İNSANSIZ HAVA ARAÇLARINDA KÜMELENME VE SÜRÜ KONTROLÜ

Batuhan KARAÇAL

2022
YÜKSEK LİSANS TEZİ
MEKATRONİK MÜHENDİSLİĞİ

Tez Danışmanı
Doç. Dr. Can Bülent FİDAN

İNSANSIZ HAVA ARAÇLARINDA KÜMELENME VE SÜRÜ KONTROLÜ

Batuhan KARAÇAL

T.C.
Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Mekatronik Mühendisliği Anabilim Dalında
Yüksek Lisans Tezi
Olarak Hazırlanmıştır

Tez Danışmanı
Doç. Dr. Can Bülent FİDAN

KARABÜK
Mart 2022

Batuhan KARAÇAL tarafından hazırlanan "İNSANSIZ HAVA ARAÇLARINDA KÜMELENME VE SÜRÜ KONTROLÜ" başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Doç. Dr. Can Bülent FİDAN .....
Tez Danışmanı, Mekatronik Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından Oy Birliği ile Mekatronik Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 11/03/2022

Ünvanı, Adı SOYADI (Kurumu)	İmzası
Başkan : Doç. Dr. Can Bülent FİDAN (KBÜ)	.....
Üye : Prof. Dr. Mehmet KARALI (Necmettin E.Ü)	.....
Üye : Dr. Öğr. Üyesi Cihan MIZRAK (KBÜ)	.....

KBÜ Lisansüstü Eğitim Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Prof. Dr. Hasan SOLMAZ .....
Lisansüstü Eğitim Enstitüsü Müdürü

"Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim."

Batuhan KARAÇAL

ÖZET

Yüksek Lisans Tezi

İNSANSIZ HAVA ARAÇLARINDA KÜMELENME VE SÜRÜ KONTROLÜ

Batuhan KARAÇAL

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Mekatronik Mühendisliği Anabilim Dalı

Tez Danışmanı:
Doç. Dr. Can Bülent FİDAN
Mart 2022, 119 sayfa

Bu çalışmada, insansız hava araçlarında sürü kontrolü ve kümelenmesi incelenerek, sürü davranışının matematiksel modeli için Parçacık Sürü Optimizasyonu (PSO), formasyon kontrol işlemi için PD tabanlı kontrol algoritması ve sürü İHA'lar arası çarpışma engelleme işleminde potansiyel fonksiyon kullanılmıştır. Diğer sürü kontrol algoritmaları, Kuş Sürüsü (BSA), Yapay Arı Kolonisi (ABA), Yarasa Algoritması (BA) ve Ateş Böceği (FA) algoritmaları modellenerek simülasyon ortamında minimum arama performansı karşılaştırılmıştır. Merkezi kontrol işlemleri için özgün yazılım çerçevesi geliştirilmiştir. Sürü üyelerinin merkezi kontrol olmadan lider tarafından kontrolü için algoritma geliştirilmiştir. Sürü liderinin belirlenmesi işleminde tek katmanlı yapay sinir ağı modeli oluşturularak sürü merkezine ve görev gereksinimlerine uygun lider atama işlemi gerçekleştirilmiştir. Her sürü bireyinin diğer sürü bireylerinin bilgilerine ulaşması için, oluşturulan Wi-Fi ağına, yer istasyonundan telemetri (konum, hız vb.) verilerini göndererek merkezi veya lider tarafından kontrolü sağlanmış ve konum eniyileme işlemi gerçekleştirilmiştir.

Geometrik şekillerin (çizgi, üçgen, kare, beşgen, hilal vb.) formasyon noktaları oluşturulmuştur. Oluşturulan geometrik formasyon noktalarına hangi bireyin gideceğini belirlemek için eş zamanlı atama algoritması kullanılmıştır. Modellerin MATLAB ile analizleri yapılarak, GAZEBO simülasyon ortamında döner kanat insansız hava araçlarının modellenmesi ve uçuş testleri ile görevlerin doğruluğu sağlanmıştır. Çalışma da askeri ve sivil alanlarda sürü halinde görev icra edilerek, askeri alanlarda üstünlük, sivil alanlarda avantaj ve kolaylık elde edilmesine olanak sağlanacaktır.

Anahtar Sözcükler : İnsansız hava araçları, sürü kontrol, kümelenme, formasyon kontrol.

Bilim Kodu **:** 92905

ABSTRACT

M. Sc. Thesis

CLUSTERING AND SWARM CONTROL IN UNMANNED AERIAL VEHICLE

Batuhan KARAÇAL

Karabük University
Institute of Graduate Programs
Department of Mechatronics Engineering

Thesis Advisor:
Assoc. Prof. Dr. Can Bülent FİDAN
March 2022, 119 pages

In this paper, swarm control and clustering for unmanned aerial vehicles are investigated. Particle swarm optimisation (PSO) is used for the mathematical model of swarm behaviour, the PD -based control algorithm is used for the formation control process, and the potential function is used for the collision avoidance process between swarm UAVs. Other swarm control algorithms, Bird Swarm (BSA), Artificial Bee Colony (ABA), Bat Algorithm (BA) and Firefly (FA) were modelled and the minimum search performance was compared in the simulation environment. A unique software framework for central control operations was developed. An algorithm was developed for leader control of herd members without central control. In determining the herd leader, a single-layer artificial neural network model was created and the process of assigning the leader was carried out according to the herd centre and task requirements. In order for each individual of the herd to reach the information of the other herd members, the created Wi-Fi network was controlled by the control centre or the leader

by sending telemetry data (position, speed, etc.) from the ground station and performing the location optimisation process.

Formation points with geometric shapes (line, triangle, square, pentagon, crescent, etc.) were created. The simultaneous assignment algorithm was used to determine which person would go to the created geometric formation points. Analysis of the models using MATLAB, modelling of unmanned rotorcraft and flight tests in the simulation environment GAZEBO ensured the accuracy of the missions. In the study, it will be possible to achieve superiority in military domains and advantage and convenience in civil domains by performing tasks in a swarm in military and civil domains.

Key Word : Unmanned aerial vehicle, swarm control,clustering, formation control.

Science Code : 92905

TEŞEKKÜR

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın hocam Doç. Dr. Can Bülent FİDAN'a sonsuz teşekkürlerimi sunarım.

Çalışmalarım sırasında TÜBİTAK 2210/C Yurt İçi Öncelikli Alanlar Yüksek Lisans Burs Programı ile sağladığı destekten dolayı bilimin ve bilim insanının destekçisi olan TÜBİTAK'a teşekkür ederim.

Karabük Üniversitesi Bilimsel Araştırma Projeleri (BAP) tarafından desteklenen FYL-2020-2367 proje numaralı İNSANSIZ HAVA ARAÇLARINDA KÜMELENME VE SÜRÜ KONTROLÜ isimli bu çalışma için desteklerinden dolayı teşekkür ederim.

Sevgili aileme manevi hiçbir yardımı esirgemeden yanımda oldukları için tüm kalbimle teşekkür ederim.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

Sayfa

Sayfa

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

Sayfa

SİMGELER VE KISALTMALAR DİZİNİ

SİMGELER

ϕ : yuvarlanma açısı

ψ : sapma açısı

θ_i : i. İHA'nın baş açısı

C_1 : kişisel öğrenme katsayısı

C_2 : sürü geneli öğrenme katsayısı

İ : döngü, iterasyon

P_i : i. bireyin konumu

N : birey sayısı

x_i : i. bireyin x konumu

y_i : i. bireyin y konumu

v_i : i. bireyin skaler hızı

w : açısal hız

G_{best} : en iyi ağırlık

P_{best} : en iyi konum

P_{fit} : bireyin en iyi uygunluk değeri

E : hata işlevi

€ : bilgisayardaki en küçük sabit

δ : formasyon mesafe matrisi

γ : iki İHA arasındaki mesafe

KISALTMALAR

AF : Amaç Fonksiyonu
APF : Artificial Potantiel Field (Yapay Potansiyel Alan)
Ar-Ge : Araştırma-Geliştirme
BA : Bat Algorithm (Yarasa Algoritması)
BFO : Bird FlightAlgorithm (Kuş Uçuş Algoritması)
BSA : Bird Swarm Algorithm (Kuş Sürüsü Algoritması)
EA : Evrimsel Algoritma
GA : Genetik Algoritma
GBPSO : Global Best PSO (Küresel En İyi PSO)
GNSS : Global Navigation Sattellite System (Küresel Uygu Seyrüsefer Sistemi)
GPS : Global Positioning System (Küresel Konumlandırma Sistemi)
İHA : İnsansız Hava Araçları
KSO : Kuş Sürüsü Optimizasyonu
PID : Potitential, Integration, Derivate (Oransal, İntegral, Türevsel)
PSO : Parçacık Sürü Optimizasyonu
RAND : Rastgele Oluşturulan Sayı
ROS : Robot Operation Systems (Robot Operator Sistemi)
RTK : Real Time Kinematik (Gerçek Zamanlı Kinematik)
SCT : Singular Case Tolerance (Tekil Durum Toleransı)
SITL : Software In The Loop (Yazılım Döngüsü)
SL : Sürü Lideri
SPP : Self Propelled Particle (Kendinden Tahrikli Parçacık)
TCP : Tranmission Control Protocol (Protokol Takımının Aktarım Katmanı)
UAV : Unmanned Aerial Vehicle (İnsansız Hava Araçları)
UDP : User Datagram Protocol (Kullanıcı Veri bloğu İletişim Kuralları)
UPD : İHA'lar Arası Mesafe
UXC : İHA'ların Ağırlık Merkezinin x Konumu
UYC : İHA'ların Ağırlık Merkezinin y Konumu
Wi-Fi : Wireless Fidelity (Kablosuz Bağlantı)
YAK : Yapay Arı Kolonisi
YSA : Yapay Sinir Ağı

BÖLÜM 1

GİRİŞ

İnsansız hava araçları, içinde herhangi bir insan bulunmayan otonom veya yarı otonom olarak uzaktan kontrol edebilen hava araçlarıdır. İlk İHA kullanımı, 1849 yılında Avusturyalıların pilotsuz 200 balonun kullanımı olarak kabul edilmektedir. Uluslararası literatürde İHA'lara genel olarak "Drone" denilmektedir. Teknolojisinin ilk yıllarında askeri amaçla üretilen drone'ların, sivil alanlarda haritalama, yüksek doğruluk, maliyet, sağladığı kolaylık ve avantajlarından dolayı sonraki dönemlerde sivil alanlarda yayın olarak kullanılmaya başlanmıştır [1].

Doğada sürü halinde hareket eden ve sürü zekasına sahip canlıların incelenmesi ve gerekli modellerin oluşturulması ile askeri ve sivil alanlarda ortaya çıkan karmaşık problemlerin çözümünde İHA'ların kullanılabileceği düşünülmektedir. İHA'ların bu alandaki teknolojilerin temeli olabileceği öngörülmektedir [2]. İHA'ların otonom hareketi, sensor füzyonu, akıllı karar verme, yapay zeka vb. çalışmalara ayrılabilir [3].

Garnier vd. (2007), yaptıkları çalışmada; "Sürü Zekâsının Biyolojik Temelleri" adlı çalışmasında, merkezi kontrol ile bilgi paylaşımı olmadan kolektif hareket edebilen böceklerin davranış modelini referans alarak, karmaşık ve zorlu görevlerin sürü insansız hava araçları ile gerçekleştirilebileceğini belirtmişlerdir [4]. Birçok akademik çalışma doğal yaşam gözleminden esinlenmiştir [2,5,6].

Sürü davranışının mobil robotlar ile ilgili uygulamaları, genellikle doğadaki koordineli hareket eden canlıların işbirlikçi davranışlarına referans almaktadır [7]. Sürü robotiği araştırmalarında odaklanılan kolektif karar verme; bireylerin herhangi bir merkezi lider komutu olmadan, sadece yerel bir etkileşim ile ortaklaşa karar verme yeteneğidir. Birlikte hareket etme davranışına, görev dağıtımı örnek olarak tanımlanır [8,9].

Belirli bir şekil oluşturma, sürü robotlarının tanımlanmış bir görevi gerçekleştirirken, robotların koordineli bir şekilde desen oluşturarak hedefe doğru hareket etmesini sağlayan davranıştır [10]. Bu davranış bakteri kolonileri ve moleküllerin dağılımı gibi desen oluşumlarından esinlenerek ortaya çıkmıştır [11]. Yiyecek arama davranışı, karıncalar ve sosyal böceklerde gözlemlenen yiyecek arama, bulma ve yuvaya getirme davranışından esinlenerek ortaya çıkmıştır [12]. Bu davranış, sürü robotlarına belirli bir alandaki nesnelerin bulunmasını ve geri getirilmesini amaçlayan bir görevdir. Robotlar birbirleri ile modüler bir bağlantı kurarak farklı ayarlamalarla bir görev için organize olabilirler [13].

Bu tez çalışmasının amacı, birden fazla insansız hava aracının (parçacık, ajan, birey), belirlenen geometrik formasyon türlerini oluşturarak kümelenebilmesini sağlayabilmektir. Sürü zekasının gelişimine katkıda bulunması için literatürdeki kontrol modellemelerinden Parçacık Sürü Optimizasyon yöntemi, formasyon kontrol ve çarpışma engelleme algoritmaları uygulanarak bir amaç fonksiyonu oluşturulmuştur. Her bir insansız hava aracının parçacık modeli, simülasyon ortamında oluşturulmuş ve davranışlarının analizi yapılmıştır. Temelde her parçacık belirlenen hedef noktaya ulaşmak için yol planlama işlemine tabi olmaktadır. Bu yol planlama işleminde aynı zamanda çarpışmadan kaçınmak için diğer bireylerle olan güvenli mesafe korunarak yol planlama işlemi gerçekleştirilmiştir.

Bu tez kapsamında, insansız hava araçlarının sürü davranışının simülasyon ortamında modellenmesi ve kümelenme davranışları için ölçeklenebilir algoritmalar geliştirilmiştir. Geliştirilen algoritma ve yöntemlerin GAZEBO-ROS simülasyon sistemlerinde gömülü sistem testleri başarı ile tamamlanmıştır. Sürü davranışının gerçekleştirilmesi amacı ile merkezi ve dağıtık sürü zeka modellemeleri incelenerek, merkezi kontrol algoritmaları için uygulama geliştirilmiştir. Sürü İHA'ların lider-takipçi (merkezi olmayan yarı dağıtık) modellemesinde, her İHA komşu İHA'lar ile etkileşime girerek kümelenme işlemleri sürü lideri tarafından yapılmasını sağlayan algoritma geliştirilmiştir. Sürü üyelerinin birbirleri ile etkileşimde kalmak için gerekli haberleşme alt yapısı, her İHA için oluşturulan Wi-Fi ağındaki ilgili porta telemetri bilgilerini (baş açısı, konum, hız vb.) aktararak komşu İHA'ların bilgilerine ulaşması

sağlanmıştır. Geliştirilen algoritmalar ve uygulanan yöntemler, gerçek dünya ile örtüşen kontrol algoritmalarının simülasyon ve gerçek zamanlı uygulamalarını sunar.

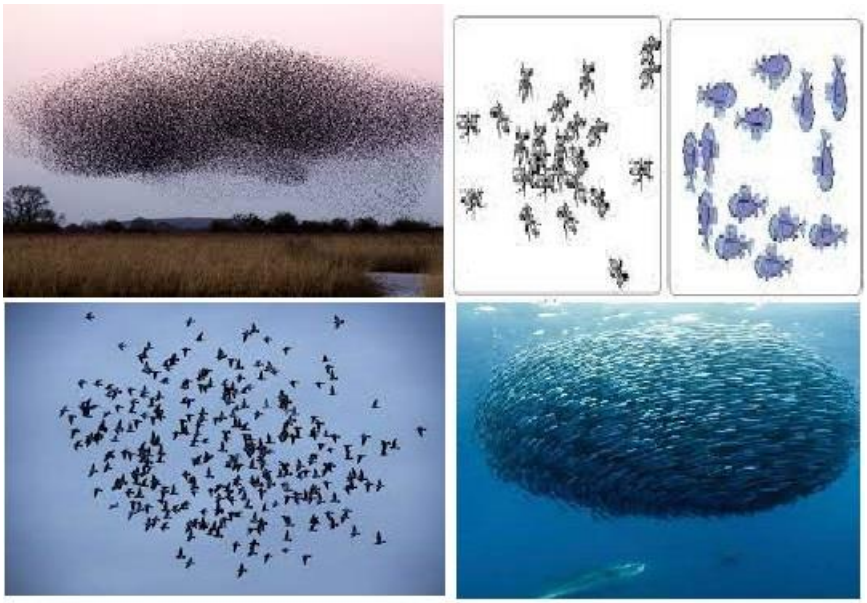
Bölüm 2'de sürü davranışı ile ilgili yapılan çalışmaların literatür taraması yapılmıştır. Bölüm 3'te bu tez çalışmasının konusu olan sürü insansız hava araçlarının modellemesi, sürü davranışının matematiksel modeli, kümelenme algoritmaları ile ilgili simülasyon çalışmaları detaylandırılmıştır. Bölüm 4'te oluşturulan modellerin ve algoritmaların GAZEBO-ROS simülasyon testleri ve gerçek zamanlı kontrol sonuçları oluşturularak Bölüm 5'te sonuçlar değerlendirilmiştir.

..

BÖLÜM 2

LİTERATÜR

Doğada sürü halinde yaşayan canlıların özellikleri bilim dünyasını uzun yıllardır etkilemektedir. Canlı sürülerinin (kuş sürüleri, balık sürüleri veya göç eden hücreler vb.) gelişmiş davranışlar sergileyerek yaptıkları kolektif hareketler biyolojik sistemlerde gözlemlenebilen etkileyici bir olgudur [14]. Bu araştırma alanlarında doğada bulunan canlı türlerinin sürü zekasının modellenmesi bu zekanın karmaşık işleri yapılabileceğinin göstergesidir. Sürü halinde yaşayan canlıların davranışlarının modellenmesi birçok alanda verimli sonuçlar ortaya çıkarabilir. Felaket sonrası arama kurtarma çalışmalarında, insanın çalışmasına uygun olmayan riskli ortamlarda veya belirli görev için bu modeller kullanılabilir [15].



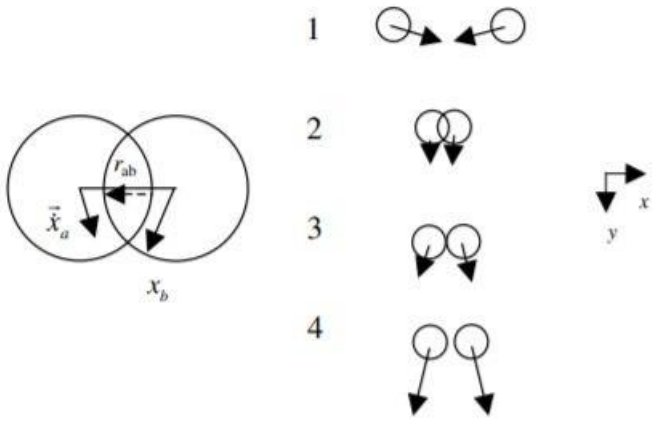
Şekil 2.1. Doğadaki biyolojik sürüler.

Çoklu ajan (multi-agent) olarak da isimlendirilen sürü araştırmaları, arama kurtarma görevleri, askeri operasyonlar, ekonomi, finans, mühendislik alanının birçok

uygulamaları, optimizasyon problemlerinin çözümünde ve diğer otonom görevlere önemli katkı sağlamaktadır. Mühendislik problemlerinin çözümünde istenilen yaklaşım, basit yerel kurallar ile global hedefe ulaşmaktır. Çoklu ajan optimizasyonun da en iyi çözüm tanımlaması, ilk olarak ekonomi alanında çalışmalar yapan F.Y. Edgeworth (1881) tarafından alım ve satım kararı problemlerine çözüm olarak yapılmıştır [16]. Ekonomi alanındaki problemleri analiz eden V. Pareto (1896)'nun çıkardığı En Iyi tanım kavramı büyük kabul görmüştür [17]. Doğadaki sürü davranışlarını inceleyen biyologların ilk çalışmaları Breder [18], Warburton ve Lazarus [19], Okubo ve Grunbaum [20–22] ve Parrish [23] tarafından yapılan çalışmalar olmuştur. Bu çalışmalardan esinlenerek, yakın tarihli birtakım çalışmalar [24–27], sürü formasyonuna önemli katkı sağlamıştır.

Çoklu ajan optimizasyon çalışmalarına da Reynolds tarafından ortaya atılan yöntem önemli katkılar sağlamıştır. Reynolds tarafından ortaya atılan bu model, kuş sürüsünü taklit eden bir algoritmayı kapsamaktadır. Renolds'a göre, çeşitli türlerin kolektif hareketi üç basit ilke sonucu oluşmaktadır [28]. Bunlar; çarpışmaları önlemek için kısa menzilli itme, yakındaki birimlerin hız vektörlerini hizalamak için hizalama kuralı adı verilen yerel bir etkileşim ve sürüyü bir arada tutmak için tercihen küresel konum kısıtlamasıdır. Kuş sürülerinde gözlemlenen bu davranışlar belirtilen kurallar ile gerçekçi sonuçlar oluşturduğunu göstermiştir [29]. Bu kurallar matematiksel formda, aracı tabanlı model olarak yorumlanabilir. Sürüyü bir arada tutmak için, her parçacığın hızlarının ayrı ayrı zamandaki değişimini tanımlayan (ayrık veya sürekli) matematiksel bir form ile dinamik bir model oluşturulmaktadır. Vicsek vd. [30] biyolojik sistemlerin sürü davranışlarında, kendinden hizalanmış hareket modelini inceleyerek kendinden tahrikli parçacık modeli (SPP) geliştirdiler. Önerdikleri model basit hali ile sabit hızda hareket eden parçacıkların, komşularına göre hizalama kuralını kapsamaktadır. SPP modelinin 1 boyutlu analitik hali ile, çöl karıncalarının sürü modelini başarılı şekilde modellediler [31]. Biyolojik yönelimli modeller, çeşitli hayvan veya koloni gruplarında gözlemlenen gerçek davranışı yeniden üretmek için daha spesifik aktif mekanizmalar uygularlar [22,31,32]. Basit kümelenme modelleri, hizalama kuralını matematiksel model olarak tanımlar. Her parçacık, hız vektörünü kendisinin ve komşularının ortalama hız vektörüne göre hizalar [30]. Bu terim; ivmelerin, tercih edilen yönlerin [33] ve daha yüksek hızların kararlılığını genişletmek

için uyarlanabilir. Karar verme şemalarının eklenmesiyle genelleştirmek mümkündür. Diğer (daha karmaşık) modellerde hizalama kuralı, etkileşim kuvvetlerinin veya aşırı sönümlü dinamiklere dayanan hız terimlerinin bir sonucudur [34].



Şekil 2.2. Etkileşim kuralları (sol) ve iki çarpışan ajanın bir taslağı (sağ) [34].

Lien vd. (2004), yaptıkları çalışmada; sürü ajanlarını kontrol eden bir yaklaşımda bulundular. Çalışmalarının performansını çeşitli yöntemlerle karşılaştırarak, performansın yaklaşma ve yönlendirme aşamalarına bağlı olduğunu bildirdiler [35]. Daha sonra bu aşamaları, ajanları belirli bir hedef noktaya hareket ettirmek, daha önce gezilmemiş bölgelere taşımak ve sürüyü toplamak gibi çeşitli çoban davranışlarını geliştirmek için kullandılar [36].

