavraki bir çalışmasında [11] engellerden kaçınırken formasyonları korumak için reaktif kontrol stratejilerine odaklanır. J. Wang ise bir çalışmasında [12] engelleri algılayarak formasyonun dinamik olarak ayarlanmasını incelemektedir.

Formasyon değişimi sadece engellere tepki olarak değil, aynı zamanda görevin o anki aşamasına veya gereksinimine göre de yapılabilir. Bunu sağlamak için sisteme görev planına dayalı olarak formasyon tipini veya parametrelerini dinamik olarak değiştiren bir görev yöneticisi modülü entegre edilmelidir. J. Ren bir çalışmasına [13] bağlantılı olarak çoklu ajan sistemlerinde kooperatif kontrol ve görev paylaşımı üzerine genel bir bakış sunar.

Daha karmaşık bir yaklaşım olarak, sistemin formasyon parametrelerini bir optimizasyon problemi olarak ele alması sağlanabilir. Belirli bir hedef fonksiyonunu, enerji tüketimi, çarpışma olasılığı veya görev tamamlama süresi, minimize veya maksimize eden formasyon parametrelerini dinamik olarak belirleyen bir optimizasyon algoritması kullanılabilir. Bu yaklaşım, anlık çevresel durum ve görev hedeflerine göre en uygun formasyon parametrelerini bulmak için bir maliyet fonksiyonu tanımlar. Maliyet fonksiyonu; engellere olan yakınlık, formasyon içindeki gerilim, İHA'ların birbirlerinden çok uzaklaşmaya çalışması, enerji tüketimi, veya görev ilerlemesi gibi faktörleri içerebilir. Bu optimizasyon, her simülasyon adımı veya belirli aralıklarla çalıştırılabilir. Ancak gerçek zamanlı performans için hesaplama maliyeti düşük algoritmalar tercih edilmelidir.

Bu geliştirmeler uygulanırken, her bir İHA'nın dinamik olarak değişen formasyon hedeflerine sorunsuz bir şekilde adapte olmasını sağlayacak kontrol algoritmalarının da bu değişikliklere tepki verebildiğinden emin olunmalıdır. Dinamik formasyon değişimi, İHA'lar arasındaki iletişim ve koordinasyonu daha da kritik hale getirecektir. Bu nedenle, olası iletişim gecikmeleri ve hata toleransı konularını da göz önünde bulundurmak önemlidir.