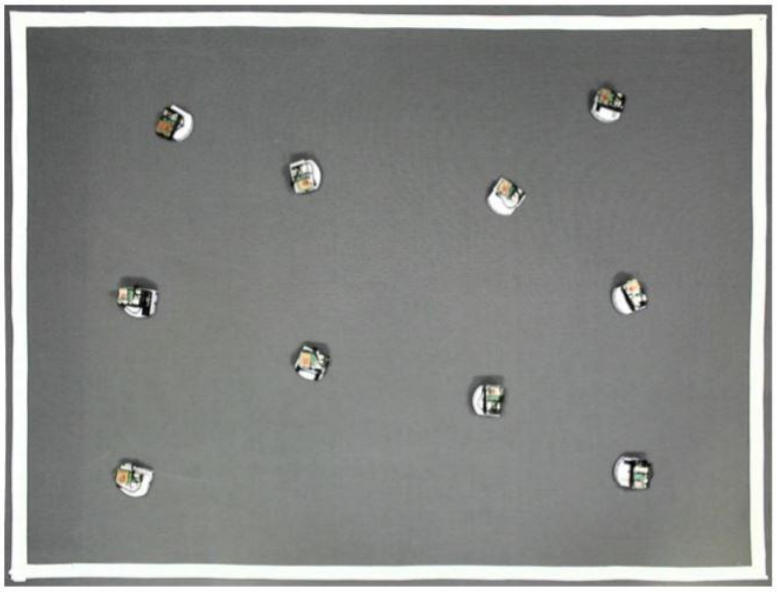
Aldana ve Huepe [37], sürüdeki ajanları temsil eden parçacıkların durağan olduğu durumları temsil eden, vektörel ağ modelini (VNM) önerdiler. Sürülerin hizalanma kuralının geliştirilmesi için parçacıkların uzun menzilli birbirleri ile olan etkileşimlerinin zorunluluğunu araştırdılar. Mermin ve Wagner'in [38] VNM modelinin araştırma kapsamına uygun olarak uzun menzilli etkileşimlerin, veri aktarma için gerekli olan donanımın uzun menzil için yeterli düzeyde olmasını veya yakın komşularının yer değiştirmesiyle komşular arası etkileşimin çok önemli olduğunu gösterdiler. SDP modelinin olabilecek en gerçekçi şekilde araştırma modelini genişlettiler ve bu amaçla asenkron algılama ve çalıştırma ile dönüş açısında kısıtlama yaparak, üç farklı yön hizalama stratejisini karşılaştırdılar [39].

Güzel vd. (2019), yaptıkları çalışmada; sürü ajanlarının bir topluluğa katılma ve her topluluk için merkezi olmayan ve yerel şekilde bir lider atama işleminde karar vermelerini sağlayan yeni bir kümelenme davranışını araştırdıklarını bildirdiler. Atanan lider bir hedefe doğru ilerlemeye devam ederken, diğer ajanların lider ile uyumlu kalarak ve belirlenen kümelenme desenini koruyarak hareket etmesini sağladılar. Her topluluğun, hedefleri verimli ve işbirlikçi olarak keşfedebilmeleri için keşif algoritması tasarladıklarını ve entegre ettiklerini bildirdiler [40].

Schmickl vd. (2008,2010), yaptıkları çalışmada; sosyal böcekler ve arıların davranışlarını gözlemleyerek yeni bir sürü algoritması olarak BEECLUST yöntemini önerdi. Önerdikleri yöntem de bir robotik engel algıladığında durur ve yerel aydınlatma boyutu ne kadar büyükse o kadar bekler ve sonra rastgele dönerek işleme devam eder. Robot-robot iletişimi gerektirmeyen ve robotların konumlarını bilmesini gerektirmediği durumlarda, robotların temas halinde sensör ölçüm değerleri ile orantılı bekleyerek kümelendiklerini bildirmişlerdir [41,42].

Collett ve Collett (2002), yaptıkları çalışmada; karıncalar ve arılar tarafından kullanılan yön bulma tekniklerini araştırarak, yön bulma işleminde işaretlerin güvenilir şekilde kullanılması ve hafızalar arası ilişkinin önemini araştırdılar [43]. BEECLUST yöntemini temel alarak, kendi kendine uyum sağlayabilen yer işareti tabanlı bir toplama yöntemi önermişlerdir. Önerilen yöntemin asıl amacı, düşük yoğunluklarda BEECLUST yönteminin düşük performansını iyileştirdiğini bildirdiler [44].

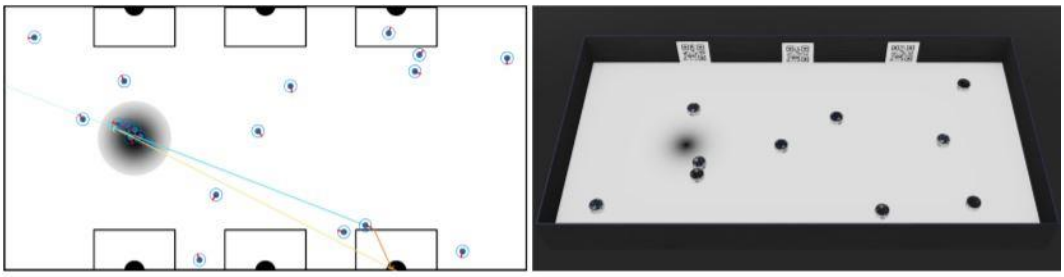
Wahby vd. (2019), yaptıkları çalışmada; sürü robotların BEECLUST yönteminin genişletilmiş bir türü ile kontrol edilebilir sürü kümelenmesini incelemişlerdir. Robotların etrafındaki en parlak noktada toplanmaları istenmektedir. Önerdikleri yöntemde, sürüdeki her robot yerel robot yoğunluğunu ve ışık yoğunluğunu algılayıp bu bilgileri komşu sürü üyeleriyle paylaşabilmesi için BEECLUST yönteminde bazı değişiklikler yapmışlardır. Düzenli deneyler ile robotların dinamik aydınlatma koşullarına uyum sağlayabildiği ve bunun sonucunda düşük yoğunluklu robot topluluğunun kümelenme performansının iyileştirildiğini bildirdiler [45].



Şekil 2.3. Robot sürüsünün kümelenme alanı [45].

BEECLUST yönteminde yapılan değişiklikler ile “bekle” ve “dur” durumu geliştirdiler. Sürü robot kümelenmesi üzerine yapılan önceki çalışmalar, bireysel robotların yeteneklerinin sınırlı olduğunun varsayıldığını ve odometriyi (uzayda konum ve hız tahmini) eklemenin yetenek sınırlarını geliştireceğine katkı sağladığını test ettiler. Odometrinin gerçekleştirilebilir olması ve biyolojik olarak uygun olduğunu ortaya koydular. BEECLUST algoritmasından ilham alan, fakat BEECLUST'tan daha hızlı bir biçimde daha sıkı paketlenmiş robot kümeleri elde etmek için devamlı aktif bir odometri tabanlı hedef arama süreci kullanan ODOCLUST adlı bir algoritma tavsiye ettiler. Simülasyon çalışmalarında yüksek doğrulukta odometrinin gerekli olmadığını bildirdiler [46].

Amjadi vd. (2021), yaptıkları çalışmada; karıncalar ve arılar tarafından kullanılan yol bulma tekniklerinden yola çıkarak, işaret tabanlı bir araya gelme davranışı için bir yöntem önerdiler. Bu yöntemde, karıncaların ve arıların kullandığı, yiyecek arama davranışlarından esinlenerek, robotların işarete dayalı izafi konum bilgisini elde etmek için ortamdaki diğer işaretlerden faydalanılmaktadır. Önerdikleri yöntemin, BEECLUST yöntemine göre, robotların hareketinin düşük robot yoğunluğu da dahil olmak üzere tüm durumlardaki performansından, daha iyi bir sonuç sağladıklarını bildirdiler [47].



Şekil 2.4. Kinematik (sol) ve gerçekçi simulasyon ile (sağ) model testleri [47].

2.1. SÜRÜ ROBOT VE İHA ÇALIŞMALARI

Sürü robot çalışmaları, 1990 yılından günümüze, özellikle askeri uygulamalar için İHA sürüsü için önerilen uygulamalar ve geliştirmeler olmuştur [48,49]. Genel uygulamalar için, sürü İHA araştırmaları yakın zamanda daha fazla ilgi çekmeye başlamıştır. Özellikle, Intel tarafından geliştirilen 300 drone'dan oluşan bir sürü, 2018 Kış Olimpiyatları'nda koordineli bir ışık gösterisi gerçekleştirdiler [50].

Merkezi veya dağıtık sürü İHA'ların kontrolü ve kümelenmesini kapsayan literatürde bildirilen sürü çalışmalarının çoğu fiziksel olarak gerçek dünya şartlarında test edilmemiştir. Teorik yaklaşımlar, yazılım simülasyonları ve laboratuvar koşullarında ara sıra yapılan çalışmalardan ibarettir [11,51].

Bürkle vd. (2011), yaptıkları çalışmada; insansız hava araçlarında sürü çalışmasının uygulanması için bireysel İHA'ların tüm özerkliğine olanak sağlayan, yerleşik işleme, iletişim birimleri ile bütünleşik bir yer kontrol istasyonu geliştirdiler. Etkileşim için olan farklı çeşitlikteki sensörler ile uyumlu şekilde çalışmasını sağladılar [52].

Worth vd. (2021), yaptıkları çalışmada; güncel sürü çalışmaları arasında bulunan ve son 15 yılda yapılan İHA çalışmalarını incelediler. Sürü İHA çalışmasının simülasyon modellemesi için çerçeve modellemesi gerçekleştirerek, sürü davranışının geliştirilebileceği ve test edilebileceği simülasyon ortamının oluşturulması gerekliliği sonucuna ulaştılar [53].

Sürü İHA’lar arasındaki haberleşmeyi tek bir ağda eş zamanlı ayarlamak için uçan geçici ağların (FANET'ler) kullanımı önerilmiştir [54]. Bir kablosuz geçici ağ (WANET) kablosuz erişim noktası olmadan bilgisayarlar arası ağ bağlantısı kurma işlemidir. Geçici bir ağ için yönlendirici veya erişim noktası gerekmez. Bunun yerine, düğümler dinamik olarak atanır ve dinamik yönlendirme algoritmalarına dayalı olarak yeniden atanır. Bu yöntemler ile İHA sürülerinde geçici iletişim ağlarının çeşitli ayarlamalarının uygulaması önerildi [52,55–57]. Bir FANET'te tüm İHA'lar, İHA'lar arasında kurulan bir iletişim ağının parçasıdır. Bu ağ, İHA'lar arasında gerçek zamanlı iletişime izin verir. İHA'lar arasındaki doğrudan iletişim, altyapı tabanlı bir karar motorunda için gerekli olmadığı için dağıtılmış karar vermeyi zorlar. Bu aynı zamanda, tüm sürü istenen görevleri yürütmek için bir altyapıya bağlı olmadığından yerleşik yedeklilik sağlar. Bu, FANET'lerin birincil avantajıdır. FANET'lerin bazı dezavantajları, takas hususlarıyla ilgilidir. Bir FANET kurmak için, her İHA’da ağ donanımı gereklidir. İHA'ların bir FANET'te birbirleriyle güvenilir bir şekilde iletişim kurabilecekleri mesafe, uygulanması için sınırlayıcı bir faktördür [54,55]. Ek olarak, UAV sürü uygulamaları için yönlendirmenin dinamik olarak yeniden yapılandırılması, paket kaybına neden olan zorlu bir iştir. İHA'lar arasında doğru veri telemetrisinin kritik olduğu uygulamalar için güvenilir bir FANET'in kurulması zor bir iştir [54,55,58].

Altyapı tabanlı sürü mimarisi, İHA sürüleri için en yaygın mimaridir [54]. GCS (yer kontrol istasyonu) yazılımı zaten temel altyapı tabanlı sürü yeteneklerini içermektedir [59]. Altyapı tabanlı kümelemenin bir avantajı, İHA’lar arasında ağ kurulmadan optimizasyon ve hesaplamaların bir GCS tarafından gerçek zamanlı olarak, bir İHA’da makul bir şekilde gerçekleştirilebilecek olandan daha yüksek performanslı bir bilgisayar aracılığıyla gerçekleştirilebilmesidir [54,56].

Girolami vd. (2017), yaptıkları çalışmada; sürü insansız hava araçları alanında çalışmalar yaptıklarını ve çeşitli sivil alanlarda İHA sürülerinin kullanımı odaklı yenilikçi uygulamalar ve ağ mimarileri ile ilgilenen SCIADRO2 adlı bir araştırma projesinin hedeflerini açıkladılar. Yenilikçi çalışmalarının ve araştırmalarının sonucunda SCIADRO2 yönteminin devam etmekte olan problemlere ve bazı çözümlere ışık tutacağını bildirdiler [60].

Gaudiano vd. (2003), yaptıkları çalışmada; İHA sürüsünü kontrolünü gerçekleştirmek için ajan tabanlı bir model geliştirdiler. Geliştirdikleri model, bir arama kurtarma görevi yapan sürü İHA'ları kontrol etmek için merkezi olmayan stratejilerden yararlanılması gerektiğini bildirdiler. Model, birkaç farklı görevde İHA sürüsünü kontrol etmek için etkili bir algoritma geliştirme girişiminde aşağıdaki algoritmaları dikkate alır: (1) Taban algoritma, İHA'ların tanımlanmış alanda düz bir rotada uçuş gerçekleştirerek tanımlı alanın sınırına geldiğinde, alandan çıkmamak için dönüş manevrası yaptığı stratejidir (2) her zaman adımında her İHA, yönünü küçük bir rastgele açıyla değiştirebilir. (3) İtme stratejisinde, her İHA belirli bir yarıçap içindeki diğer komşu İHA'ları algılayabilir ve diğer İHA'ları bu itme yarıçapının dışında tutacak şekilde manevra yapar. (4) Feromon stratejisi, bir İHA'nın bir arazi hücresi üzerinde uçtuğunda, hücrenin ziyaret edildiğini gösteren bir işaret bıraktığını varsayar. Diğer İHA'lar daha sonra, hemen etraflarındaki küçük bir yerel alanda, hücrelerin ziyaret edilip edilmediğini belirleyebilirler. İHA'lar daha sonra keşfedilmemiş hücreler üzerinde uçmak için uçuş düzenlerinde küçük ayarlamalar yapar. (5) Sürüdeki her İHA alan sınırlarını bilir ve alan içinde kalmak için sınır yaklaşımında dönme işlemi gerçekleştirir. Çalışmalarında İHA sürülerinin merkezi olmayan kontrol ile başarı şekilde sonuçlandırdıklarını bildirdiler [61].

Corner ve Lamont (2004), yaptıkları çalışmada; bir mikro-İHA sürü sistemi için görev yapma modeli için, doğadan sürü halindeki kuşların, bir balık sürüsünün ve arıların hareketinden esinlendiler. Sürü içinde karar verme davranışının modelini, yüksek performanslı bir paralel ayrık olay simülasyonu ile geliştirdiler. Sistem tasarımından sonra, çeşitli senaryolar için deneyler tasarlayarak test ettiler. Görev yapma modelinin verimlilik ve etkinlik açısından analiz edilmesi gerektiğini geliştirdikleri simülasyon çalışması ile kanıtladılar [62].

Lamont vd. (2007), yaptıkları çalışmada; İHA sürüleri için kapsamlı bir görev planlama sistemi tasarladılar. Sistem, hiyerarşik modele dayalı olarak yol planlama, rota planlama ve sürü davranışı ile birlikte birçok sürü davranış problemlerini bütünleştirmektedir. Geliştirdikleri sistem paralel, çok amaçlı evrimsel algoritma tabanlı arazi takip eden paralel yol planlayıcı ve evrimsel algoritma tabanlı araç yönlendiriciden oluşmaktadır. Hedef, genellikle üç boyutlu bir aracın rota problemi

(VRP) ile ilişkili maliyeti ve riski en aza indirmektir. Bu çabanın doruk noktası, sürü davranışıyla bütünleştirilmiş ve paralel bir İHA simülasyonu ile test edilmiş, genişletilebilir bir gelişimsel yol planlama modelinin geliştirilmesidir. Lamont vd. yol planlaması için verimli çok amaçlı bir evrimsel algoritma geliştirdiklerini belirttiler [63].

Chung vd. (2013), yaptıkları çalışmada; sürü robotik alanlarında yeni operasyon konseptlerine ilham vermek amacıyla sürü insansız hava araçlarında canlı uçma yarışma senaryosu tasarladılar. Bu senaryoya göre sürü İHA'ların bölgelerini savunduğu ve karşı saldırı gerçekleştirmelerini içermektedir. Tasarlanan simülasyonda elde edilen mimari ve modelleme, robotik sürüler için gelişmiş taktikler ve modelleri keşfetmek için yeni fırsatlar sunmaktadır [64].

Campion vd. (2018), yaptıkları çalışmada; İHA sürüleri için hücresel mobil ağ altyapısını kullanarak daha yüksek sürü özerkliği ve güvenilirliğine izin verecek bir sürü mimarisi önerdiler. Ek olarak, önerilen bu mimariyi karşılamak için ilk test ortamı geliştirmesini anlattılar. İHA'dan İHA'ya iletişim ve uyum yeteneği ile daha yüksek düzeyde bağımsız sürülerin özel olarak geliştirilmesi, İHA sürülerinin etkinliğini arttırmanın merkezinde yer aldığını ve hücresel mobil çerçevenin kullanımının, İHA'ların sürü iletişim çalışmaları için kısıtlayıcı etkenleri azaltacağını ve makineden makineye olan iletişim yetenekleri olan 5G ağların kullanımında, bağımsız sürülerin oluşturulması ile verimliliğin büyük oranda artacağını bildirdiler [65].

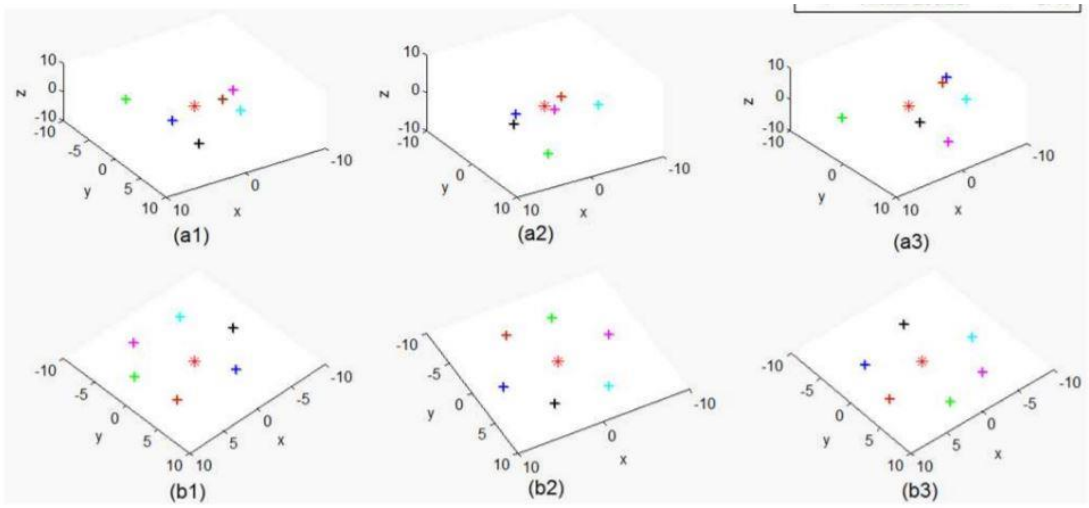
Chen vd. (2019), yaptıkları çalışmada; yüksek hizmet kalitesi (QOS) verileri elde etmek için bulut sunucuları ile İHA sürüsünü bir araya getiren UAV-Edge-Cloud isimli karma bir veri işlem modeli önerdiler. İlk olarak, kaynağın sıkı uygulamaları ve bulut ağlardaki gerçek zamanlı görevleri yerine getirmek ve güçlü kaynaklar sağlamak için bu yeni karma bilgi işlem çerçevesini tasarladılar. Geliştirdikleri bulut sisteminin, İHA sürülerinin ortak bir görev gerçekleştirme ve yönlendirme işleminde oluşan gecikmeyi etkili bir şekilde iyileştirdiğini bildirdiler [66].

Zhu vd. (2015) yaptıkları çalışmada; İHA'ları basit if-then yapısı yerine çok etmenli sistemler olarak modellemenin daha doğru olacağını öne sürdüler. Bu modelin ilk

adımının araç modeli (İHA) davranışlarını, sınırlarını ve bilinmeyenlerini temsil edecek doğru modeli oluşturmak olduğunu belirttiler. Bu çalışma doğrultusunda oluşan kısıtlama ve belirsizlikleri temsil etmek için 6 serbestlik derecesine sahip bir matematiksel model kullandılar. Çalışmalarında Netlogo3D dünyasının modellerini yavaşlattığını, hafızayı tükettiğini belirttiler. Bu nedenle bir sonraki model sürümün uygulama bölümünde, sürü İHA kullanımında İHA modelini yürütmek için esnek, hafif, eş zamanlılık ve kanal iletişim gibi özelliklerinden dolayı Go programlama dilini kullanacaklarını bildirdiler [67].

Demirci ve Uslu (2021), yaptıkları çalışmada; sosyal hayvanların kendi kendine organize davranışlarından ilham alan kolektif hareketin robotikte bir yaklaşımı olduğunu, bu davranışların basit kurallar ve yerel etkileşimler ile, sürü robotların uyumlu hareketi için ölçeklenebilir ve esnek sürü davranışı için yeni bir yöntem olabileceğini öne sürdüler. Çalışmalarında, simülasyon ortamında sınırlı iletişim ve merkezi olmayan yönetim ile sürü davranışı gerçekleştirdiler. Bu sürü davranışlarını şu şekilde sıraladılar; formasyon, formasyonu koruma, ayrılma-birleşme, engelden geçme, lider seçimi ve lider takibi. Önerdikleri yöntemde sürüdeki İHA'ların konumu, varsa kesişmelerinin konumu ve büyüklüğüne göre durumunu belirttiğini bildirdiler [68].

Belkadi vd. (2015), yaptıkları çalışmada; sanal lidere dayalı bir sürü İHA kontrol çalışması yaptılar. Sanal lider kontrol yaklaşımını, parçacık sürüsü optimizasyon yöntemini (PSO) geliştirmek için optimizasyon yöntemleri kullandılar. Yöntemin amacı, önceden tanımlanmış bir amaç fonksiyonunu en aza indirerek belirli bir görevin en iyi performansını garanti etmek için her İHA'nın her anında en uygun pozisyonları bulmak ve İHA'ların sanal bir lider tarafından yönetilen bir görevi takip ederek ve aynı anda grubun içinde olası çarpışmalardan kaçınarak, önceden tanımlanmış bir mimaride 2 boyutlu bir düzlemde İHA'ların organize olabildiklerini belirttiler [69].



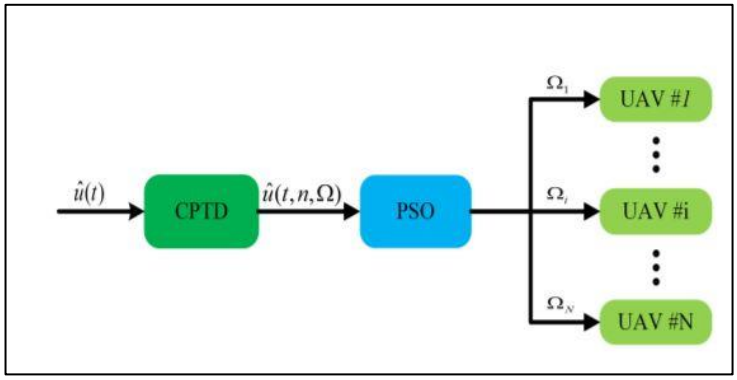
Şekil 2.5. Sabit Sanal Lider ile İHA Filosunun Çember Oluşumu [69].

Chen ve Fei (2017), yaptıkları çalışmada; İHA'lar için üç boyutlu rota planlama performansını artırmak için GBPSO adlı geliştirilmiş bir parçacık sürüsü optimizasyonu (PSO) algoritması önerdiler. Parçacıkların hızlarını ve arama becerisini geliştirmek için, parçacıkların evrim süresince küresel en uygun çözümü optimize etmek için normal bir PSO algoritmasına bir katsayı eklenir. Parçacığa ait bir ajan haline gelerek, başka bir küresel en uygun aday yol, başlangıç noktasından uçuşun son hedefine kadar olan optimal bir yörünge olarak oluşturulur. Ardından, iki aday yolun önceden belirlenmiş yükseklik fonksiyonu değerleri dikkate alınarak global en iyi yol belirlenir. Son olarak, önerilen yöntemin performansını doğrulamak için GBPSO, farklı engellere sahip iki simülasyon senaryosunda mevcut bazı yol planlama yöntemleriyle karşılaştırılır. Çalışmalarının sonucu, GBPSO'nun İHA yol planlaması için daha etkili, sağlam ve uygulanabilir olduğunu göstermiştir [70].

Zhou vd. (2021), yaptıkları çalışmada; değişken bir çevrede yarasa algoritmasının (BA) gelişmiş sürü optimizasyon algoritmasına dayalı insansız hava araçları (İHA) için uçuş yolu planlamasını açıkladılar. Bu çalışmanın temel amacının, karmaşık üç boyutlu ortamda İHA'ların başlangıç ve bitiş rotası arasında çarpışma olmadan güvenli bir uçuş ortaya koyması olduğunu belirttiler. Standart BA ve yapay arı kolonisi algoritmasını (YAK) temel alarak, yarasa algoritmasının (IBA) düzenlenmiş halini, geliştirdiler. Çalışmalarında, IBA'nın yakınsamasını göstermekte ve etkinliğini

doğrulamak için MATLAB ortamında simülasyonlar gerçekleştirdiler. Simülasyonlar, IBA'nın optimum çözümü elde etmesi için gereken sürenin BA'dan yaklaşık %50 daha kısa olduğunu ve optimum çözümün kalitesinin ABC'den yaklaşık %14 daha yüksek olduğunu göstermiştir. Geliştirdikleri algoritmanın diğer sürü yol planlama algoritmaları ile karşılaştırıldığında, IBA, İHA'lar için daha hızlı, daha kısa, daha güvenli, kazasız bir uçuş yolu planladığını bildirdiler. Son olarak, bu makalede IBA'nın işlevleri optimize etmede de iyi bir performansa sahip olduğunu ve geniş uygulama potansiyeline sahip olduğunun kanıtlandığını bildirdiler [71].

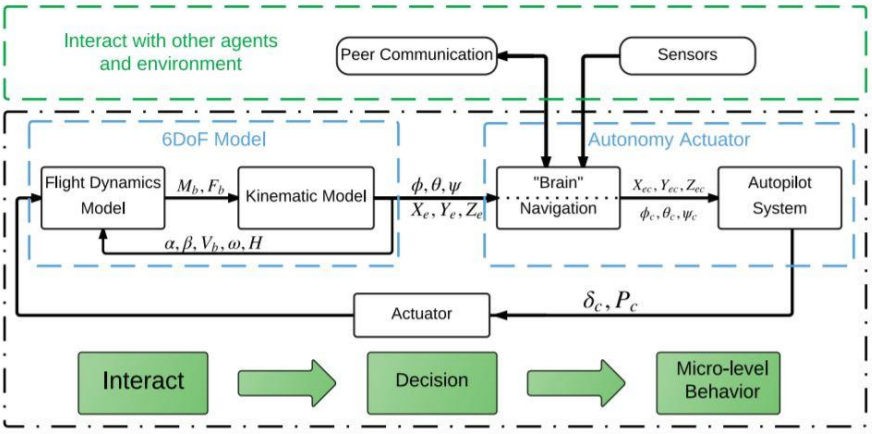
Ghamry vd. (2017), yaptıkları çalışmada; İHA'ların orman yangınları ile mücadele uygulamalarını incelediler. Genel olarak yangınla mücadele görevlerinde esas zorluk, görevi minimum sürede yüksek performansla yerine getirmektir. Çalışmada, ekip açık artırma tabanlı yöntemi kullanarak, görev atama problemine çözdüklerini bildirdi. Yöntemin amacı, sürüdeki her İHA ile yangın noktası arasındaki mesafeyi azaltmak için sürüdeki her İHA'yı en yakın noktaya atamaktır. Her İHA için PSO algoritmasını kullanarak atanmış nokta için rota planlama işlemi yaptıklarını ve önerdikleri algoritmanın İHA'lar arası çarpışmayı önlediğini bildirdiler [72].



Şekil 2.6. PSO algoritmasının kontrol blok diyagramı [72].

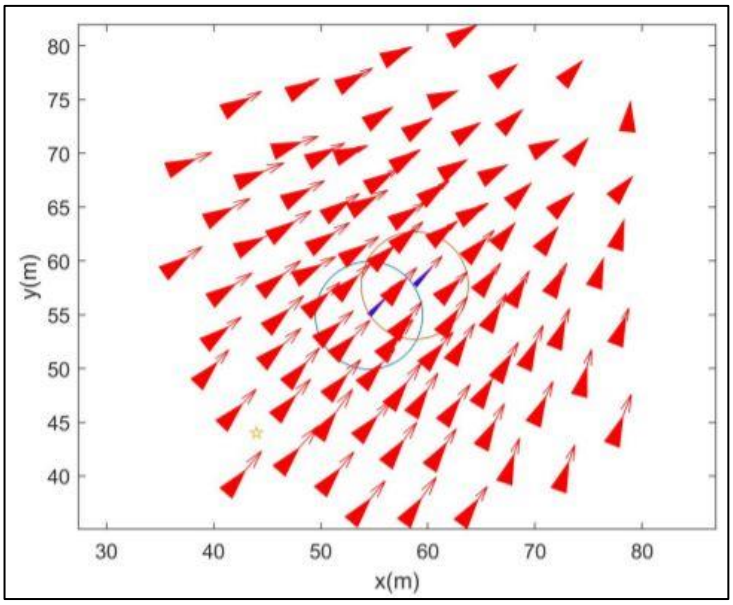
Aniket vd. (2021), yaptıkları çalışmada; sensörlerden alınan veriler ile İHA sürüsü kullanılarak birden fazla hedef arama problemi için merkezi olmayan bir yöntem önerdiler. Önerdikleri yöntem üç ana hedefle ilgilenir: Zamana göre optimize edilmiş çok hedefli arama, optimize edilmiş faydalı yük düşüşleri ve İHA sayısından bağımsız olarak İHA'lar arası çarpışmadan kaçınma işlemlerini kapsamaktadır. Önerdikleri

denetleyici, işbirlikçi ve çok hedefli arama için değiştirilmiş bir Parçacık Sürü Optimizasyonu kullanmaktadır. Bu yöntemi, Çok Hedefli Parçacık Sürü Optimizasyonu (MTPSO) olarak adlandırdılar [73].



Şekil 2.7. Kontrol algoritmasının blok şeması [73].

Liu vd. (2021), yaptıkları çalışmada; karga sürülerinin birlikte uçtuğu doğal olaydan esinlenerek, İHA'nın uyumlu ve yakın uçuşları gerçekleştirmek için sürüde ikili yapıya sahip yeni bir iş birliği modeli ve değişken yapılı kayan kipli denetleyici yöntemi önerdiler. Önerdikleri yöntem ile sürü İHA'ların göreli mesafeye yakınsayabildiğini bildirdiler. İkili mesafe ve akın kontrolünü entegre eden dağıtılmış kontrol protokolü tasarladılar. Tüm sürünün iş birliğini bozmadan İHA'nın sürüde atanmasını sağladılar. Son olarak, iki teoremi Lyapunov kararlılık teoremi ile kanıtladılar. Noktalara atanmış İHA'ların arasındaki mesafe korunarak çarpışmalar engellenmektedir. Simülasyon deneyleri, yöntemin etkinliğini ve doğruluğunu ve eşleştirilmiş İHA'nın sürünün en uzak köşelerinde olduğunda bile mesafe kontrolünün sağlanabileceğini bildirdiler [74].



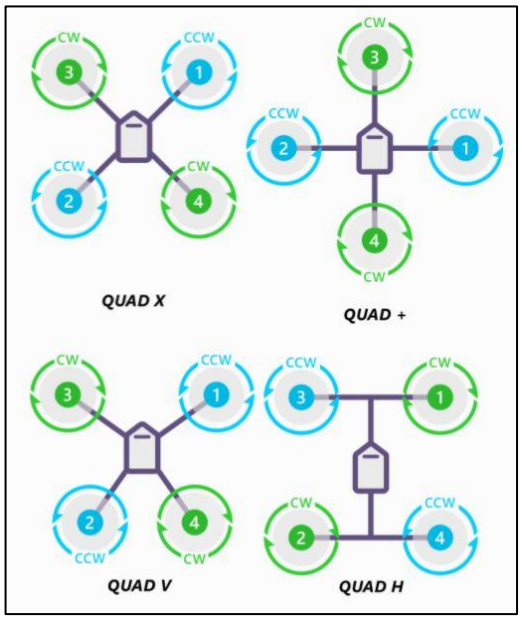
Şekil 2.8. 100 İHA'nın akın işlemi sonrası pozisyon ve yönleri [74].

Petráček vd. (2020), yaptıkları çalışmada; açık iletişim ve merkezi olmayan İHA sürüleri için bir mimari sundular. Önerdikleri mimari, sürü üyelerinin, komşularını tespit etmesi için yön ve menzil tekniğini kullandılar. Teknik, balık, kuş veya sığır sürülerinde görülen biyolojik ilhamlı sürü modeli, engellerin olduğu açık veya kaplı ortamlar için uygun olduğunu bildirdiler. Simülasyon testleri sonrası gerçek dünya testlerini, açık hava koşullarında başarı ile gerçekleştirdiler [75].

Şekil 2.9. Merkezi olmayan sürü İHA (kırmızı), ölçek feransı (mavi) [75].

2.2. DÖRT ROTORLU İNSANSIZ HAVA ARACI

Dört rotorlu İHA (quadrotor) hareketi pervanelerin birbirine göre hızlarının değiştirilmesiyle sağlanır. Pervanelerin ikisi saat yönü, diğer iki pervane saat yönü tersine hareket etmektedir. Bu zıt yönlü hareket sayesinde helikopterde zorunlu olan kuyruk dönüş hareketi ihtiyacı ortadan kalkmıştır [76].

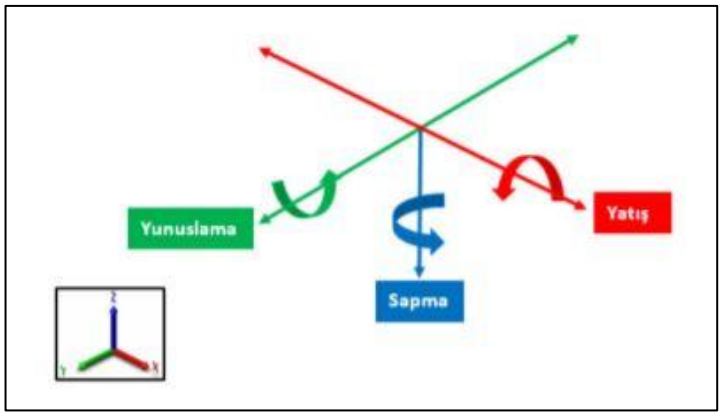


Şekil 2.10. Dört rotorlu İHA çerçeve türleri.

Silva vd. (2019), yaptıkları çalışmada; İHA'nın SITL (Software in the Loop) verileri ile irtifa kontrolünü gerçekleştirdiler. Bu alt yapı ile PID kontrolör tasarlanmıştır. İHA'nın bilgisayar ile haberleşmesi MAVLİNK protokolü ile sağlanmıştır. Yer istasyonu geliştirilerek İHA'nın dronekit kütüphanesi aracılığı ile uzaktan iletişiminin nasıl olacağını gösterdiler [77].

2.2.1. Eksen Modeli

İHA'nın kinematik modelinin oluşturulması ve simülasyon ortamın doğruluğunun sağlanması için Newton-Euler yöntemi kullandılar. İHA'nın hareketi yunuslama açısı (θ), yuvarlanma açısı (ϕ), ve sapma açısı (ψ) bilgilerini içeren açısal konumu PID kontrolör ile sağlamışlardır [78].



Şekil 2.11. Quadrotorun x, y ve z ekseninde hareketleri [78].

$$R = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}, \quad (2.1)$$

$$\dot{\theta} = [\phi \quad \theta \quad \psi]^T = T\omega, \quad (2.2)$$

T olarak ifade edilen, Euler Matrisi olarak bilinir.

$$T = \begin{bmatrix} 1 & sin\phi tan\theta & cos\phi tan\theta \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi sec\theta & cos\phi sec\theta \end{bmatrix}, \quad (2.3)$$

Quadrotorun 6 serbestlik dereceli hareket denklemi çıkarılırken;[79].

$$\varsigma = J_r \dot{\theta} \Omega_r, \quad (2.4)$$

$$R_t = (-1)^{i+1} \sum_{i=1}^{4} R_{mxi}, \quad (2.5)$$

$$I_{xx} = \dot{\theta}\dot{\psi}(I_{yy} - I_{zz)} + \varsigma + I(-F_2 + F_4) - h(\sum_{i=1}^{4} H_{yi}) + R_t, \quad (2.6)$$

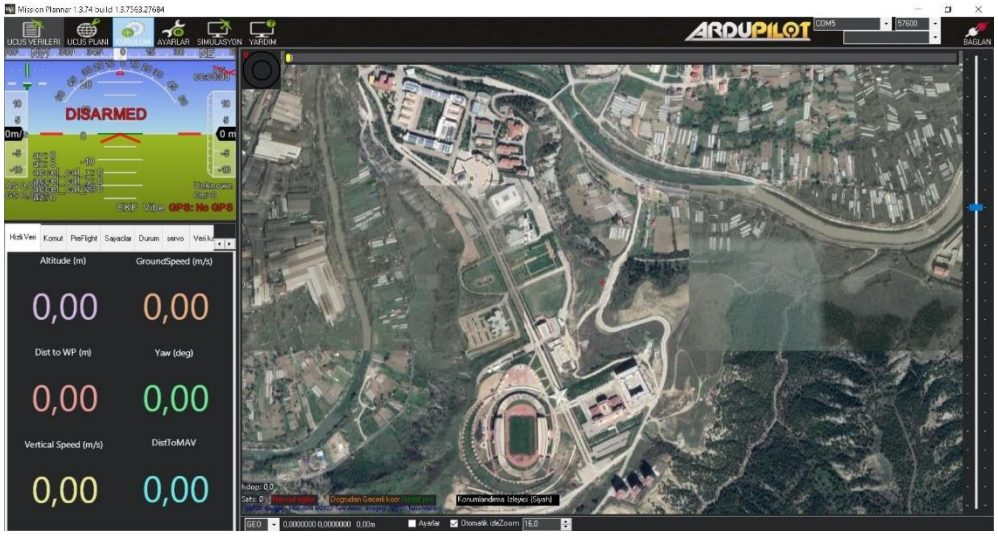
$$I_{yy} = \dot{\theta}\dot{\psi}(I_{zz} - I_{xx)} + \varsigma + I(-F_1 - F_3) - h(\sum_{i=1}^{4} H_{xi}) + R_t, \quad (2.7)$$

$$I_{zz} = \dot{\theta}\dot{\phi}(I_{xx} - I_{yy)} + \varsigma - I\left(H_{y1} - H_{y3}\right) + I(H_{x2} - H_{x4}) - 1^{i} \sum_{i=1}^{4} Q_i, \quad (2.8)$$

2.2.2. Konum Belirleme (GPS)

Yüksek doğrulukla konum noktası belirleme işlemi küresel navigasyon uydu sistemi (GNSS) yaygın olarak kullanılmaktadır. GNSS, uydu verileri ile elde edilen konumsal noktanın yüksek doğruluk ve gerçek zamanlı konum kontrolü işlemlerinde kullanılmaktadır [80].

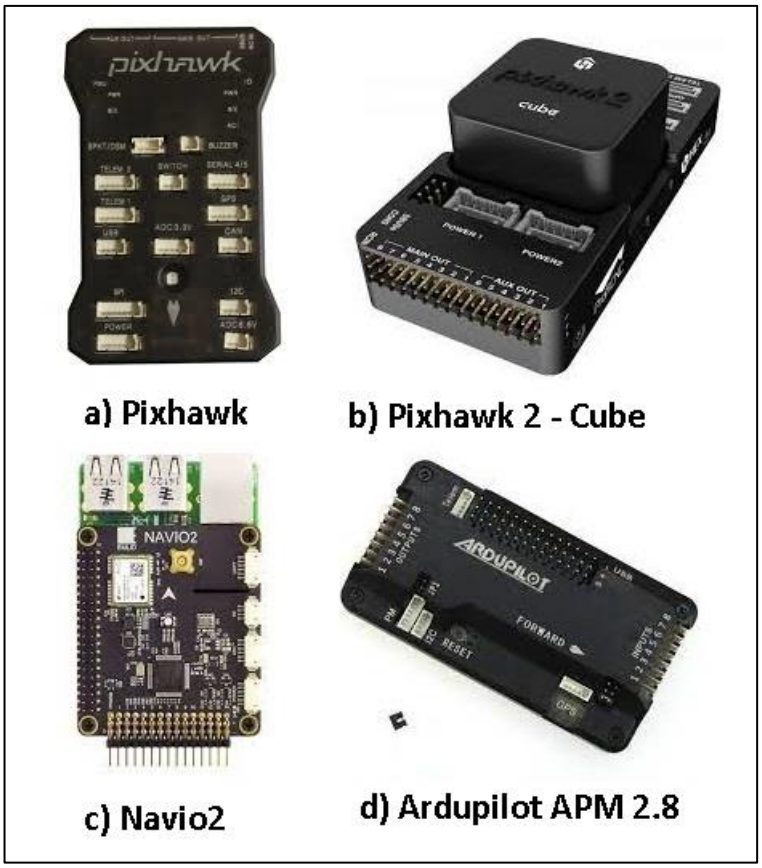
GPS tarafından anlık olarak alınan konum bilgisi haritada enlem, boylam ve yükseklik olarak aktarıldığında küresel konum formatında nokta tespiti yapılabilmektedir. Bunun için Mission Planner (Ardupilot) tarafından, okunan GPS verisi değerlendirilir ve harita ortamında işaretlenir.



Şekil 2.12. Mission planner (ardupilot) İHA kontrol-navigasyon yazılımı.

2.2.3. Kontrol Kartları

Uçuş kontrol kartları, kontrol yasalarına göre (PID, kalman filtresi vb.) uygulayarak motorları kontrol eder. Dahili ve harici sensörleri, navigasyon işlemlerini veya komşu İHA'lar ile iletişim kurar. Genellikle çözünürlükleri 32 bit olan işlemci kullanır. Performansı kullanılan gömülü işlemci birimine bağlıdır [81].



Şekil 2.13. Otopilot özelliği olan örnek uçuş kontrol kartları.

2.3. FORMASYON

Doğal olayların matematiksel modelini incelemek bilim dünyasının ilgisini çekmiştir. Kümelenme davranışı veya yaşam formlarının topluluk halinde bir araya gelmesi doğada çok sayıda mevcuttur. Bu kümelenme davranışı hayvanlarda, kaz sürülerinde, kuş sürülerinde balık sürülerinde, memeli sürülerinde vb. görülebilir. Kaz sürüleri genellikle ters "V" oluşumunda uçar. Ters "V" şeklinde formasyonunda uçuş yapmaları birtakım avantajlar sağlamaktadır. Bu avantajlardan bazıları, her bir kazın kendi başına uçmasına göre %24 daha hızlı uçma gücüne ve %71 daha fazla uçuş menzili sağlamaktadır [82]. Formasyonun kontrolü ve İHA'ların oluşturulan formasyon noktaları ile eşleştirilmesi gerekmektedir. Atama işlemi, formasyona minimum sürede ve çarpışma olmadan ulaşmayı amaçlamaktadır. Hungarian (Macar) algoritması [83] kesikli problemlerin (kombinatoryal) çözümünde kullanılmaktadır.

Formasyon kontrol işlemi etkileşim topolojisine göre, konum, yer değiştirme ve mesafe kontrolü olarak sınıflandırılabilir [84].

Turpin vd. (2004), yaptıkları çalışmada; sürü robotlar için eş zamanlı atama ve yörünge oluşturma (CAPT) problemine çözüm olarak merkezi olmayan bir yöntem (D-CAPT) ile çözüm önerdiler. C-CAPT algoritması, CAPT sorununa engelsiz ortamda çarpışma olmadan en uygun çözümleri sunan algoritmadır. Önerdikleri yöntemde, yörünge süresinde hızın karesine dayalı maliyet fonksiyonunu azaltan, atama işlemi ve rota planlama sorununa merkezi çözüm olan C-CAPT algoritmasını geliştirerek, sekiz adet dört rotorlu mikro hava aracına başarıyla uyguladıklarını, yörüngelerin küresel olarak optimum ve güvenli olduğunu bildirdiler [85].

Gazi vd. (2007), yaptıkları çalışmada; sürülerin kümelenme, formasyon kontrol problemleri üzerinde kayan kip kontrol yöntemi ve yapay potansiyel fonksiyonlar ile stratejiler geliştirdiler. Çalışmalarında yiyecek arama, formasyon kontrolü ve hareketli hedefleri, aynı yaklaşımı uygulayarak detaylandırma çalışmaları yaptıklarını ve oluşturdukları stratejilerin simülasyon çalışmalarını başarıyla gerçekleştirdiler[86].
Yao vd. (2006), yaptıkları çalışmada; sürü ajanlarının formasyon kontrolü ve hareketli hedefi takip etmesi için merkezi olmayan kararlı bir kontrol yöntemi sundular. Hedef izleme ve formasyon kontrol işlemleri için yapay potansiyelleri kullandılar. Önerdikleri yöntemin lider-takipçi modelinden daha kararlı olduğunu yaptıkları simülasyon testleri ile kanıtladılar [87].

Tanner vd. (2004), yaptıkları çalışmada; lider takibine dayanan mobil ajan oluşumunun kararlılık özelliklerini incelediler. Formasyon da gözlemlenen, ajanlar arası mesafe hatalarının, lider davranışını nasıl etkilediğini doğrusal olmayan kazanç tahminleri oluşturdular. Lider Formasyon Kararlılığı (LFS) kazanımlarının iyileştirilmesi için yöntem önerdiler [88].

Miswanto vd. (2015), yaptıkları çalışmada; sürülerin formasyon kontrolünü, lider rolüne sahip ajanın izlediği yolu takip ederek sağladılar. Liderin kontrolünü Pontryagin Maximum Principle yöntemi ile formasyon kontrolünü ise geometri

yaklaşımını kullanarak, her ajanın oryantasyonu ve konumunun lidere bağlı olarak kontrol edilebileceğini gösterdiler [89].

Desai vd. (2001), yaptıkları çalışmada; holonomik olmayan mobil robotlar ile engeller bulunan arazide, grafik formasyon kontrolü kullanarak istenilen formasyonun korunması ve istenildiğinde formasyon geçişini sağlamak için kontrol stratejileri oluşturdular [90].

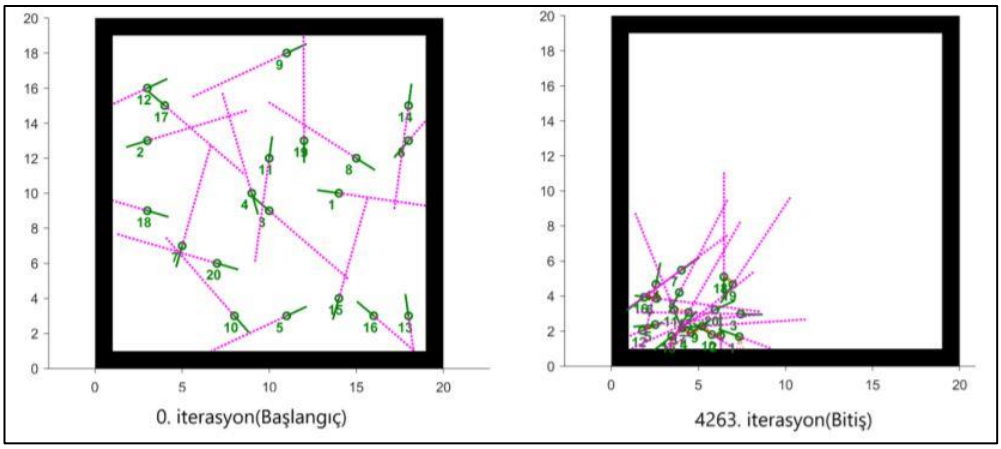
Xie vd. (2000), yaptıkları çalışmada; kontrol yasasını tasarlarken kayan mod teorisinden gelen fikirlerle birlikte zincirleme form sisteminin doğal cebirsel yapısını incelediler. Holonomik olmayan sistemden zincirleme olarak adlandırılan sistem için stabilizasyon ve izleme problemine kayan mod yaklaşımının göz önünde bulundurulması gerektiğini bildirdiler [91].

Mancini vd. (2007), yaptıkları çalışmada eşit sayıdaki bir sürü robot topluluğu durumunda bir lideri takip etme probleminin çözümü için yaklaşım önerdiler. Görevi takip eden bir lider gerçekleştiren holonomik olmayan robotların kontrolü için ayrık zamanlı kayan mod yaklaşımı kullandılar [92].

Pranoto vd. (2012), yaptıkları çalışmada; formasyon kontrolü için lider takip algoritmasını simülasyon ortamında gerçekleştirdiler. Belirli bir geometri oluşumu ile hareket eden Dubin'in araba modeli ile sürü için en iyi takip kontrol problemini ele aldılar. Çalışmalarında üç ajan ve bir sürü lideri modeli oluşturarak, ajanların lider yolunu takip etmek için hareket ettiklerini ve bu takip sırasında oluşan hatanın çok düşük olduğunu gösterdiler. Her sürü üyesinin (ajan) konum ve oryantasyon değerlerinin sürü liderine bağlı olarak kontrol edildiğini gösterdiler. Oluşturdukları simülasyon ile çalışmalarını başarı ile tamamladıklarını belirttiler [82].