## 5. KAYNAKLAR

- Yang, Y.; Xiong, X.; Yan, Y. UAVFormation Trajectory Planning Algorithms: A Review. *Drones* 2023, 7, 62. <https://doi.org/10.3390/drones7010062>
- H. T. Do, H. T. Hua, M. T. Nguyen, et al., "Formation control algorithms for multiple-UAVs: A comprehensive survey," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 8, no. 27, pp. 1–13, 2021, Number: 27.
- Y. Zhang and H. Mehrjerdi, "A survey on multiple unmanned vehicles formation control and coordination: Normal and fault situations," 2013 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 2013, pp. 1087-1096, doi: 10.1109/ICUAS.2013.6564798.
- Chen, Q.; Wang, Y.; Jin, Y.; Wang, T.; Nie, X.; Yan, T. A Survey of An Intelligent Multi Agent Formation Control. *Appl. Sci.* 2023, 13, 5934.
- [1] National Aeronautics and Space Administration (NASA). 2008. Formation Flight as a Cooperative Game. NASA Technical Reports Server. Accessed January 15, 2025.
- [2] Defense Advanced Research Projects Agency (DARPA). Collaborative Operations in Denied Environment (CODE). Accessed January 15, 2025.
- [3] F. Samaniego, J. Sanchis, S. García-Nieto, R. Simarro, Uav motion planning and obstacle avoidance based on adaptive 3d cell decomposition: Continuous space vs discrete space, in: Ecuador Technical Chapters Meeting (ETCM), 2017 IEEE, IEEE, 2017, pp. 1–6.
- [4] R. Siegwart, I.R. Nourbakhsh, D. Scaramuzza, *Autonomous mobile robots*, Massachusetts Institute of Technology.
- [5] M. Kang, Y. Liu, Y. Ren, Y. Zhao, Z. Zheng, An empirical study on robustness of uav path planning algorithms considering position uncertainty, in: *Intelligent Systems and Knowledge Engineering (ISKE)*, 2017 12th International Conference on, IEEE, 2017, pp. 1–6.
- [6] V. Roldão, R. Cunha, D. Cabecinhas, C. Silvestre, and P. Oliveira, "A leader-following trajectory generator with application to quadrotor formation flight," *Robotics and Autonomous Systems*, vol. 62, no. 10, pp. 1597–1609, 2014.
- [7] J. Huang, S. M. Farritor, A. Qadi, and S. Goddard, "Localization and follow-the-leader control of a heterogeneous group of mobile robots," *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 2, pp. 205–215, 2006.
- [8] Kavraki, L. E., Svestka, P., Overmars, M. H., & Latombe, J. C. (1996). "Probabilistic roadmaps for path planning in high-dimensional configuration spaces." *IEEE Transactions on Robotics and Automation*, 12(3), 566-580.
- [9] Lozano-Pérez, T., & Wesley, M. A. (1979). "An algorithm for planning collision-free paths among polyhedral obstacles." *Communications of the ACM*, 22(10), 560-570.
- [10] Khatib, O. (1986). "Real-time obstacle avoidance for manipulators and mobile robots." *The International Journal of Robotics Research*, 5(1), 90-98.
- [11] "Decentralized Reactive Control of Multi-Robot Formations" by S. Singh, L. E. Kavraki. *IEEE Robotics and Automation Magazine*, 2004.
- [12] "Adaptive Formation Control for Multi-UAV Systems with Obstacle Avoidance" by J. Wang, G. Lin, H. Li. *Journal of Intelligent & Robotic Systems*, 2018.
- [13] "Cooperative Control of Multi-Agent Systems: A Survey" by J. Ren, K. J. Zhou, Y. C. Soh. *Automatica*, 2010.

## 6. EKLER

[Tez Teslim Koşulları](#)

[Proje Planı](#)

[Çalıştay](#)

### STANDARTLAR ve KISITLAR FORMU

Projenin hazırlanmasında uyulan standart ve kısıtlarla ilgili olarak, aşağıdaki soruları cevaplayınız.

1. Projenizin tasarım boyutu nedir? (Yeni bir proje midir? Var olan bir projenin tekrarı mıdır? Bir projenin parçası mıdır? Sizin tasarımınız proje toplamının yüzde olarak ne kadarını oluşturmaktadır?)

Proje var olan bir projedir. Benim tasarımım projenin çözümünde farklı algoritmalar kullanılarak optimizasyon sağlamaya yöneliktir.

2. Projenizde bir mühendislik problemini kendiniz formüle edip, çözdünüz mü? Açıklayınız.

Bu projede bilinen çevrelerde uçuş yapılırken beklenmedik engellerle karşılaşılan durumlarda hangi yörüngelerin izlenmesi gerektiği problemi çözmeye çalıştım.

3. Önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?

Nesne yönelimli programlama, paralel programlama

4. Kullandığınız veya dikkate aldığınız mühendislik standartları nelerdir? (Proje konunuzla ilgili olarak kullandığınız ve kullanılması gereken standartları burada kod ve isimleri ile sıralayınız).

ISO 22139  
RTCA DO-178C  
ISO 8373  
ASTM F3201

5. Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir? Lütfen boşlukları uygun yanıtlarla doldurunuz.

## a) Ekonomi

İnsansız hava araçlarının bilinen çevrelerde uçuşlarında optimum yörüngeyi seçerek yakıt ve zamandan tasarruf edilmesine dikkat edildi.

## b) Üretilbilirlik:

Bir sorunun var olan çözümlerinden hariç farklı yöntemlerle optimum yöntemlerin üretilmesi amaçlanmıştır.

## c) Etik:

Projede geliştirilen teknolojiler, etik olmayan veya saldırgan askeri operasyonlarda kullanılmamalıdır.  
Teknolojinin, afet yönetimi, arama-kurtarma veya çevre koruma gibi insani ve barışçıl amaçlara hizmet etmelidir.

## d) Güvenlik:

İHA'ların birbirleriyle ve çevredeki engellerle çarpışmaması için, minimum mesafelerin korunması zorunludur.