Mısır vd. (2020), yaptıkları çalışmada; sürü robotları için bir araya gelme (kümelenme) yöntemini önerdiler. Önerdikleri kümelenme metodu homojen robotların, merkezi kontrol olmadan, kısıtlı mesafe sensörü ve açı verileri ile kümelenmesidir. Çarpışmadan kaçınma kontrolcüsünün, kümelenme kontrolcüsünden ayrı olduğunu bildirdiler. Her robotun kümelenme işleminde bireysel karar verdiği ve sürüde bulunan

robotların her birinin algıladığı robotlardan en yakın olanını takip ettiğini ve robotların algıladığı diğer robotların hareketine göre yönlendiğini belirttiler. Kümelenme davranışının başarıyla uygulandığını, robot sayısı arttıkça performansın düştüğünü bildirdiler [93].

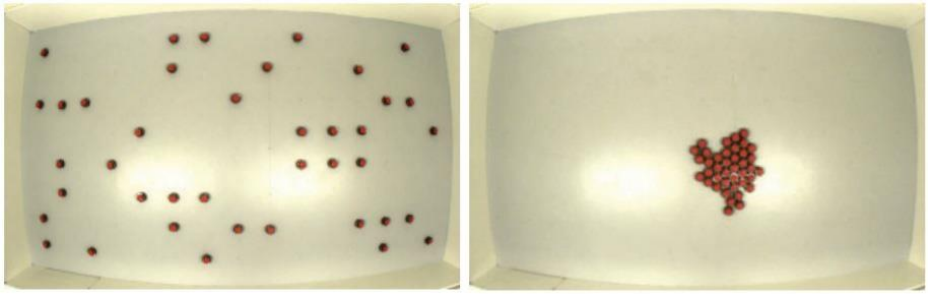


Şekil 2.14. 20 robot sürüsünün toplanma davranışı [93].

Sial vd. (2021), yaptıkları çalışmada; İHA'ların dağıtık formasyonlarının kontrol yöntemi için yeni bir arama ve görev yürütme modeli geliştirdiler. Çalışmalarının ilk aşamasında çoklu yapay potansiyel alan ve ajan çizge teorisi ile dağıtık İHA oluşumu için akın etme, çarpışmadan kaçınma ve izleme problemlerine çözüm bulmayı hedeflediler. Önerdikleri algoritmanın ilk aşaması, hedefi arama ve görev yürütme işlemlerini içermektedir. Son aşamada ise görev işlemi sırasında akın etme işlemi uygulanmaktadır. Simülasyon ortamında sürü akın etme işlemleri ve görev yürütme algoritmalarının başarı ile test edildiğini bildirdiler [94].

Mechali vd. (2021), yaptıkları çalışmada; İHA'ların lider-takipçi etkileşimi oluşturan ve sürekli olmayan bozucu etkiler ile doğrusal olmayan davranışları için yeni kontrol yöntemi önerdiler. Dağıtılmış bir dizilim için kontrol yasası tasarladılar. Formasyonun oluşum noktaları belirlenerek, referans oluşum yörüngesi oluşturulur ve sentezlenmiş sabit zamanlı konum kontrolü yöntemi kullanılarak oluşturulan yörünge takip edilmiştir. Kontrol performansını analiz etmek için ROS/GAZEBO ortamında simülasyon çalışması gerçekleştirdiler. Çalışmalarının, diğer kontrol yöntemlerine göre yüksek performans gösterdiğini bildirdiler [95].

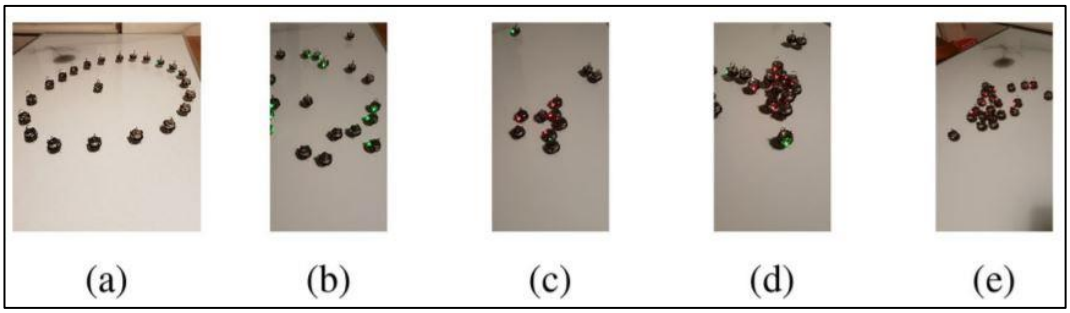
Gauci vd. (2014), yaptıkları çalışmada; yapılan bir araya gelme davranışı çalışmasında, kontrolcü olarak "if-then-else" yapısı kullanarak hesaplama gerektirmeyen kendi kendine organize olabilen bir toplanma sorununa çözüm olarak kontrolcü önerdiler. Çalışmada kullandıkları sürü robotlarının komşularını algılamaları için 2 bitlik algılayıcı kullandılar. Kullandıkları bu algılayıcıların algılama mesafe sınırını ilk olarak sınırsız olarak belirlediler. Daha sonra uyguladıkları deneylerle algılama aralığını değiştirerek, bu etkinin bir araya gelme davranışını ve kontrolcü performansını incelediler [96].



Şekil 2.15. Örgütlenebilir sürü kümlenme davranışı deneyi [96].

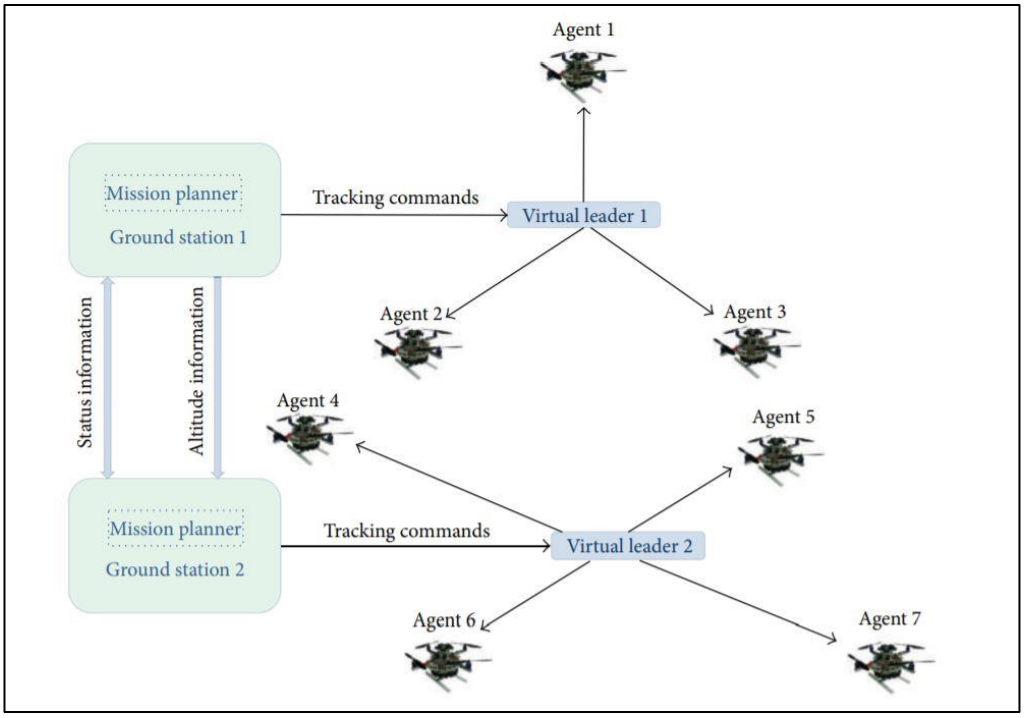
Mısır vd. (2020), yaptıkları çalışmada; bulanık mantık tabanlı öz örgütlenebilir bir araya gelme metodu önerdiler. Çalışmalarında geleneksel kümelenme yöntemlerinden farklı olarak, denetleyici olarak bulanık mantık kullanarak kısıtlı sensör verilerini değerlendirdiler. Benzetim ortamında farklı algılama alanlarına sahip, farklı sayıdaki sürü robotlarına, farklı büyüklüklerdeki alanlarda düzenli deneyler yaptıklarını ve sürü robotlarının, yapılan düzenli deneyler aşamasında, algılama mesafesi ve robot sayısındaki değişikliklere rağmen kümelenme davranışı sergilediklerini bildirdiler [97].

Parhizkar vd. (2020), yaptıkları çalışmada; bir sosyal amip olan Discotyostelium Discodeum'un kümelenme (bir araya gelme) davranışlarından esinlenerek, bu biyolojik örneği sürü robotları üzerinde uygulamışlardır. Bir araya gelme davranışında biyolojik örneğin, sinyal yayılım modelini benzetim ortamında incelemişlerdir. Çalışmada biyolojik bir araya gelme davranışını, fiziksel sürü robotları kilobot'lar üzerinde de göstermişlerdir [98].



Şekil 2.16. Alanın merkezindeki bir kilobotların aşamalı olarak toplanması [98].

Ali ve Montenegro (2016), yaptıkları çalışmada; belirlenmiş farklı formasyon deseni için 2 İHA oluşumunun birlikte çalışması için yeni bir çerçeve önerdiler. Formasyon türleri yer istasyonu tarafından görev ataması ile gerçekleştirilerek sanal lider ile komuta edilmiş, Doğrusal Kuadratik Regülatör Orantılı İntegral (LQR PI) kontrol algoritması ile konum kontrolü sağlanmıştır. Bu yöntemle, formasyon oluşumundaki İHA'lar, istenilen noktalara yol planlama işlemi ile belirlenen sürede noktalara ulaşır. İletişim kaybı olması durumunda komşu İHA'lar ile komutlar bu İHA'lara aktarılır. Yaptıkları çalışmayı MATLAB/Simulink ile doğruladıklarını bildirdiler [99].



Şekil 2.17. İki oluşum için mimari [99].

Toksöz vd. (2019), yaptıkları çalışmada; merkezi olmayan sürü formasyon kontrolü için hibrit bir sürü kontrol mekanizması önerdiler. Önerdikleri çalışma formasyon kontrolü ile çarpışmadan kaçınmayı da kapsamaktadır. Her İHA'nın formasyon oluşumunun bozulmaması ve aralarındaki mesafenin korunması için düşük seviyeli kontrol algoritması tasarladılar. Mesafenin korunması için her İHA'nın istenilen konumda olması gerektiğinden konum kontrolü sağlanarak önerdikleri algoritmanın simülasyon ve gerçek dünya deneyleri ile doğruladıklarını bildirdiler [100].

2.4. ÇARPIŞMADAN KAÇINMA

Waydo ve Murray (2003), yaptıkları çalışmada; hidrodinamik analizden esinlenerek, iki boyutlu potansiyel alan tabanlı rota planlama fonksiyonları oluşturmak için yerel-minimum olmadan yeni bir model olarak Laplace denklemini sağlayan akış fonksiyonlarını bildirdiler. Bu işlevlerin, diğer yöntemlere kıyasla daha yumuşak yollar (hava araçları benzeri araçlar) oluşturduğunu belirttiler. Engellerden kaçarken, rastgele araç davranışları oluşturmak için analitik akış fonksiyonlarının üretilmesi adına bir yöntem geliştirdiler. Yöntemin bir engelin olduğu ve düzgün hareket ettiği durum için kesin bir çözüm sunduklarını bildirdiler [101].

Ye vd. (2005), yaptıkları çalışmada; sürülerin kolektif hareketleri sırasında engellerden kaçmak için bir çerçeve önerdiler. Önerdikleri çerçeve, karmaşık sürü davranışlarını uygulayabilmek için potansiyel akışlardan, yapay potansiyellerden ve dinamik bağlantılardan gelen kavram ve teknikleri akıllıca birleştirir. Başlangıç olarak, akışkanlar mekaniğindeki potansiyel akışlardan mevcut kavramlar, tek-etmenli seyrüsefer problemini çözmek için kullanılır. Ek olarak, durgunluk noktası problemine analitik bir çözüm sunulur. Potansiyel akışa dayalı çerçeve daha sonra, birden fazla engel arasında seyreden sürülerin koordineli kontrolünü kolaylaştırmak için önemli ölçüde değiştirilir. Arttırılmış engellerden kaçınmanın yanı sıra kümelenme için yapay potansiyeller kullanılır. Böylece, engellerden kaçınma (Görüş Hattı Bağlantısı) performansını geliştirmek ve çeşitli sürü davranışlarını düzenlemek (Rastgele Bağlantı) için yeni bir dinamik bağlantı kavramı kullanılmaktadır. Önerilen çerçevenin

uygulanabilirliğini göstermek için bir dizi geliştirilmiş algoritma ile simülasyon sonuçları çalışmalarına dahil edilmiştir [102].

Merheb vd. (2016), yaptıkları çalışmada; viskoz akışlarının sıkıştırılamaz potansiyel fonksiyonlarına ve özelliklerine bağlı basit uygulamalı bir sürü navigasyon (yol planlama) algoritması önerdiler. Panel yöntemleri, kompleks desenli engellerin etrafındaki akış denklemlerini çözmek ve hedef konuma çarpışma olmayan yolları takip eden akış çizgilerini oluşturmak için kullanılır. Oluşturulan akış çizgileri takip edilerek güvenli sürü navigasyonu sağlanır. Navigasyon sırasında grup uyumunu veya geometrik bir oluşumu sağlamak ve sürdürmek için potansiyel işlevler kullandıklarını bildirdiler. Geliştirdikleri algoritmanın, uygulanmasının kolay olduğunu ve robotik sürülerin navigasyonu için etkili bir araç olarak hizmet edebileceğini bildirdiler [103].

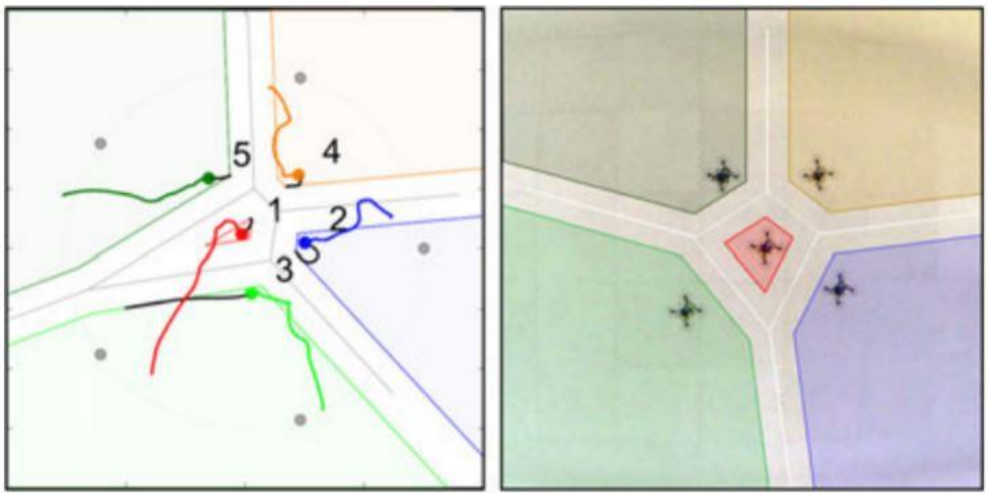
Huang vd. (2019), yaptıkları çalışmada, farklı uçuş koşullarında, birden fazla İHA sürüsünün çarpışmalarını engellemek için yöntem geliştirdiler. Önerdikleri yöntem, sürü uçuş işlemleri sırasında her İHA sürüsünün uçuşunu bireysel olarak sağlayabildiğini ve her sürü içerisinde İHA'lar, kendi kendini organize ederek çarpışmadan kaçınma sürecinde sürü uçuşuna devam edebileceğini bildirdiler. Çarpışmadan kaçınmanın İHA sürüleri tarafından başarıyla sağlandığı ve bu yaklaşımın etkinliğinin doğrulandığını bildirdiler. Sürü uçuşu sırasında, çarpışmadan kaçınma işleminin çok önemli olduğunu ve önerdikleri algoritmanın etkinliğini göstermek için sayısal ve deneysel testler sonucunda çarpışmadan kaçınmanın İHA sürüleri tarafından başarıyla sağlandığı bu yaklaşımın etkinliğini doğruladıklarını bildirdiler [104].

Xu vd. (2021), yaptıkları çalışmada; İHA'ların ortak bir hedefi takip ederken çarpışmadan kaçınmasını incelediler. Çalıştıkları problem, ortalama girişimi en aza indirmek ve İHA'lar arasında çarpışmadan kaçınmayı sağlamak için ortaklaşa formüle ettiler. Tekil durum toleransı (SCT)-yapay potansiyel alanı (APF), İHA'lar arasındaki itme kuvveti kazanç katsayısının karşılık gelen müdahaleler tarafından dinamik olarak kontrol edildiği, geleneksel APF'lerin çarpışmadan kaçınmadaki başarısızlığının üstesinden gelmek için önerdiklerini bildirdiler. İHA'ların yörüngelerini ve gücünü ortaklaşa ayarlamak için çift alanlı bir iş birliğine dayalı kontrol yaklaşımı önerdiler.

Önedikleri çift alanlı iş birliği yaklaşımı, İHA sürüsü hedefe yakın olduğunda yaklaşık %117 verim kazancı ve %88 parazit azalması sağlayabildiğini bildirdiler [105].

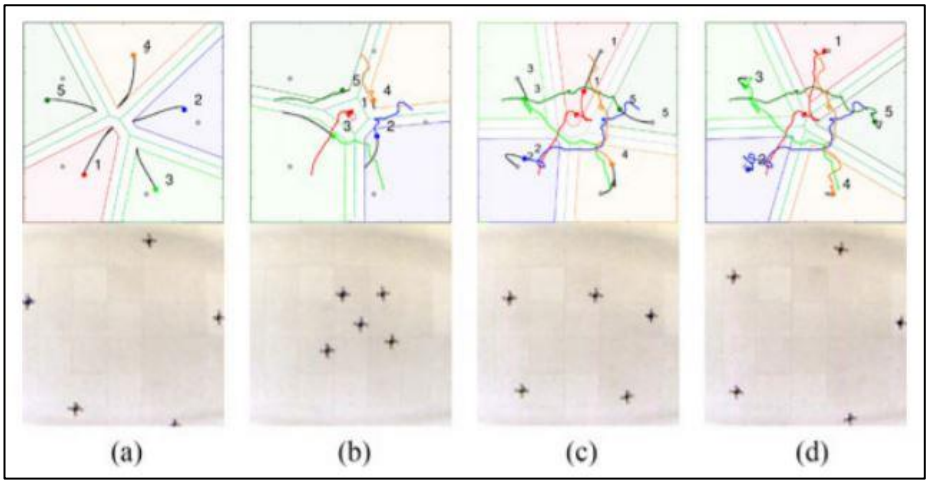
Kablosuz iletişim, sensörler ve pil teknolojilerinin hızla gelişmesiyle birlikte, Sürü İnsansız Hava Araçları (İHA'lar), trafik gözetimi ve askeri uygulamalar için yaygın olarak kullanılmaktadır. Bununla birlikte, Sürü İHA'larının, aynı anda birden fazla Muntasha vd. (2021), yaptıkları çalışmada; Yapay Arı Kolonisi (ABC) algoritması kullanılarak sürü İHA'ların rota planlama ve çarpışma engelleme işlemlerini gerçekleştirdiler. ABC algoritması, arıların organize olma yeteneği, yerel ve küresel optimum oluşturmak için kendilerini koordine etmelerine olanak tanır. Önerdikleri sistem, İHA'nın hızını kontrol etmek için ABC algoritmasını kullandılar. Yaptıkları simülasyon çalışmasında, 12 ve 20 drone ile İHA'lar arası potansiyel çarpışma ile hedefe başarı ile ulaştıklarını bildirdiler [106].

Zhou vd. (2017), yaptıkları çalışmada; rastgele hareket eden birden fazla dinamik araç için dağıtılmış bir çarpışmadan kaçınma algoritması sundular. Algoritmalarında, her robot gerçek zamanlı Tamponlanmış Voronoi Hücresini (BVC) hesaplayarak, hücre içindeki rotasını uzaklaşacak şekilde oluşturur. Önerdikleri algoritmanın, optimal karşılıklı çarpışmadan kaçınma (ORCA) algoritmasıyla aynı olan O(k) hesaplama karmaşıklığına sahip olduğunu, sıralı dışbükey programlama (SCP) ve model öngörücü kontrol (MPC) işlemlerinden çok daha hızlı olduğunu belirttiler. Algoritmalarının sadece algılanan göreli konumu gerektirmesi ve gürültülü göreceli konum sensörleri ile iyi çalıştığını bildirdiler [107].



Şekil 2.18. Çarpışmadan kaçınmak için kullanılan Voronoi hücresi [107].

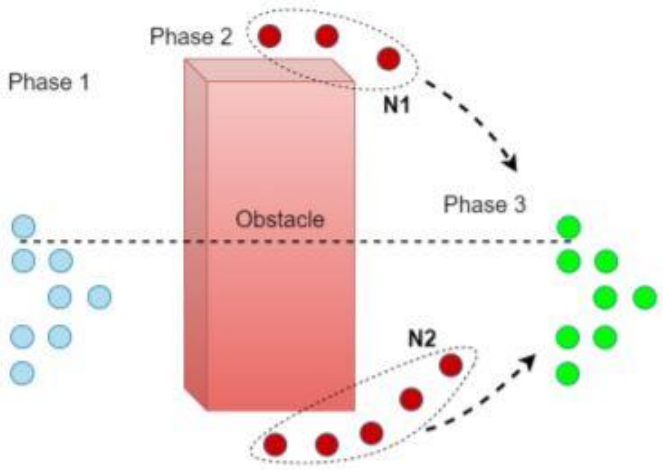
Çalışmalarının [107] simülasyonunu, bir robotik simülatörü olan Gazebo'da sekiz adet İHA kullanarak algoritmalarını 3B uzayda başarıyla doğruladıklarını bildirdiler. Simüle edilmiş quadrotor dinamiklerinin aslına uygunluğunu sağlamak için, robotu oldukça kapsamlı bir şekilde modelleyen Robot İşletim Sisteminde (ROS) hector quadrotor paketini [108] kullandıklarını bildirdiler.



Şekil 2.19. Algoritmanın a(0), b(10), c(20), d(30) deney sonuçları [107].

Yasin vd. (2020), yaptıkları çalışmada; engel ile karşılaşan sürü ajanlarının, formasyonlarını en uygun şekilde dağıtarak engel etrafında oluşacak dengesiz

dağılımını engellemek için bir yöntem geliştirdiler. Bu problem iki aşama olarak, bozulma ve yakınmasa olarak değerlendirdiler. İlk problem olarak engeli geçerken hız, yörünge ve konum bilgileri ile farklı dağılım kombinasyonlarını deneyen ve en uygun olanı seçen, diğer problem için çarpışmadan kaçınma işlemi sonrası dağılmış formasyonun istenilen formasyona gelmesi için ince levha eğrilerini (TPS) referans alan yöntemi oluşturdular. Çalışmalarının geleneksel yöntemlerle kıyasla 14.7% daha az enerji tükettiğini bildirdiler [109].



Şekil 2.20. 1) başlangıç fazı, 2) en yüksek bozulma, 3) yakınsama fazı [109].

2.5. MATEMATİKSEL MODELLER

2.5.1. Ağırlık Noktası

Çok ajanlı sürü çalışmalarında yaygın olarak kullanılan dinamik modellerden olan, ağırlık noktalı model, çift integral modeli olarak da adlandırılmaktadır.

$$\dot{p}_\iota \doteq v_i = \frac{1}{m_i} u_i, = 1,\ldots,\mathrm{N}, \tag{2.9}$$

$$x_c = P \sum_{i=1}^{N} x_i, \tag{2.10}$$

Burada $p_i$konum, v_i hız, m_i parçacık kütlesi, $u_i$kuvvet girdisi. Sistem durum uzay denklemi, $x_i^T = [p_i^T, v_i^T]$ olarak tanımlanmaktadır. Parçacıkların ağırlık merkezi x_c olarak tanımlanmaktadır.

$$\theta_i = \arctan(\dot{y}_i, \dot{x}_i), \tag{2.11}$$

Her i. parçacığın baş açısı θ_i olarak tanımlanmaktadır. Baş açısı parçacık yönelimini bulmak ve tespit etmek için gereklidir. Bu model birçok mühendislik ve biyolojik optimizasyon yöntemlerinde, ağırlık merkezi odaklı çalışmalar yapılmıştır [110–114].

2.5.2. Kendinden Tahrikli Parçacık

Model ilk olarak Vicsek [30] tarafından kendinden tahrikli parçacık modeli olarak ortaya atıldı. Parçacık hızlarının basit kural ve sinyal gürültüsünün olduğu dengesiz ortamlarda, parçacıkların kümelenmesi, taşınım ve form geçişini temsil etmek araştırmak için tanımlanmış modeldir. Modelin basit hali Czirok [115] tarafından tek boyutta verilmiştir. Model, her parçacığın sabit hızda hareket ettiğini ve her adımda yönelimlerin (baş açılarının) değiştiğini varsayar.

$$p_{ix}(t+1) = p_{ix}(t) + v \times \cos(\theta_i(t+1)), \tag{2.12}$$

$$p_{iy}(t+1) = p_{iy}(t) + v \times \sin(\theta_i(t+1)), \tag{2.13}$$

$$\theta_i(t+1) = \omega_i(t), \tag{2.14}$$

Her i. parçacık için, p_{ix} ve p_{iy} kartezyen koordinat sistemindeki x, y konumları, v öteleme hızları, θ_i, baş açısı (heading) olarak tanımlanmaktadır. Baş açısı belirlenmesinde kontrol girişi olarak, $\omega_i(t)$ parçacığın mevcut konumu ile komşularının yönüne bağlı olarak hesaplanır.

2.5.3. Parçacık Sürü Optimizasyon

Kennedy ve Eberthart (1995) tarafından, stokastik tabanlı topluluk optimizasyonu için doğrusal olmayan fonksiyonların optimizasyonunda kullanılan bir yöntem geliştirdiler [5].

Parçacığın elde ettiği en iyi çözümü sağlayan koordinatlar pbest, popülasyonda tüm parçacıklar için o ana kadar elde edilen en iyi çözümü sağlayan koordinatlar ise gbest olarak adlandırılır. Her iterasyon için pbest ve gbest bulunduktan sonra parçacığın konumu ve hızı kurala göre güncellenir.

$$V_s = c_1 \times rnd_1^t \times \left(p_{best_i}{}^k - p_i^t\right), \tag{2.15}$$

$$V_c = c_2 \times rnd_2^t \times \left(g_{best}{}^t - p_i^t\right), \tag{2.16}$$

$$V_i^{t+1} = w \times V_i^t + V_s + V_c \ , \tag{2.17}$$

$$X_i^{t+1} = x_i^t + v_i^{t+1}, \tag{2.18}$$

“c1” ve “c2”; öğrenme faktörleri olup, parçacığın konumu, en iyi değerlere (pbest ve gbest) yaklaştırır. “rand1” ve” rand2” ise 1’den küçük pozitif rasgele katsayılardır. “k”; iterasyon sayısı, “w”; eylemsizlik ağırlığıdır. Seçilecek “w” değeri 1’den küçük ve pozitif olacak şekilde her iterasyonda, iterasyon sayısına bağlı olarak azaltılmalıdır.

2.5.4. Kuş Sürüsü Optimizasyon

Meng vd. (2015), yaptıkları çalışmada; optimizasyon problemlerini çözmek amacıyla kuş sürü optimizasyon (KSO) çalışmasını önerdiler. Kuş sürülerindeki sosyal davranış ve etkileşimlerden elde edilen sürü zekasına dayalı algoritmada, kuşların temel üç davranışını, yiyecek arama, uyanıklık ve uçuş davranışlarını sürü zekasına modelleyerek beş basit kuralla ilişkili dört arama yöntemini formüle ettiler. Yaptıkları kıyaslama problemlerinin simülasyon ve karşılaştırmalarında yöntemin üstünlüğünü gösterdiler [116].

a) Yiyecek arama davranışı matematiksel formülü:

$$x_{ij}^{t+1} = x_{ij}^{t} + \left(p_{ij} - x_{ij}^{t}\right) \times rnd(0,1) + (g_{ij} - x_{ij}^{t}) \times S \times rnd(0,1), \quad (2.19)$$

b) Uyanık olma davranışını:

$$x_{ij}^{t+1} = x_{ij}^{t} + A1\left(\bar{x}_j - x_{ij}^{t}\right)rnd(0,1) + A2(\left(p_{kj} - x_{ij}^{t}\right)S)rnd(-1,1), \quad (2.20)$$

$$A1 = a1 \times \exp\left(-\frac{pFit_i}{sumFit + \varepsilon} \times N\right), \quad (2.21)$$

$$A2 = a2 \times \exp\left(\left(-\frac{pFit_i - pFit_k}{|pFit_k - pFit_i| + \varepsilon}\right)\frac{N \times pFit_k}{sumFit}\right), \quad (2.22)$$

c) Uçuş davranışı

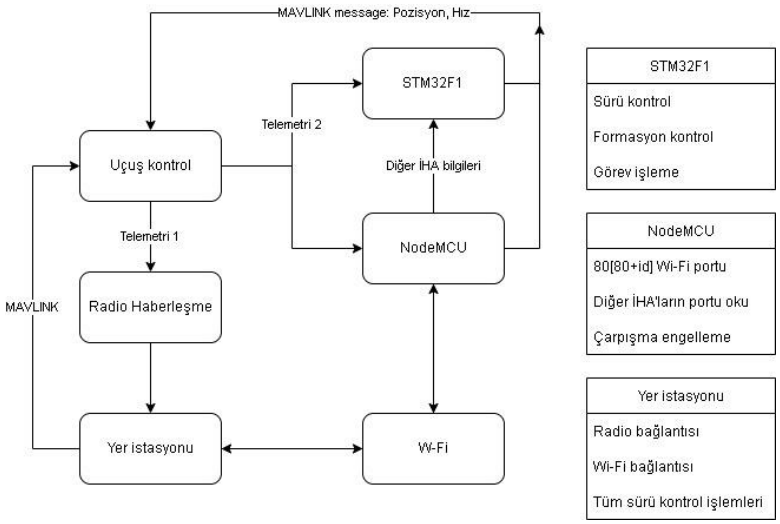
$$x_{ij}^{t+1} = x_{ij}^{t} + \text{randn}(0,1) \times x_{ij}^{t}, \quad (2.23)$$

$$x_{ij}^{t+1} = x_{ij}^{t} + (x_{kj}^{t} - x_{ij}^{t}) \times FL \times rand(0,1), \quad (2.24)$$

BÖLÜM 3

MATERYAL VE METOTLAR

Sürü insansız hava araçlarında kümelenme ihtiyacı topluluğun bir arada kalarak manevra kabiliyetini arttırmak askeri alanda savunma ve saldırı stratejilerine önemli etki oluşturmaktadır. Güvenlik çözümlerinde, sınır güvenliği gibi kontrol gerekliliğinde sürü İHA'ların sınırları belirlenmiş alanda insan gücüne gerek kalmadan sürü İHA'ların devriye görevini gerçekleştirebilmelerine olanak sağlamaktadır. İHA'ların birbiriyle etkileşimli olarak sürü zekasını işlemesi, sivil alanda arama kurtarma görevleri gibi zaman sınırı olan, geniş arama alanı ve zorlu şartlar altında gerçekleşen arama kurtarma görevlerinde zamandan ve insan gücünden önemli ölçüde tasarruf yaparak tüm alanı daha kısa sürede gezinme imkanı sağlamaktadır. Bu tez çalışmasında, birbirleriyle etkileşimde olan ve kümelenme kabiliyeti bulunan sürü İHA'ların, sürü algoritmalarının simülasyon ortamında ve sanal-gerçek entegre şekilde uçuşunu gerçekleştirebileceği sistem tasarımı gerçekleştirilmiştir. Bu bölüm kapsamında sürü İHA'ların kümelenme ve sürü kontrolünde kullanılan materyal ve kullanılan yöntemler açıklanmaktadır.



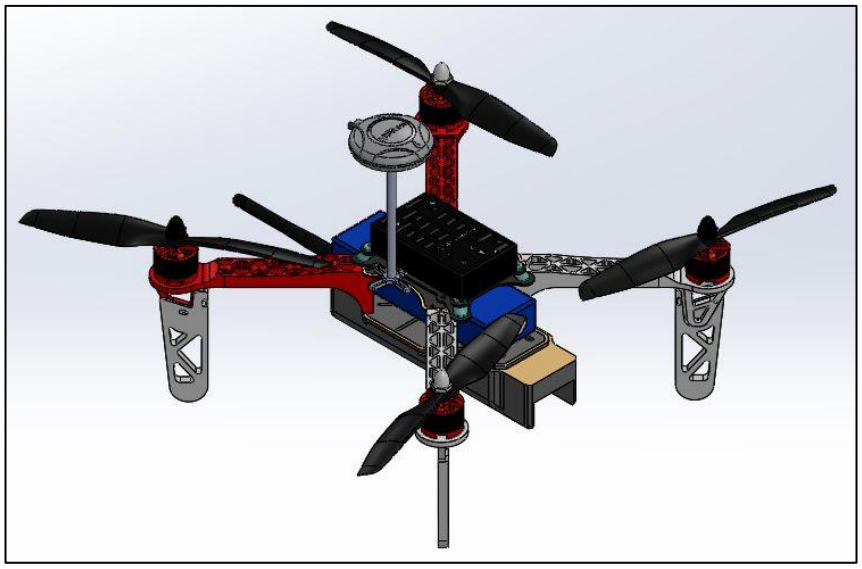
Şekil 3.1. Sistem akış diyagramı.

3.1. METARYEL

Bu çalışmada, modellenen fiziksel sistem entegrasyonunda, telemetri verileri ile merkezi komuta kontrol sistemi yönetimi kullanılmıştır. Uçuş kontrol sistemi STM32F4 ve yardımcı STM32F1 işlemcisi ile C/C++ yazılımı kullanılarak kontrol algoritması tasarlanmıştır. Telemetri bilgileri her üye için oluşturulan TCP/UDP bağlantılarından aktarılarak yer istasyonundan Wi-Fi ağına aktarılmıştır. Bu iletişim sistemi ile İHA'ların merkezi veya ağ erişimi olmaksızın kendi aralarında haberleşmesi ve görevi tamamlaması amaçlanmıştır. Alınan telemetri (konum, hız vb.) bilgileri STM32F1 ve nodemcu kontrol kartına aktarılmıştır. Alınan bilgiler doğrultusunda PSO ve meta-sezgisel algoritmaları uygulanmıştır. Bu işlemler ve beklentiler doğrultusunda materyaller oluşturulmuştur.

Fiziki sistem öncesinde yapılan simülasyon çalışması için mavlink haberleşmesi kullanılarak sanal drone oluşturulmuş ve python üzerinden kontrol yöntemleri uygulanmıştır. Gazebo ve Ros ile simülasyon uygulamaları yapılmıştır. Materyal oluşturulmadan önce simülasyon ortamında merkezi ve sürü lideri kontrol algoritmalarında kullanılacak Wi-Fi ağı simülasyon ortamında oluşturularak seçimlerin belirlenmesi desteklenmiştir.

Bilgisayar destekli modelleme ve simülasyon ortamında sürü dağılımında bazı parametrelerin gerçek dünya ile uyumsuzluğu söz konusudur. Bu parametreler; hava akışı ve iletişim ağı olarak belirlenebilir. Hava akışı yüksek veya alçak hava basıncı oluşturması sonucu irtifa kaybı, mevcut konumdan sapmalar oluşturabileceği ön görülmektedir. Aynı zamanda GPS konumlama sisteminin kullandığı uydu sayısı ve iletişim süresi mevcut konumun belirlenmesinde önemli bir etkendir. Birden fazla kullanılacak GPS modülü ile konum yenileme süresi ve doğruluğu iyileştirilebilmektedir. Bu problemler rota planlaması ve sürü hareket kabiliyetini doğrudan etkilemektedir. İletişim ağı probleminde ise, simülasyon ortamında her bir sürü üyesinin herhangi bir sinyal kaybı, gürültü ve haberleşme trafik yoğunluğu yok sayılmaktadır.



Şekil 3.2. Uçuş testleri için kullanılan İHA'nın katı model tasarımı.

Tasarlanan model F330 çerçeve referans alınarak modellenmiştir. Tasarımda ağırlık merkezi dikkat edilerek tasarlanmıştır. Uçuş kontrol ve GPS birimleri üst katmanda, telemetri ve sürü kontrol birimleri oluşabilecek parazitleri engellemek için alt katmanda tasarlanmıştır.

Şekil 3.3. Deneysel testler için kullanılan drone'ların montaj aşaması.

Dört rotorlu İHA modelinde racerstar br2212 980 kv fırçasız doğru akım motoru kullanılmıştır. Motor seçiminde dikkat edilecek konu uygun pervane ile tek bir motor için gereken kaldırma kuvveti ve verimlilik değerleridir. Bu değerler Çizelge 3.1'de motorun teknik özellikleri olarak belirtilmiştir.

Şekil 3.4. 980kv fırçasız dc motor.

Çizelge 3.1. Kullanılan 980 kv motor özellikleri.

Model	Voltaj	Pervane	Yük akımı	İtki(g)	Güç(w)	Verimlilik(g/w)	Ağırlık
Br2212 980kv	11.1 V	8045	8.1	535	90	5.9	50 (g)
		1045	10.6	710	118	6.0	

Çizelge 3.1'deki bilgiler, üretici sayfasındaki bilgilerinden alınmıştır. Kullanılan motorun teknik özelliklerine göre akım, kaldırma kuvveti (itki) ve verimlilik göz önünde bulundurularak F330 çerçevesi için 8045 pervane kullanılmıştır.

Çalışmada otonom uçuş kabiliyeti bulunan pixhawk 2.4 modeli kullanılmıştır. Kontrol kartı seçiminde dikkat edilen başlıca konular; piyasada bulunabilirliği, gömülü sistem için kullanılan ARM işlemcisi ile uçuş kontrol sistemlerinde sık tercih edilmesi,

UART haberleşme potu sayısı, PPM/SBUS desteği ve dahili güç dönüştürücüsü olması.

Şekil 3.5. Kullanılan uçuş kontrol kartı pixhawk 2.4.

Pixhawk 2.4 özellikleri;

- 32-bit ARM Cortex M4 işlemci
- MPU6000 IMU
- ST Micro 16-bit gyroscope
- ST Micro 14-bit ivmeölçer/pusula
- MEAS barometre
- 5x UART seri haberleşme portu
- PPM/SBUS sinyal desteği
- I2C, SPI, 2x CAN, USB
- DSM/2/X uydu (GPS)girişi
- RSSI (PWM girişi)
- 3.3V ve 6.6V ADC
- 14 PWM çıkışı
- Aşırı akım koruma (ESD)
- Ağırlık 38g
- Boyut: 81.5x15.5

Otonom uçuşları gerçekleştirebilmek için uçuş kontrol kartı küresel konumlama sistemine (GPS) veya yerel konumlama sistemine ihtiyaç duymaktadır. Konum tespiti için NEO-M8N GPS modülü kullanılmıştır. Dahili pusula (yön bulma) ve gelişmiş RF mimarisi ile parazit bastırma özellikleriyle yüksek performans sağlamaktadır.

- Eşzamanlı GNSS: 10 Hz
- GNSS yenileme oranı: 18 Hz
- Pozisyon Doğruluğu: 2.0m
- Yerleşik RTC kristal
- Alıcı tipi: 72 kanal
- Hassasiyet & Nav: –167 dBm
- Çap: 53 mm
- Ağırlık: 31 g

Sürülerin birbirleri ile yedek haberleşebilmesi (merkezi ve dağıtık haberleşmesi) için Wi-Fi bağlantısı ile ağ oluşturularak her İHA'nın telemetri bilgileri bu ağa nodemcu aracılığı ile aktarılmaktadır. Bu sayede İHA'ların komşularını ve komşularının konum, hız ve lider bilgilerine erişerek sürü zekası davranışı için gerekli algoritmaları yürütebilmektedir. Nodemcu ESP8266 tabanlı Wi-Fi modülü bulunan platformdur.

Şekil 3.6. Wi-Fi alt yapısı için kullanılan nodemcu.

- ESP8266 SDK tabanlı Wi-Fi modülü
- CH340G Seri dönüştürücü
- 10 GPIO
- GPIO pinleri PWM, ADC, I2C olarak kullanılabilmektedir

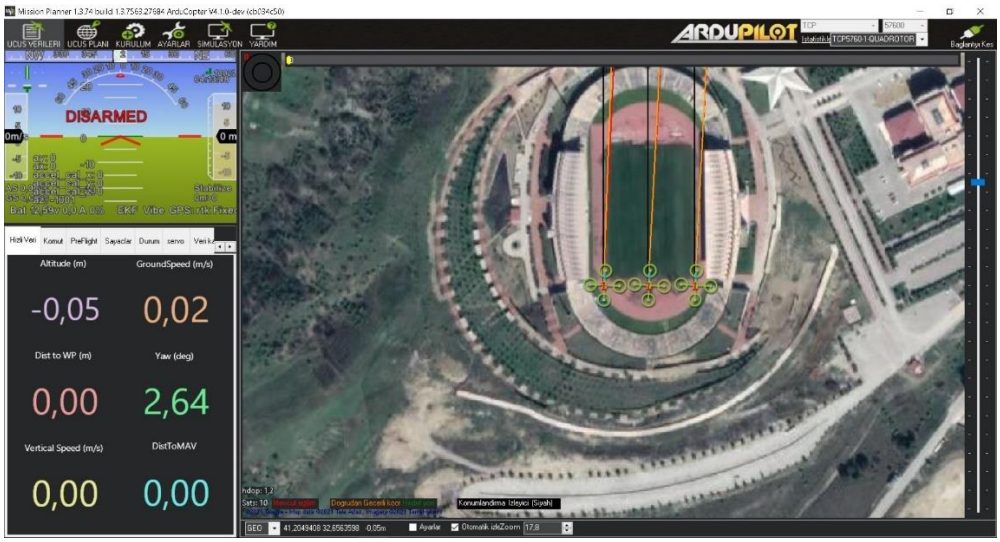
İHA'ların pil seçiminde, sistemin toplam ağırlığı, motorun itki kuvveti ve uçuş süresi dikkate alınarak seçilmiştir. Sistemde 3S 3400 mAh 45C lipo pil kullanılmıştır. Toplam motorun itki kuvveti 710g*4 ve anlık akım çekimi 8.1*4 olarak hesaplanmaktadır.

Pilin sağladığı 3400 mAh değerinden, dakikada sağladığı akım değeri 3.4*60 A olarak

hesaplanmaktadır. Pil dahil toplam gövde ağırlığı 1100 gr ve her motora düşen ağırlık 275 gr olarak belirlenmiştir. Motor teknik özelliklerine bakıldığında 8045 pervane için 535g itki oluşturduğunda 8.1 A çekmektedir. Bu değeri 275 g ağırlık kaldırması için oranlandığında 4.1 A olarak hesaplanmıştır. Tüm motorların toplam anlık akım 16.65 A olarak hesaplanmıştır.

$$\text{Uçuş süresi} = \frac{dakikadaki\_pil\_akımı}{motor\_akımı} = \frac{3.4 \times 60}{16.65} = 12.25\ dakika \qquad (3.1)$$

Hesaplanan uçuş süresi yaklaşık değer olmakla birlikte uçuş parametreleri ve hava şartlarına göre değişmektedir. Uygun PID parametreleri ile uçuş testlerinde, bu sürenin 15-20 dakika arasında değiştiği gözlemlenmiştir. SITL simülasyon ortamının, İHA'ların haberleşmesinde kullanılan mavlink protokolünü desteklemesi sebebi ile gerçeğe yakın simülasyon ortamı sağlamaktadır. Oluşturulan Belirlediğimiz ilgili porta bağlanarak o portta görev yapan hava aracının telemetri bilgisine ulaşarak port üzerinden veri alışverişi sağlanmaktadır.



Şekil 3.7. SITL ortamında 3 İHA bağlantısı.

SITL ortamında 3 İHA'nın TCP-5760, TCP-5770 ve TCP-5780 portlarında, mavlink verileri ile oluşturulan her ihanın, GPS, İMU ve diğer telemetri bilgileri gerçek ortam ile uyuşmaktadır. Her İHA'nın diğer İHA'lar ile iletişim kurarak konumlarını

optimizasyon algoritmaları ile iyileştirmeleri ve çarpışmadan kaçınarak görevlerini belirli yörünge izlenerek yerine getirilmesi amaçlanmaktadır.

Örnek SITL uygulamasında python programlama dili mavlink kütüphanesi ile TCP-57/60/70/80 portlarına bağlanılarak İHA'lar Karabük Üniversitesi ay yıldız stadyumu sahasını uçtan uca, ara nokta takibi ve çarpışma algoritmaları koşturarak çalıştırılmış ve görev sonlandırılmıştır.

Şekil 3.8. SITL simülasyonu eş zamanlı sürü görev testi.

Simülasyon sonucunda 3 hava aracı birbirleri ile bağlantılı olarak stadyumu, uçtan uca hareket noktaları oluşturarak taramış ve belirlenen hedef noktalarına sürü halinde hareket ederek ulaştıkları ulaşabilmişlerdir.

3.2. METOT

Sürü kontrol işlemleri için PSO yöntemi MATLAB tarafında matematiksel modellemeleri grafiksel olarak analiz edilmiştir. Sanal gerçek entegrasyonu gerçeğe yakın değerler oluşturması için 3D simülasyon testleri GAZEBO ve ROS ile İHA'lar modellenmiş ve gerçek verilere dayalı testler gerçekleştirilmiştir. Kümelenme

işlemleri için gerekli geometrik formüller oluşturularak MATLAB ve Python ile modellenerek testleri gerçekleştirilmiştir.

Çizelge 3.2. Simülasyon ve test işlemlerinde kullanılan bilgisayar özellikleri.

Değişken	Değeri
İşlemci	i7-8750H - 2.20GHz
RAM (rastgele erişim belleği)	16 GB
Ekran kartı	GTX 1050 Ti 4 GB
Disk belleği (SSD)	128 GB

3.2.1. Koordinat Dönüşümü

GPS verilerinden elde edilen, dünya üzerindeki enlem ve boylam değerleri ile iki nokta arasındaki mesafe Eşitlik3.4'deki Haversine formülü [120] ile bulunur.

$$a = sin^2(\Delta\psi/2) + \cos(\psi_1) \times \cos(\psi_2) \times sin^2(\Delta\lambda/2) + , \quad (3.2)$$

$$b = 2 \times atan2(\sqrt{a}, \sqrt{1-a}), \quad (3.3)$$

$$d = R \times b, \quad (3.4)$$

Eşitlik 3.2'de, ψ enlem değeri (latitude), λ boylam (longitude) değerleridir. Eşitlik 3.4'de R değeri dünyanın ekvatorda yarı çapı (R= 6 378 137 metre) değeridir. Yerel konumlar arası mesafe üç boyutta (x, y, z) Öklid uzaklığı Eşitlik 3.5'te verilen denklem ile bulunur.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}, \quad (3.5)$$

Sürü İHA'ların konum bilgilerinin yerel çerçevede işlenmesi için GPS verileri xyz kartezyen koordinat sistemine dönüştürülmüştür [121]. Bu işlem için dünyanın yaklaşık elipsoidal yapısından yararlanılarak çözümlenmektedir.

$$R_1 = 6378137, \tag{3.6}$$

$$R_2 = 6356752, \tag{3.7}$$

$$f = (a - b)/a, \tag{3.8}$$

$$e = f \times (2 - f), \tag{3.9}$$

R_1, dünyanın ekvatorda yarıçapı, R_2, kutuplarda yarıçapıdır.

$$\vartheta = a/\sqrt{1 - e \times \sin(\theta)^2}, \tag{3.10}$$

$$U_x(\theta, \beta, \alpha) = (\vartheta - \alpha) \times \cos(\theta) \times \cos(\beta), \tag{3.11}$$

$$U_y(\theta, \beta, \alpha) = (\vartheta - \alpha) \times \sin(\theta) \times \sin(\beta), \tag{3.12}$$

$$U_z(\theta, \alpha) = \left(\frac{b^2}{a^2} \times \vartheta + \alpha\right) \times sin(\theta), \tag{3.13}$$

Elipsoidal denklemler ile GPS verileri x, y ve z kartezyen dönüşümleri sağlandıktan sonra elde edilen konumlar WGS84 formatındaki dünyanın merkez noktası olarak kabul edilen referans noktasına olan çerçeve uzaklığıdır. Eşitlik 3.11, 3.12 ve 3.13'te θ enlem, β boylam, α yüksekliği temsil etmektedir. Eşitlik 3.14, 3.15 ve 3.16'da U_x^{ref}, referans GPS konumunu (başlangıç konumu) mevcut konum ile referans konumu (başlangıç pozisyonlarını bulmak için ilk GPS verileri) farkı hesaplanır.

$$x^* = U_x - U_x^{ref}, \tag{3.14}$$

$$y^* = U_y - U_y^{ref}, \tag{3.15}$$

$$z^* = U_z - U_z^{ref}, \tag{3.16}$$

$$x_{doğu} = -\mathrm{s}(\beta^r) \times x^* + \mathrm{c}(\beta^r) \times y^*, \quad (3.17)$$

$$y_{kuzey} = -\mathrm{c}(\beta^r) \times \mathrm{s}(\theta^r) \times x^* - \mathrm{s}(\theta^r) \times \mathrm{s}(\beta^r) \times y^* + \mathrm{c}(\theta^r) \times z^*, \quad (3.18)$$

$$z = \mathrm{c}(\theta^r) \times \mathrm{c}(\beta^r) \times x^* + \mathrm{c}(\theta^r) \times \mathrm{s}(\beta^r) \times y^* + \mathrm{s}(\theta^r) \times z^*, \quad (3.19)$$

Eşitlik 3.17, 3.18 ve Eşitlik 3.19'da s ve c kısaltmaları sırası ile sinüs ve kosinüs temsil etmektedir. Eşitlik 3.17'da $x_{doğu}$ değeri x+ eksenindeki doğu yönünü, Eşitlik 3.18'de y_{kuzey} değeri y+ eksenindeki kuzey yönünü ve Eşitlik 3.19'da z ise yüksekliği göstermektedir. θ^r ve β^r referans başlangıç konumudur.

Çizelge 3.3. İHA'ların başlangıç GPS konumları.

İHA NO	Latitude	Longitude	Altitude
1 - referans	41,20563259513842	32,65663238180573	384,9
2	41,20567901177595	32,65688184530396	384,9
3	41,20567699381704	32,65708837538899	384,9
4	41,20550546708149	32,65656266244528	384,9
5	41,20552968264788	32,65685502321499	384,9

Çizelge 3.4. GPS konumlarını kartezyen koordinat konumları.

İHA NO	X	Y	Z
1- referans	0,0	0,0	0
2	5,1554	-21,7252	0
3	4,9314	-39,0491	0
4	-14,1199	6,6481	0
5	-11,4303	-19,4754	0

Çizelge 3.3'de ilk İHA'nın konumu referans konumu olarak kabul edilerek işlemler yapıldığında Çizelge 3.4'de referans olarak kabul ettiğimiz ilk İHA'nın x,y ve z konumları 0 noktası olduğu görülmektedir.

3.2.2. Hedef Açının Belirlenmesi

Sürünün rotasyon, ara nokta takibi ve gezinme işlemleri sırasında belirlenen hedef nokta ile İHA arasındaki açının belirlenmesi gerekmektedir. Bu işlem için Eşitlik 3.22'deki denklem kullanılır.

$$y' = y2 - y1, \tag{3.20}$$

$$x' = x2 - x1, \tag{3.21}$$

$$\alpha = atan2(y', x'), \tag{3.22}$$

Burada, x' ve y' parametreler İHA'nın x ve y eksenindeki konumu ile hedef konum arasındaki fark, α açısı, [+180,-180] aralığında olduğundan bunu Eşitlik 3.23'de gösterildiği şekilde [0,360] aralığına alınır.

$$\alpha' = \begin{cases} \alpha + 360, & \alpha < 0 \\ \alpha, & \alpha \geq 0' \end{cases} \tag{3.23}$$

İHA'nın oryantasyon matrisinden elde edilen θ (yaw) değeri ile hedef açı α' değeri arasındaki fark β açısı rotasyon işlemine açı değeri olarak uygulanır.

$$\beta = |\theta - \gamma'|, \tag{3.24}$$

3.2.3. Normalizasyon

Kontrol algoritmalarının uygulanması sonucu, hata değerlerinin artmasını engellemek ve istenmeyen üst ve alt limitleri belirlemek için normalizasyon filtreleme işlemi yapılmaktadır. Bu sayede kontrol fonksiyonu çıktısının belirlediğimiz limitlerin dışına çıkmasını engelleyebilir veya belirli bir değer arasına indirgeyebiliriz. Örneğin: yol uzunluğu 100 metre olarak belirlenen bir rota planlama işleminde çarpışma engelleme fonksiyonunun çıktısı 0 iken normalizasyon işlemini [0,100] olarak ayarlandığında maksimum hız ile başlayıp yol sonunda hızı 0 olarak ayarlanır.

Çıkış değerini [min,max] aralığına ayarlamak için;

$$f(u') = \frac{u - \min(u)}{\max(u) - \min(u)}, \tag{3.25}$$

Çıkış değerini [0,1] ‘den [-1,1] aralığına ayarlamak için;

$$f(u') = 2 \times \frac{u - \min(u)}{\max(u) - \min(u)} - 1, \tag{3.26}$$

Eşitlik 3.26’da, [0,1] aralığında olan değer, [-1,1] aralığına dönüştürülmesi sağlanmaktadır.

3.2.4. Formasyon-Küme Oluşturma

Sürü İHA’ların kümelenme (formasyon) işlemlerinin modellenmesinde, MATLAB ve Python/matplotlib kütüphanesi ile gerçekleştirilmiştir.

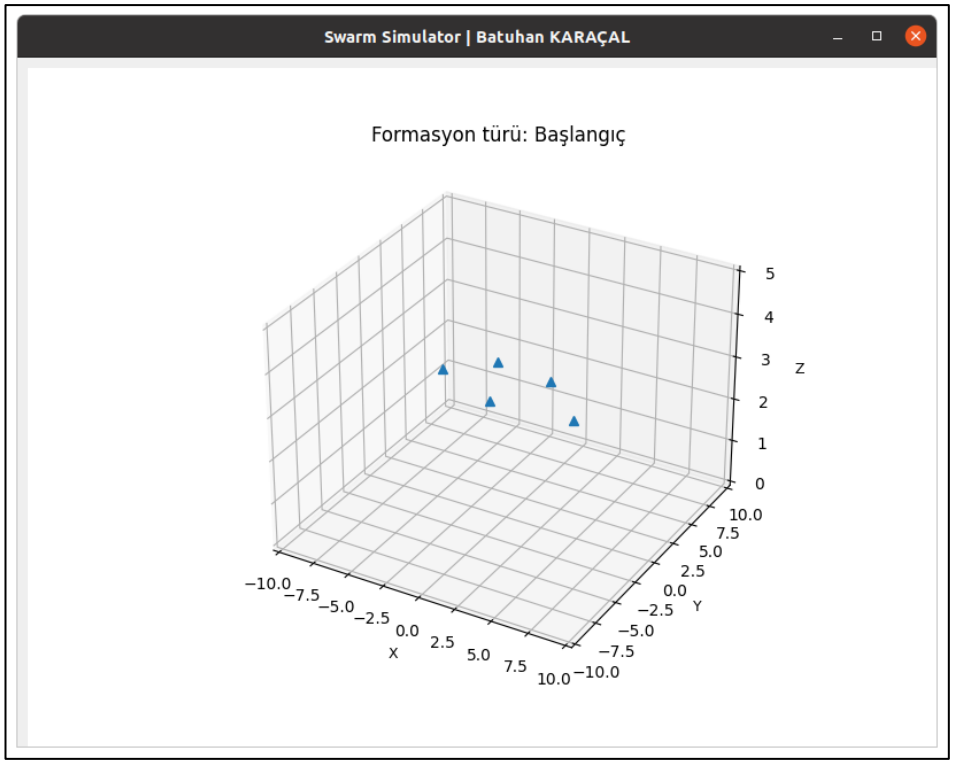
Formasyon türlerinin oluşturulması için mevcut konumları bilinen İHA’ların merkezi hesaplanması gerekir.

$$x_{center} = \frac{\sum_{i=1}^{N} x_i}{N}, \tag{3.27}$$

$$y_{center} = \frac{\sum_{i=1}^{N} y_i}{N}, \tag{3.28}$$

Çizelge 3.5. Simülasyon ortamındaki İHA’ların başlangıç konumları.

İHA NO	X	Y	Z
1	0,0	-2,0	3,0
2	2,0	2,0	3,0
3	-4,5	0,0	3,0
4	-2,0	2,5	3,0
5	5,5	-1,5	3,0



Şekil 3.9. Başlangıç konumlarındaki İHA'ların simülasyon ekranı.

3.2.1.1. Çizgi Formasyonu

Modelin temel amacı, İHA'ların merkezini referans alarak, merkeze bir İHA koymak ve kalan İHA'ları sağ ve sol kollara eşit şekilde dağıtmaktır.
γ: İHA'lar arası belirlenen formasyon mesafesi

Sol bölge için; i = 1, …, N/2, Sağ bölge için; i= N/2, …, N

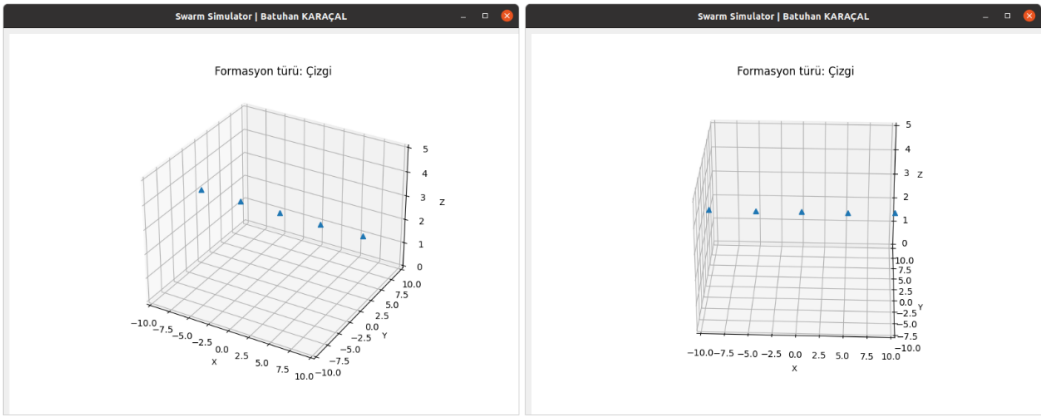
$$x_i = x_{center} + \gamma \times i, \tag{3.29}$$

$$y_i = y_{center}, \tag{3.30}$$

Formasyon deseninin belirlenmesinde, x veya y ekseni sabit kalarak, diğer eksende İHA'ların merkez noktasına belirlenen mesafe kadar öteleme işlemi yapılmıştır.

Çizelge 3.6. Çizgi formasyonu final İHA pozisyonları.

İHA NO	X	Y	Z
1	0,2	0,2	3,0
2	5,2	0,2	3,0
3	10,2	0,2	3,0
4	-4,8	0,2	3,0
5	-9,8	0,2	3,0



(a) (b)

Şekil 3.10. Çizgi formasyonunun simülasyon sonucu (a) perspektif, (b) ön.

Algoritma 1 Örnek çizgi formasyonu oluşturma

```
1  procedure cizgi_formation()
2      right = round(int(uav_count - 1)/2)
3      left = (uav_count-1) - right
4      for i < right and left
5          goal_x[i] = uav_x_center ± uav_per_dist*i
6          goal_y[i] = uav_y_center
```

Çizgi formasyonun için verilen algoritma 1'in Çizelge 3.5'te verilen başlangıç konumları ile oluşturulan parametre ve konum çıktıları Çizelge 3.7'da gösterilmiştir.

Çizelge 3.7. Çizgi formasyon algoritmasının 1 iterasyon konumları.

Değişken	Değeri	
uav_per_dist	5,0	
right	2	
left	2	
goal[0] = [x y]	0,2	0,2
goal[1] = [x y]	5,2	0,2
goal[2] = [x y]	10,2	0,2
goal[3] = [x y]	-4,8	0,2
goal[4] = [x y]	-9,8	0,2

3.2.1.2. Üçgen Formasyonu

Üçgen formasyonun oluşturulması için, merkez noktası referans alınarak sağ ve sol kollardan ekleme yapılarak oluşturulur. Gereksinim için 3 İHA ve 6. İHA'dan itibaren sağ ve sol kolların ortasına eşit şekilde eklenir.

Sağ bölge için;

$$x_i = x_1 + \frac{\gamma * i}{2}, \tag{3.31}$$

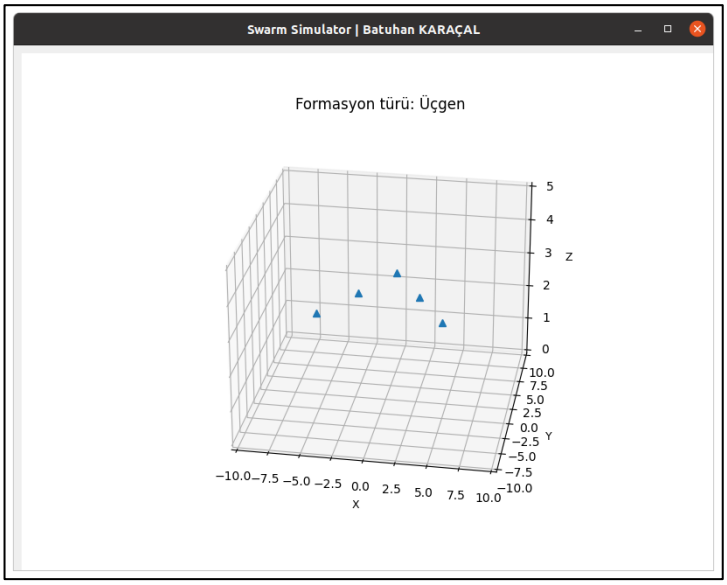
$$y_i = y_1 - \sqrt{i \times \gamma^2 - (\frac{i \times \gamma}{2})^2}, \tag{3.32}$$

Sol bölge için;

$$x_i = x_1 - \frac{\gamma * i}{2}, \tag{3.33}$$

Çizelge 3.8. Simülasyon sonuçlarında üçgen formasyonun final konumları.

İHA NO	X	Y	Z
1	-2,2779	0,1789	3,0
2	2,6912	0,2226	3,0
3	-4,8024	-4,1630	3,0
4	0,1729	4,5051	3,0
5	5,1993	-4,1360	3,0



Şekil 3.11. Üçgen formasyonu.

3.2.1.3. Kare Formasyonu

Kare formasyonun oluşumunda İHA (parçacık) sayısı 4*n+1 olmak zorundadır. İHA sayısının bu eşitlikte sağlanmadığı durumlarda artan İHA'ların formasyona dağıtılması gerekmektedir. Şekil 3.12'de 5 İHA'nın kare formasyon için (sol 4'lü kare), artan 1 İHA matematiksel modele göre sağ tarafa konumlandırılmıştır. Matematiksel modelinin oluşturulması için geliştirilen algoritma aşamaları;

- İHA sayısının 4 ile bölümünden oluşan bölümü bul
- Karenin sol köşe noktasını belirle

- Sol köşeden sağa bölüm sayısı kadar ekle
- Sol köşeden aşağı bölüm sayısı kadar ekle
- Sol alttan sağa bölüm sayısı kadar ekle
- Sağ üst köşeden aşağı bölüm sayısı kadar ekle

Sol köşenin belirlenmesi için; x ekseni sabit tutulup, y ekseninde köşeye ilerlenmektedir.

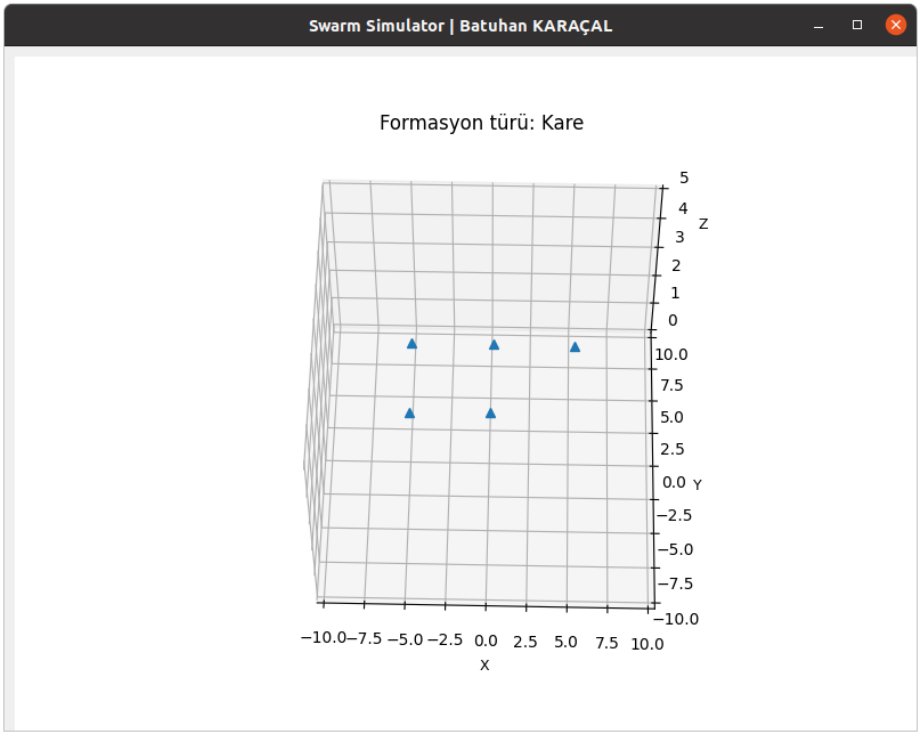
γ: İHA'lar arası istenilen formasyon mesafesi

$$x_{left\_corner} = x_{center} - \gamma, \tag{3.34}$$

$$y_{left\_corner} = y_{center} + \frac{N/4 \times \gamma}{2}, \tag{3.35}$$

Çizelge 3.9. Kare formasyon simülasyon sonucu oluşan pozisyon bilgileri.

İHA NO	X	Y	Z
1	-4,8255	2,7255	3,0
2	5,1962	2,6962	3,0
3	-4,8001	-2,3425	3,0
4	0,1994	2,7413	3,0
5	0,1291	-2,2740	3,0



Şekil 3.12. Kare formasyon simülasyon sonucu.

3.2.1.4. Beşgen Formasyonu

Beşgen formasyon formülünü oluştururken, bu formasyon için gerekli minimum İHA sayısı 5 olduğundan ve bundan fazla İHA bulunması durumunda İHA sayısının 5 ile bölümünden kalan sayı “bolum” olarak formüle edilmiştir.

$$corner_{dist} = \gamma/(2 \times \cos(54), \tag{3.36}$$

$$corner_{dist2} = corner_{dist} - \gamma \times \cos(54)/2, \tag{3.37}$$

$$k_p = \begin{cases} corner_{dist}\ ,\ bolum \neq 0, || \ uav\_count\ = 0 \\ corner_{dist2},\ bolum = 0, || \ uav\_count\ \neq 0' \end{cases} \tag{3.38}$$

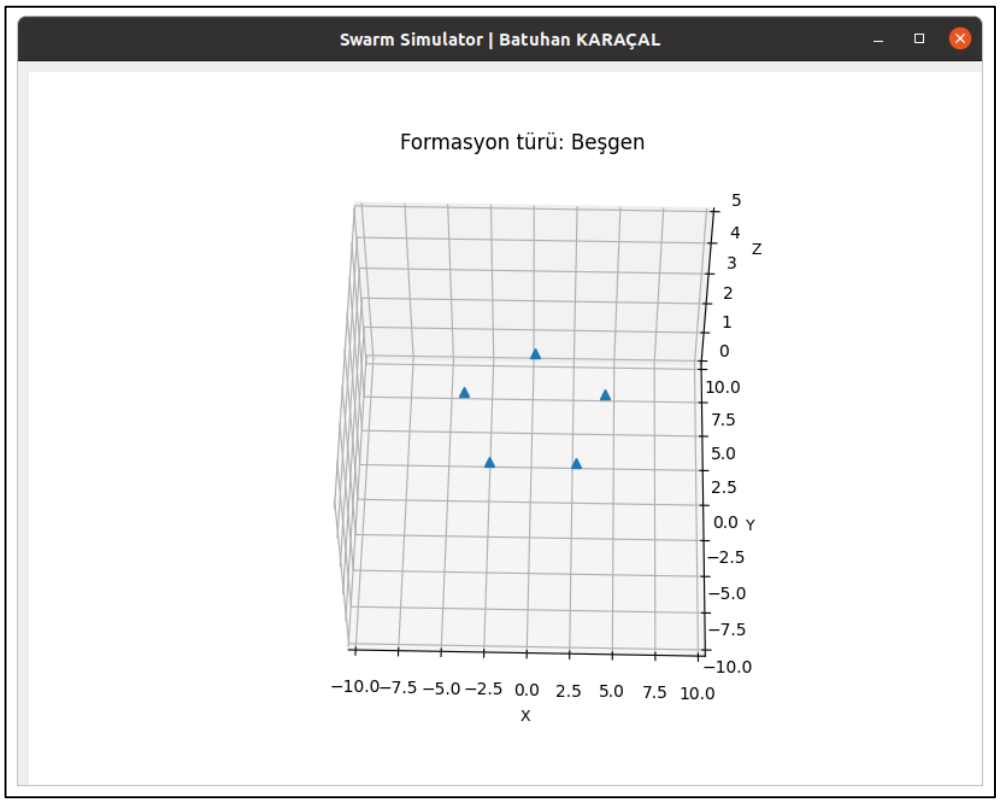
$$goal_{ix} = x_{center} + k_p \times \sin(\theta) * i, \tag{3.39}$$

$$goal_{iy} = y_{center} + k_p \times \cos(\theta) * i, \tag{3.40}$$

Burada, $corner_{dist}$, olarak oluşturulan değişken, oluşturulacak beşgen formasyonun köşegen uzunluğu, $corner_{dist2}$ formasyon noktaları oluşturulacak İHA sayısı 5'ten büyük olması durumunda ilk hesaplanan $corner_{dist}$ değerinden çıkarılarak formasyonun bozulmaması için ara noktalara yerleştirilir. Bu şekilde İHA sayısı 5'ten fazla olduğunda beşgen formasyonu herhangi bir bozulma olmadan çalışmaktadır.

Çizelge 3.10. Beşgen formasyon sonucu final pozisyonları.

İHA NO	X	Y	Z
1	-3,8750	1,5513	3,0
2	4,2797	1,5574	3,0
3	-2,3247	-3,2321	3,0
4	0,2000	4,3997	3,0
5	2,6741	-3,2312	3,0



Şekil 3.13. Beşgen formasyon sonucu.

3.2.1.5. Hilal Formasyonu

Hilal formasyonu belirlenen geometrik şekil yarım daire olduğu için, dairenin yay uzunluğundan (İHA'lar arası mesafe ile İHA sayısının çarpımı) çevre açı bulma formülü ile oluşturularak, İHA'lar arası mesafenin açıya dönüştürülmesi sağlanmıştır.

γ: İHA'lar arası belirlenen mesafe,
uav_count: Toplam İHA sayısı,
r: Oluşturulan çember yarıçapı
α: Yay uzunluğundan çevre açısının hesaplanması

$$r = \frac{((uav\_count \times 2) - 2) \times \gamma}{2 * \pi}, \tag{3.41}$$

$$\alpha = (\gamma \times 360)/(2 \times \pi \times r), \tag{3.42}$$

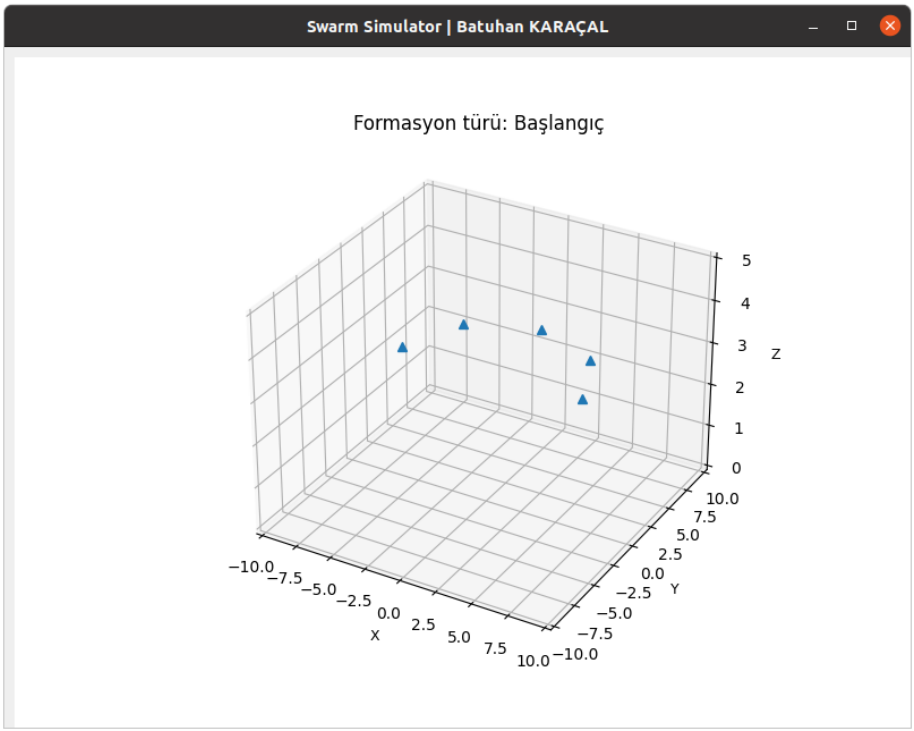
$$\theta_i = \alpha \times i, \tag{3.43}$$

$$goal_{ix} = x_{center} + r \times \cos(\theta_i), \tag{3.44}$$

$$goal_{iy} = x_{center} + r \times \sin(\theta_i), \tag{3.45}$$

Çizelge 3.11. Hilal formasyon sonucu final pozisyonları.

İHA NO	X	Y	Z
1	-4,3806	4,6719	3,0
2	6,5886	0,1998	3,0
3	0,2148	6,6039	3,0
4	4,7808	4,6744	3,0
5	-6,2245	0,2397	3,0



Şekil 3.14. Hilal formasyonu simülasyon sonucu.

3.2.1.6. V Formasyonu

V formasyonu oluşturmak için, referans noktasını İHA'ların merkezi noktası kabul ederek ilk noktayı burası olacak şekilde sağ ve sol kollara kalan İHA'ların formasyon noktası belirlenir.

N = (iha_sayisi -1)/2
i = 1, …, N,

$$goal_{0x} = x_{center}, \tag{3.46}$$

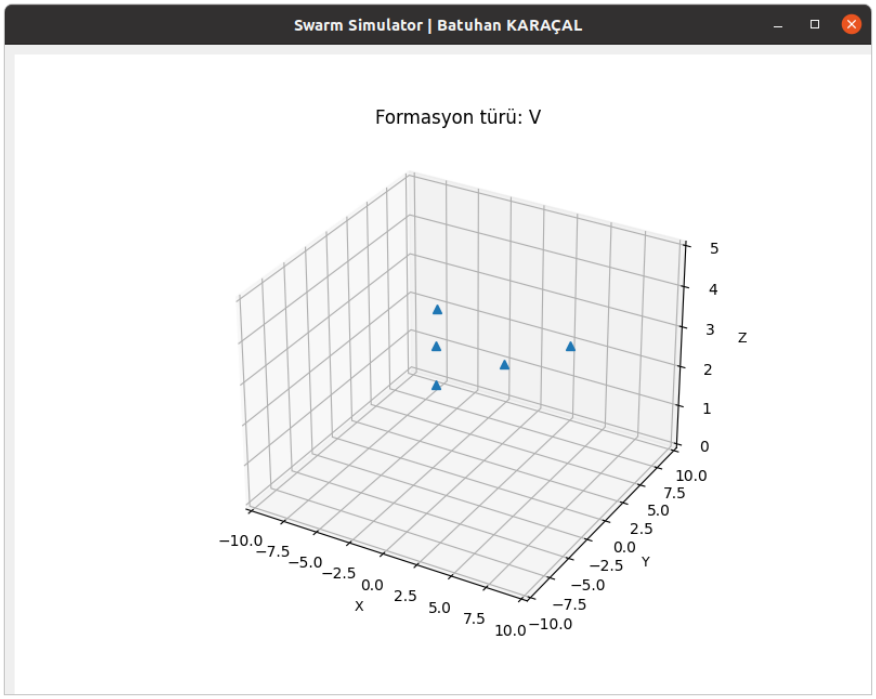
$$goal_{0y} = y_{center} - (\sqrt{\gamma^2 - \frac{\gamma^2}{2}}, \tag{3.47}$$

$$goal_{ix} = goal_{0x} \pm i \times \gamma/2, \tag{3.48}$$

$$goal_{iy} = \sqrt{i \times \gamma^2 - i \times \frac{\gamma^2}{2}} + goal_{0y}, \tag{3.49}$$

Çizelge 3.12. V formasyonu simülasyon sonucu final pozisyonları.

İHA NO	X	Y	Z
1	-4,7956	4,5160	3,0
2	2,7425	0,2365	3,0
3	-2,3000	0,2005	3,0
4	5,2008	4,5301	3,0
5	0,1971	-4,1312	3,0



Şekil 3.15. V formasyonu simülasyon sonucu.

3.2.1.7. Eşkenar Dörtgen Formasyonu

Eğer i=0 olmak üzere N'e kadar i tek sayı ise;

$$goal_{ix} = x_{center} + \gamma \times \cos(60), \tag{3.50}$$

$$goal_{ix} = y_{center}, \tag{3.51}$$

Eğer i çift sayı ise;

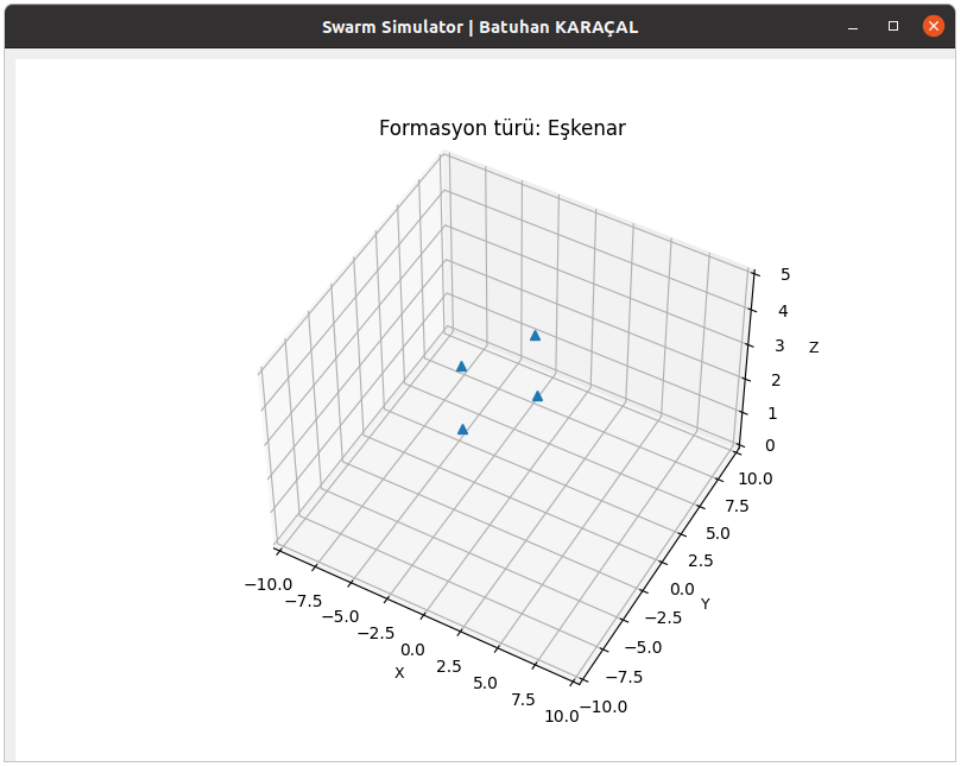
$$goal_{(i+1)x} = x_{center}, \tag{3.52}$$

$$goal_{(i+1)x} = y_{center} + \gamma \times \sin(60), \tag{3.53}$$

Eşkenar dörtgen kolları paralel olduğu için, mevcut döngü sayısı tek ise Eşitlik 3.51'de y ekseni sabit, çift olduğunda diğer köşegen için Eşitlik 3.52'de x ekseni sabit kalmaktadır.

Çizelge 3.13. Eşkenar dörtgen final pozisyonları.

İHA NO	X	Y	Z
1	-2,2958	0,8923	3,0
2	0,2032	-3,4842	3,0
3	0,2139	6,4775	3,0
4	2,7032	0,8461	3,0
5	-6,2245	0,2397	3,0



Şekil 3.16. Eşkenar dörtgen formasyonu.

3.2.1.8. Yıldız Formasyonu

Yıldız formasyonun (kümenin) oluşması için, iç içe 2 adet beşgenin oluşturulması ve bunun için en az 10 adet İHA'nın olması gerekmektedir.

γ: İHA'lar arası belirlenen mesafe,
iha_sayisi: Toplam İHA sayısı,
r: Oluşturulan çember yarıçapı
α: Yay uzunluğundan çevre açısının hesaplanması (İHA'lar arsı mesafe 2 katı alınır)

$$r = \frac{iha\_sayisi \times \gamma}{2 * \pi}, \tag{3.54}$$

$$\theta = (2\gamma \times 360)/(2 \times \pi \times r), \tag{3.55}$$

İlk iç daire ve içerisinde 5 nokta oluşturmak için, i= 1, …, 5,

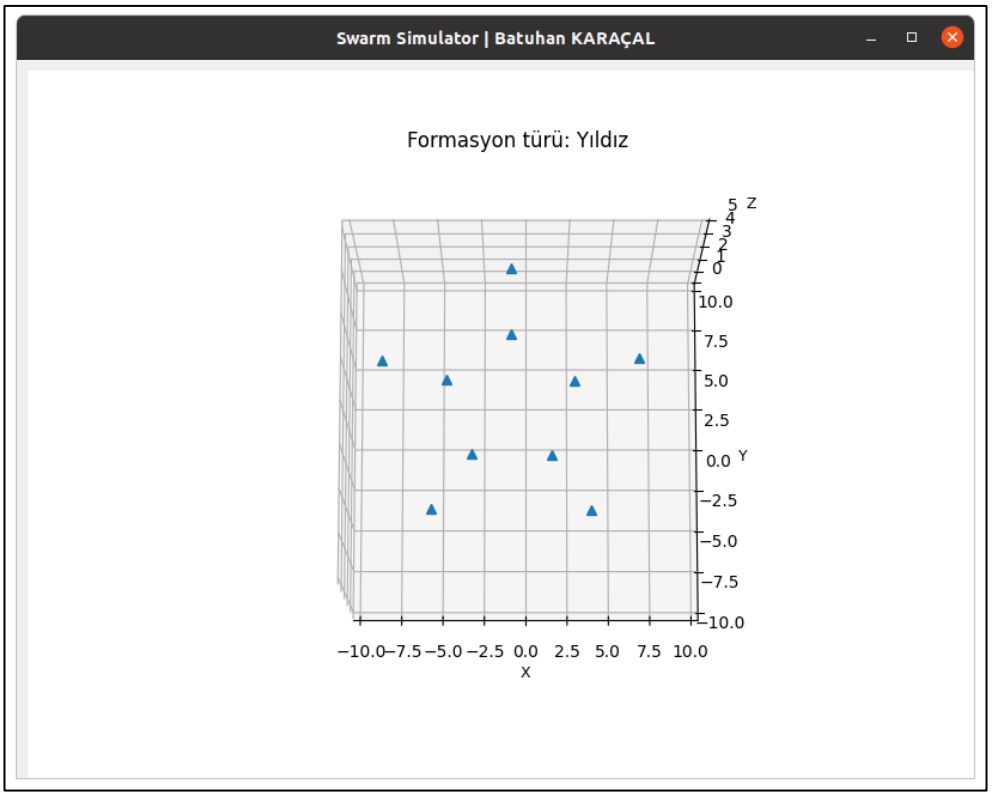
$$goal_{ix} = x_{center} + r \times \cos(\theta \times i), \tag{3.56}$$

$$goal_{iy} = y_{center} + r \times \sin(\theta \times i), \tag{3.57}$$

Dış daireyi oluşturmak için, i=5, …, 10, olacak şekilde aynı işlemler tekrarlanır.

Çizelge 3.14. Yıldız formasyonu final pozisyon bilgileri.

İHA NO	X	Y	Z
1	-5,5031	-5,1948	3,0
2	2,8169	2,3969	3,0
3	-8,3998	3,5960	3,0
4	-4,6429	2,4306	3,0
5	1,5064	-2,0103	3,0
6	-3,1523	-1,9639	3,0
7	-0,8674	5,1629	3,0
8	6,5877	3,7194	3,0
9	-0,8586	9,1055	3,0
10	3,7664	-5,2133	3,0



Şekil 3.17. Yıldız formasyon.

3.2.2. Formasyon Noktalarına Atama

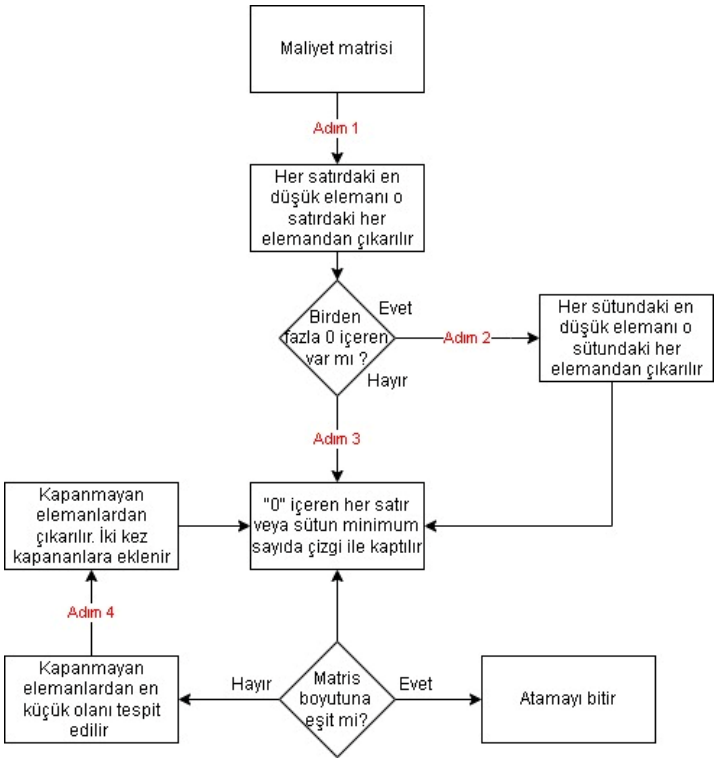
Kümelenme işlemlerinde belirlenen formasyon türünde, oluşturulan her nokta ile İHA'nın eşlenmesi gerekmektedir. Bu işleme eş zamanlı atama işlemi denmektedir. Her İHA'nın, her bir formasyon noktasına olan uzaklığı ile maliyet oluşmaktadır. Amaç, bu maliyeti en düşük sevide tutarak toplam maliyeti düşürmektir. Atama işlemi için hungarian (macar) algoritması kullanılmıştır.

$$Cost_{ij} = Pi - Pj, \tag{3.58}$$

Eşitlik 3.58'de gösterilen maliyet hesabı ile i. elamanın maliyeti hesaplanırken kendisi hariç diğer tüm elemanlara olan uzaklığı iki nokta arası uzaklık formülü ile bulunur ve maliyet matrisine aktarılır.

$$Goal_i = \min\,(Cost_i), \tag{3.59}$$

Eşitlik 3.59'da gösterilen her IHA'nın gitmesi gereken minimum uzaklık bulunarak hedef noktası belirlenir bu işlemden sonra matris sıralanır ve her IHA gitmesi gereken formasyon noktasını belirlemiş olur.



Şekil 3.18. Hungarian atama algoritması.

Atama işleminde Çizelge 3.5'te verilen İHA'ların başlangıç konumlarının Çizelge 3.6'da oluşturulan çizgi formasyon noktalarına olan uzaklığın maliyet matrisi formatında oluşturulması:

Çizelge 3.15. Çizgi formasyonu maliyet matrisi.

	HEDEF 1	HEDEF 2	HEDEF 3	HEDEF 4	HEDEF 5
İHA 1	2,2091	5,6462	10,4346	5,2802	10,0439
İHA 2	2,5456	3,6715	8,3952	7,0342	11,9365
İHA 3	4,7043	9,7021	14,7014	0,3606	5,3038
İHA 4	3,1828	7,5584	12,4149	3,6235	8,132
İHA 5	5,566	1,7263	4,998	10,4393	15,3942

Çizelge 3.15'te oluşturulan maliyet matrisinin Şekil 3.18'de atama algoritmasının ilk adımı (adım 1), her satırdaki en düşük maliyet ilgili satırdaki her satırdan çıkarılarak oluşturulan matris Çizelge 3.16'da gösterilmiştir.

Çizelge 3.16. Atama işleminin ilk adımı.

	HEDEF 1	HEDEF 2	HEDEF 3	HEDEF 4	HEDEF 5
İHA 1	0	3,4371	8,2255	3,0711	7,8348
İHA 2	0	1,1259	5,8496	4,4886	9,3909
İHA 3	4,3437	9,3415	14,3408	0	4,9432
İHA 4	0	4,3756	9,2321	0,4407	4,9492
İHA 5	3,8397	0	3,2717	8,713	13,6679

Şekil 3.18'de belirtilen atama algoritması minimum çizgi ile kapanan elemanların sayısı matris sayısına eşit olana kadar adım 3 ve 4 tekrarlanır. Belirtilen adımlar sonunda oluşan maskeleme matrisi (en uygun maliyet Çizelge 3.17'da gösterilmiştir.

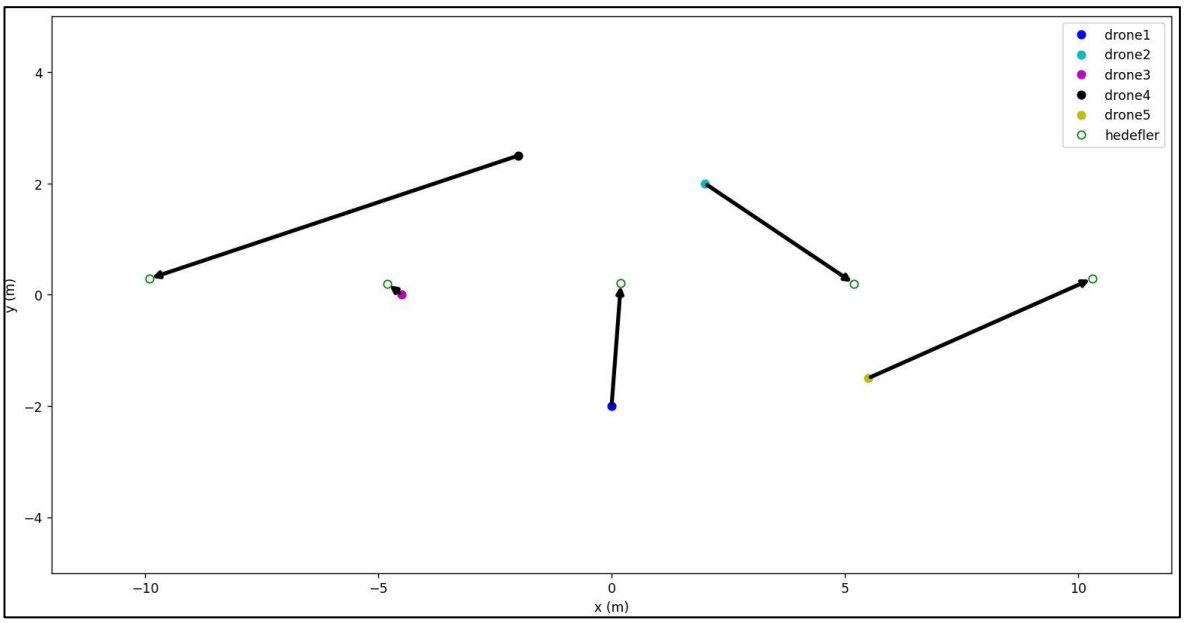
Çizelge 3.17. Atama işleminin son adımında oluşan matris.

	HEDEF 1	HEDEF 2	HEDEF 3	HEDEF 4	HEDEF 5
İHA 1	0	2,3112	3,8279	3,0651	2,8856
İHA 2	0	0	1,452	4,4826	4,4417
İHA 3	4,3497	8,2216	9,9492	0	0
İHA 4	0	3,2497	4,8345	0,4347	0
İHA 5	4,9656	0	0	9,8329	9,8446

Çizelge 3.17'de algoritmanın son adımı olarak minimum sayıda kapatılabilecek satır ve sütun sayısı matrisin 1 boyutuna (5) eşit olduğundan en uygun atama işlemi tamamlanmaktadır. Atama işlemini her satırdaki (İHA) sıfır içeren ilk eleman (HEDEF) atanır. Atanan her hedef numarası, birden fazla sıfır içeren satır olması durumu için sonraki İHA atamalarında kontrol edilir. İHA için ilk sıfır atanmış ise birden fazla sıfır olacağı anlamına gelir ve sonraki sıfır atanır. Bu işlem sonucunda İHA'lar için minimum yol uzaklığı ile görev dağılımı sağlanmaktadır.

Çizelge 3.18. Atama işlemi sonucu İHA'ların hedefler ile eşleştirilmesi.

İHA	HEDEF
1	1
2	2
3	4
4	5
5	3

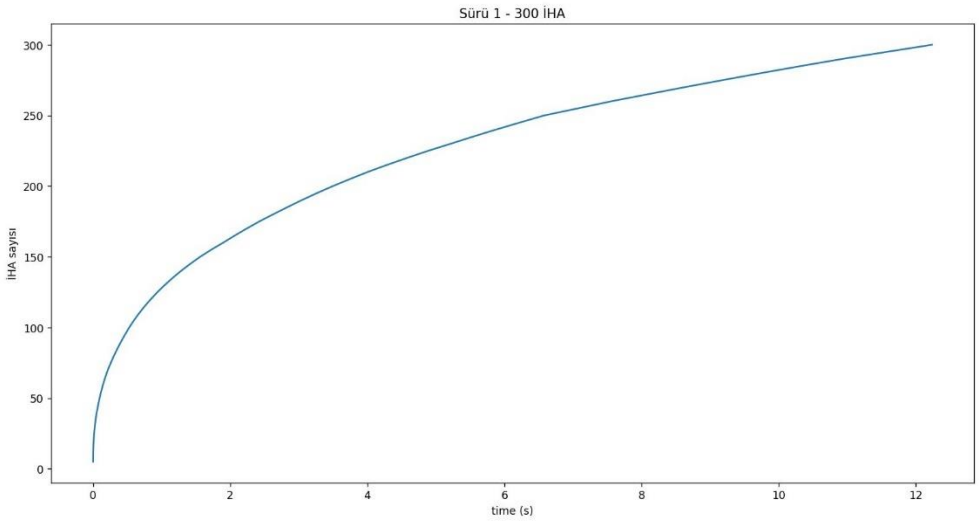


Şekil 3.19. Çizgi formasyonu atama işlemi.

Şekil 3.19'da Çizelge 3.5'te başlangıç konumları verilen İHA'ların Çizelge 3.6'da oluşturulan çizgi formasyon noktalarına atama işlemi gösterilmiştir.

Her İHA'nın istenilen görevlere en düşük maliyet ile atanması işleminde, İHA sayısının artması ile oluşturulan maliyet matrisindeki boyutu sebebiyle en uygun eşleştirmeleri bulmak için harcanan zaman artmaktadır. Şekil 3.20'de başlangıç koşulları Çizelge 3.15'te oluşturulan 5 İHA ile çizgi formasyonu 60 iterasyon uygulanarak her iterasyonda ilk oluşturulan maliyet matrisinin maliyet değerleri değiştirilerek yeni maliyet matrisi oluşturulmuştur. Her iterasyon İHA sayısının 5

artması ile i*5xi*5 boyutunda, son iterasyonda 300x300 boyutunda maliyet matrisi oluşturularak atama performansı değerlendirilmiştir.



Şekil 3.20. Atama işleminin 300 İHA performansı.

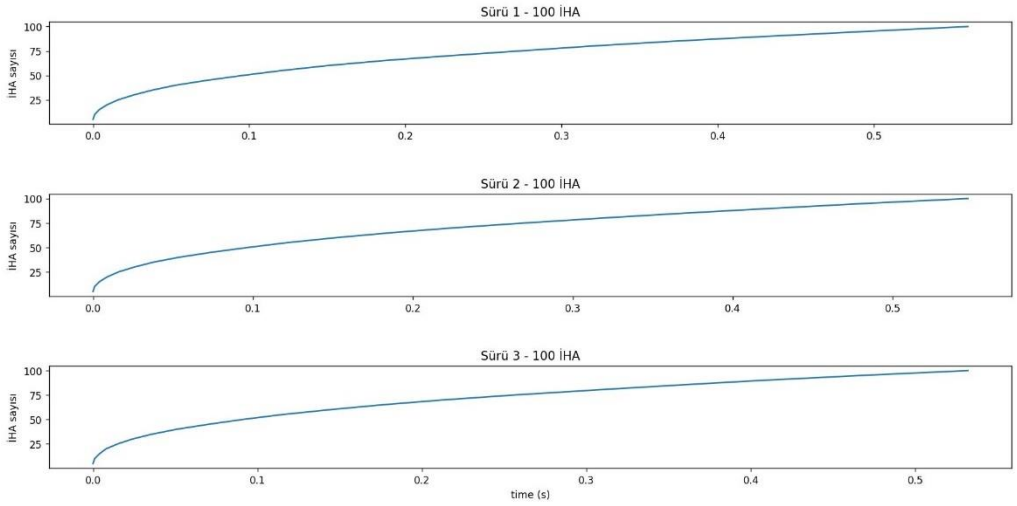
Performans testlerinde kullanılan bilgisayar performansı doğrudan etkilemektedir. Bu çalışmada yapılan testler için kullanılan bilgisayar özellikleri Çizelge 3.2'de verilmiştir. Şekil 3.20'de başlangıç durumunda 5 İHA ve 5 formasyon noktası (5x5) maliyet matrisinin İHA ve formasyon sayısının 300 olduğu durumda atama işlemini 12,2369 saniyede tamamlamaktadır.

Çizelge 3.19. Atama performansının 5 iterasyon çıktıları.

İterasyon (tekrar) sayısı	Saniye
1	12,3840
2	12,3345
3	11,7519
4	12,2720
5	11,91

Çizelge 3.19'da atama performansı için yapılan 5 iterasyon (tekrar) sayısının ortalaması alınarak atana işleminin 12,13 saniyede tamamlandığı gözlemlenmiştir.

İHA sayısının ve formasyon noktalarının (görevlerin) artması ile oluşan hesaplama süresinin azaltmak için İHA'ların sürü ayrılma işlemi sonrasında İHA'lar ve formasyon noktaları sürü merkezinden istenilen sürü sayısı kadar bölgelere ayrılarak İHA sürüleri ve formasyon noktaları kendi bölgelerinde atama işlemi gerçekleştirmektedir.



Şekil 3.21. Atama işleminin 3 sürü ile performansı.

Şekil 3.21'de 300 İHA'nın 3 sürüye bölünmesi ile gerçekleştirilen atama işleminde her sürü gurubu 100 İHA'nın yani 100x100 maliyet matrisi ile atama işlemini ortalama 0,5 saniyede gerçekleştirmektedir. Test sonuçlarına göre İHA sayısının artması ile gerçek zamanlı atama işleminde sürülerin ayrılmasının performansı önemli ölçüde etkilediği gözlemlenmiştir.

3.2.5. Formasyon-Küme Kontrol

Formasyon işleminin kontrol algoritması için [100] PD benzetimi, İHA'lar arası formasyon bozulması engellenmiştir.

$$f(u_i) = \sum_{j=1}^{N} \left(P_i - P_j - \delta_{ij}\right), \tag{3.60}$$

Kontrol girişi her birey i. eleman için diğer bireylerin konumlarının farkından belirlenen formasyon mesafesi çıkarılarak sürünün tüm bireyleri için uygulanarak formasyona girmesi beklenmektedir. Eşitlik 3.61'de δ_{ij} parametresi, bireyler arası mesafe matrisi i. elemanın j ile gösterilen diğer bireyler arasındaki olması istenen mesafeyi temsil etmektedir (formasyon oluşturma işlemlerinde, İHA'lar arası mesafe γ olarak belirtilmiştir.). Bu matris tanımlaması Eşitlik 3.61'de gösterilmiştir.

$$\delta_{ij} = \begin{matrix} 0 & p_1 - p_2 & p_1 - p_3 & p_1 - p_4 & p_1 - p_5 \\ p_2 - p_1 & 0 & p_2 - p_3 & p_2 - p_4 & p_2 - p_5 \\ p_3 - p_1 & p_3 - p_2 & 0 & p_3 - p_4 & p_3 - p_5, \\ p_4 - p_1 & p_4 - p_2 & p_4 - p_3 & 0 & p_4 - p_5 \\ p_5 - p_1 & p_5 - p_2 & p_5 - p_3 & p_5 - p_4 & 0 \end{matrix} \tag{3.61}$$

Eşitlik 3.61'de gösterilen kontrol algoritmasında formasyonun bozulmaması için İHA'lar arası olması gereken mesafe ile konum kontrolü sağlanmaktadır. Kontrol işlemini hız kontrolüne ve oluşabilecek dalgalanmaları engellemek ve PD tipi kontrolcü benzetimi yapmak için katsayı eklenir.

Formasyon kontrol matrisinin Çizelge 3.6'de çizgi formasyonu için

$$\delta_{ij} = \begin{matrix} 0 & -5 & 5 & 10 & -10 \\ 5 & 0 & 10 & 15 & -5 \\ -5 & -10 & 0 & 5 & -15, \\ -10 & -15 & 5 & 0 & -20 \\ 10 & 5 & 15 & 20 & 0 \end{matrix} \tag{3.62}$$

Hızın, yer değiştirmenin zamana göre türevi olduğu bilinmektedir.

$$v = \lim_{\Delta t \to 0} \frac{\Delta p}{\Delta t}, \tag{3.63}$$

$$\dot{p} = v = \frac{dp}{dt}, \tag{3.64}$$

Eşitlik 3.61'de verilen formasyon kontrol denklemine, hız ve Kp ve Kd sabitleri eklenerek düzenlenir.

$$V_f = K_p \sum_{j=1}^{N} \left(P_i - P_j - \delta_{ij}\right) + K_d \sum_{j=1}^{N} \left(v_i - v_j\right), \quad (3.65)$$

$$K_p = \frac{1}{N} \times K_{pp}, \quad (3.66)$$

$$K_d = \frac{1}{N} \times K_{dd}, \quad (3.67)$$

K_{pp}, ve K_{dd} değişkenleri kontrol parametrelerinde katsayı belirlemek oluşan hatayı ve kararsızlığı azaltmak için belirlenmiştir.

3.2.6. Çarpışma Engelleme

Sürüleri bir arada tutmak ve belirlenmiş görevleri yerine getirmesi için her İHA'nın çarpışmadan kaçınması gereklidir. Çarpışmadan kaçınma algoritmasının sürü içi ve dışı olarak iki ayrı bölümde incelenmektedir. İHA'ların sürü uçuşu, formasyon ve ara nokta takibinde oluşan olası bozulmadan kaynaklı belirlenmiş güvenli alan ihlali, diğer bölümde ise sensör tabanlı çarpışma engelleme işlemidir.

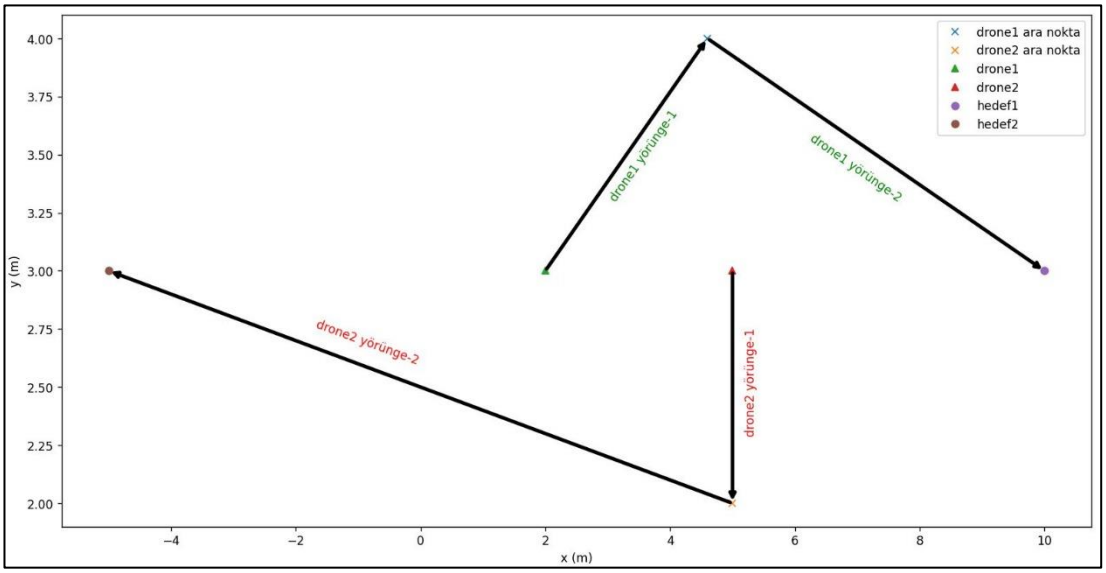
Çarpışma engelleme işleminde, birimlerin birbirlerine karşı vektörel (yönlü) bir kuvvet uygulanmaktadır. Eşitlik 3.69'da güvenli alan ihlali yapan İHA'lar tespit edilerek birbirlerini potansiyel bir itki kuvveti oluşturarak baş açıları ve hedef yönlerine göre iterler. Bu şekilde birbirlerinden ve sürü merkezinden uzaklaşarak çarpışmadan kaçınmaları sağlanır. Çalışma için geliştirilen potansiyel fonksiyon Eşitlik 3.68'de gösterilmektedir.

$$f(u)_{ij} = p_i + K_p\left(\sum_{j=0}^{N} \exp\left(-\frac{p_i - p_j}{\gamma^s}\right)\right) \times sinx\cos\left(\Delta\theta\right) \times \left(\gamma^s - p_{ij}\right), \quad (3.68)$$

$$p(t)_c = \begin{cases} f(u)_{ij}, & p_{ij} < \gamma^s \\ 0 & \end{cases}, \quad (3.69)$$

$$V_c = \frac{p(t)_c - p(t-1),}{\Delta t}, \quad (3.70)$$

Eşitlik 3.68'de γ^sgüvenli çarpışma mesafesi, γ^eitki kuvveti etkisi, K_p potansiyel fonksiyon sabiti, kontrol çıktısının etkisini kontrol etmek için eklenmiştir. Eksponansiyel fonksiyon, güvenli mesafeyi ihlal eden İHA'ların konum farkının güvenli mesafeye bölünmesi ile oluşan kuvvetin yumuşak geçiş sağlaması için eklenmiştir. Yön etkisini oluşturmak için baş açılarının farkını x ekseninde kosinüs, y ekseninde sinüs fonksiyonu olarak eklenmiştir. Bu sayede eksenel yön değişimi sağlanmıştır.



Şekil 3.22. Çarpışma engelleme fonksiyonu 1 iterasyon çıktısı.

Kp parametresi çarpışmaya katılan İHA sayısı olarak belirlendiğinde, eksponansiyel fonksiyonun aritmetik ortalaması olarak kuvvet etkisi göstermektedir.

Güvenli alan ihlali yapan her İHA'nın çarpışma engelleme algoritmasının işlenmemesi için ek bir kontrol algoritması eklenmiştir. Bunun amacı, baş açıları farklı doğrusal hareketine devam eden İHA'larn çarpışma yörüngesine yaklaştığında fakat çakışma olmayan durumların tespiti için kullanılmıştır.

Konumları bilinen iki nokta arasında (p1 ve p2) yörünge üzerinde farklı bir nokta (p3) kontrolü için;

$$c_p = \left(p_{3y} - p_{1y}\right) \times (p_{2x} - p_{1x}) - (p_{3x} - p_{1x}) \times (p_{2y} - p_{1y}), \quad (3.71)$$

$$d_p = (p_{3x} - p_{1x}) \times (p_{2x} - p_{1x}) - (p_{3y} - p_{1y}) \times (p_{2y} - p_{1y}), \quad (3.72)$$

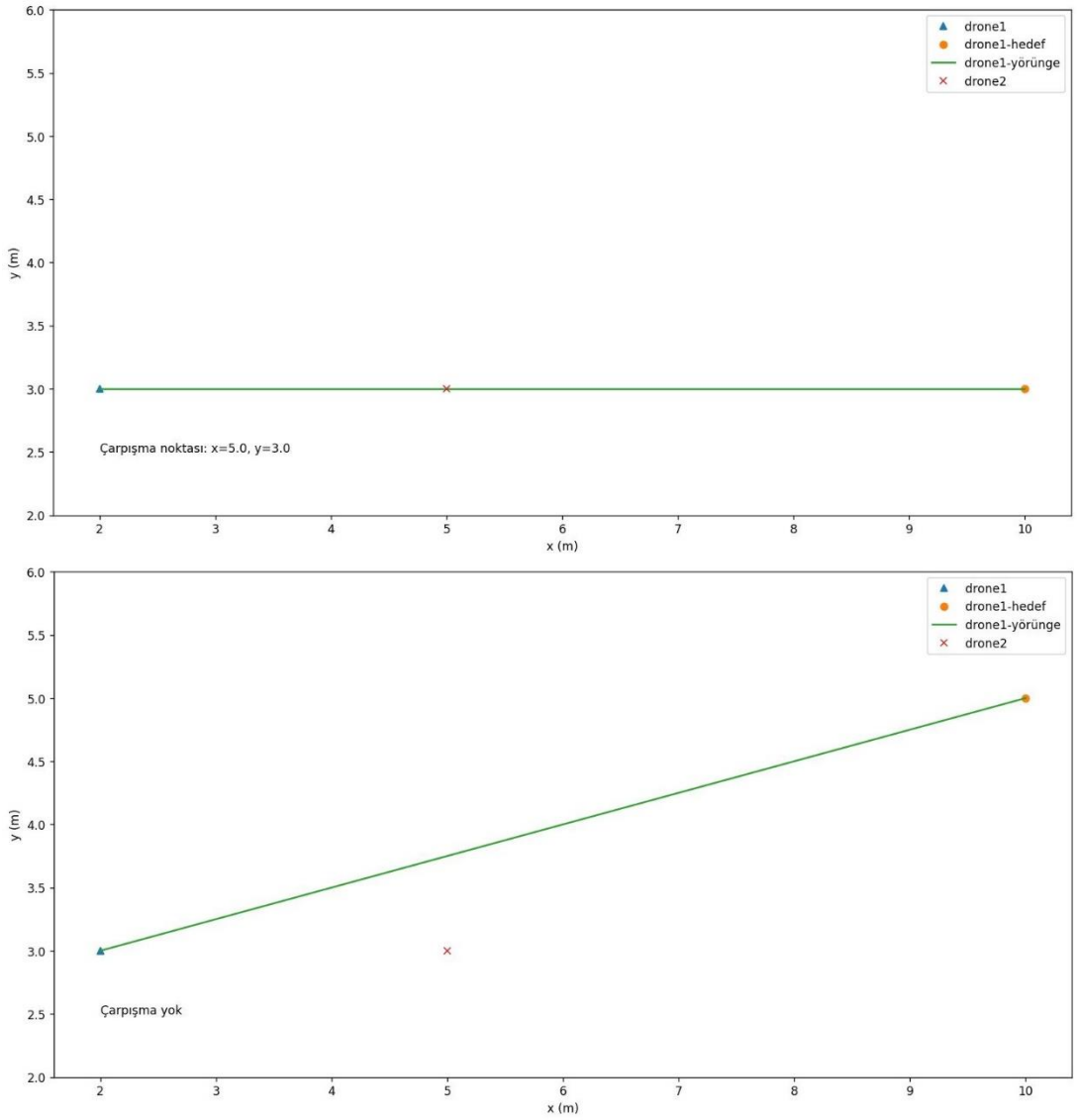
$$s_p = (p_{2x} - p_{1x}) \times (p_{2x} - p_{1x}) - (p_{2y} - p_{1y}) \times (p_{2y} - p_{1y}), \quad (3.73)$$

$$u = \begin{cases} 0, \ c_p > 0 \\ 0, \ d_p < 0 \\ 0, \ d_p > s_p \\ 1, \ diğer \end{cases}, \quad (3.74)$$

u çıktısı 1 olduğu durumlarda p1 ve p2 arasında p3 noktası yörünge üzerinde olduğu, diğer çıkışın 0 olduğu durumlarda yörünge üzerinde herhangi bir çakışma olmadığı anlaşılmaktadır.

Çizelge 3.20. Aynı yörünge üzerinde olan 3 noktanın parametreleri.

Parametre	Çıktı
p_1	x:2, y:3
p_2	x:10, y:3
p_3	x:5, y:3
c_p	0,0
d_p	24,0
s_p	64,0



Şekil 3.23. İHA ile hedef nokta arasında üçüncü nokta tespiti.

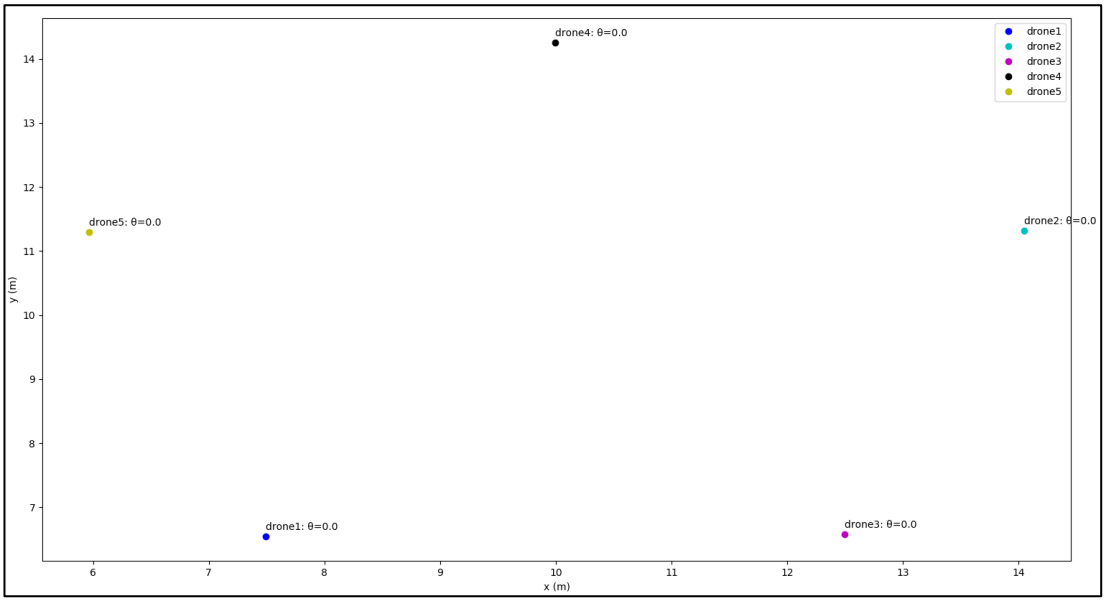
3.2.7. Sürü Rotasyon

İnsansız hava araçlarının belirli bir yörüngede ilerlerken dönüş yapması gerektiğinde örneğin; navigasyon görevi için her bir açılı navigasyon noktasını tamamlandıktan sonra 2 navigasyon noktası arasındaki konumun eğimine göre dönüş işlemi gerçekleştirerek ilerlemelidir. Aynı zamanda her IHA baş açısını düzeltmek için rotasyon işlemi yapmak zorunda. Hedef ile veya sürü lideri ile kendi konumu arasındaki konumundan eğimi hesaplanır ve bu eğime göre rotasyon işlemi gerçekleştirilerek baş açısı düzeltilir.

$$\delta_{ij} = \delta_{ij}\, x \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}, \tag{3.75}$$

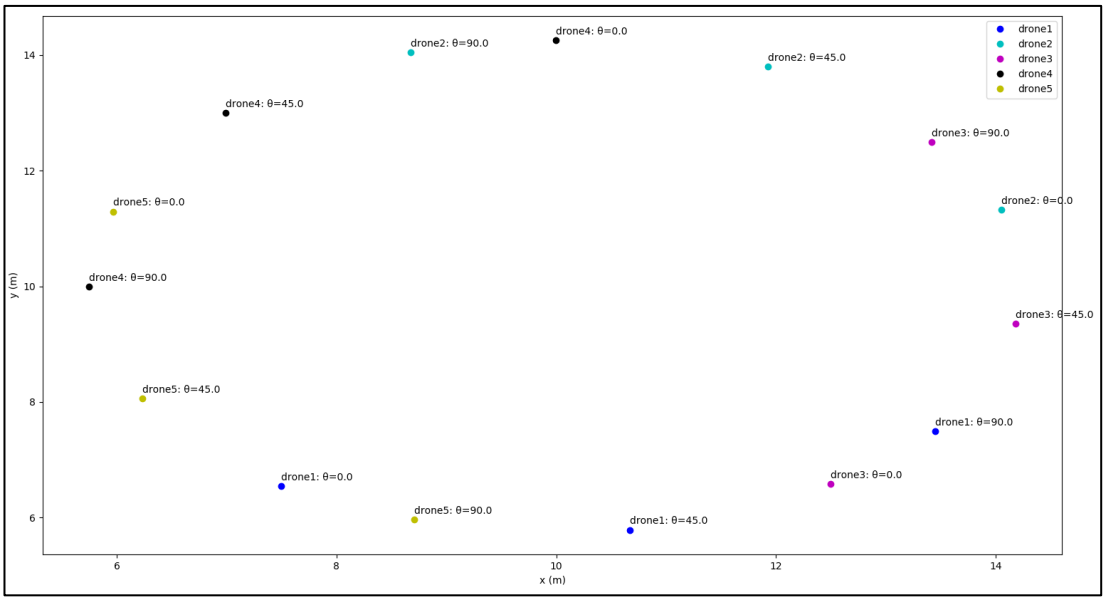
Rotasyon işleminin geniş açılı dönüşlerde kesin dönüş içermemesi ve sürünün belirlenen rotasyon açısı kadar dönüş işlemini gerçekleştirebilmesi için, rotasyon matrisini dönüş açısı ile oluşan final noktası sonrası bu iki nokta arasında geçiş noktaları oluşturulur. İHA'lar bu noktaları takip ederek rotasyon işlemlerini tamamlar.

Eşitlik 3.75'te verilen rotasyon matrisi, Eşitlik 3.61'de verilen formasyon kontrol matrisi ile çarpılarak rotasyon işleminde yeni formasyon kontrol matrisi oluşturulur. Rotasyon işleminde her İHA'nın diğer İHA'lar ile x ve y eksenlerindeki mesafesi değişmektedir.

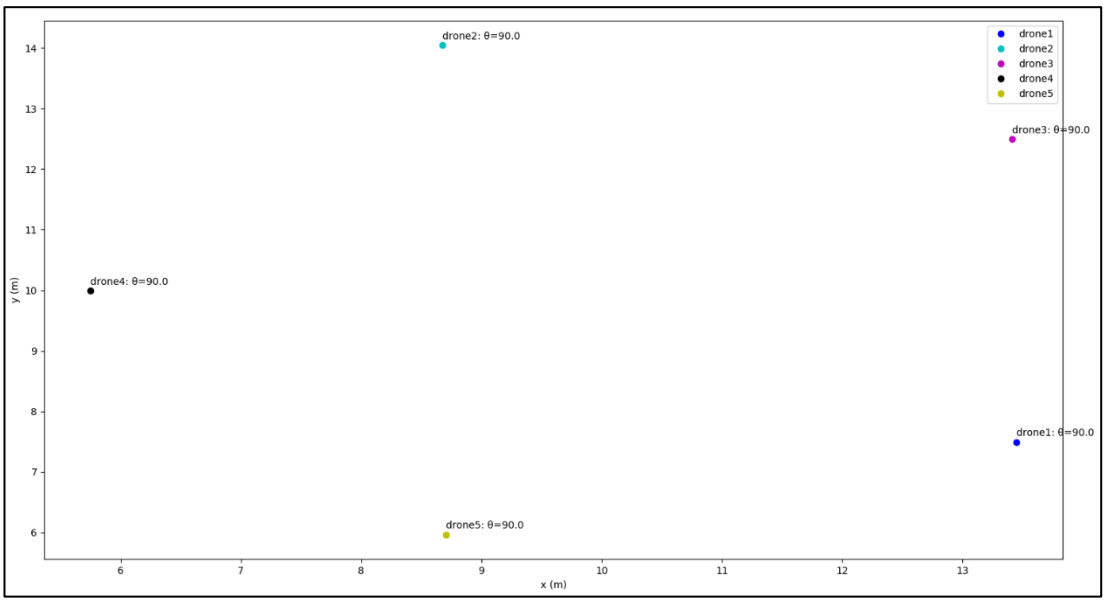


Şekil 3.24. Rotasyon öncesi İHA'ların beşgen formasyon konumları.

Eşitlik 3.75'te rotasyon açısı $\theta = 90°$, verildiğinde başlangıç ve bitiş dahil 3 nokta oluşturularak rotasyon gerçekleştirilir. Şekil 3.25'te rotasyon başlangıcındaki baş gezdirme açısı $\theta = 0°$, istenilen sürenin yarısında $\theta = 45°$ ve final noktasında ulaştığında baş gezdirme açısı $\theta = 90°$ olarak tamamlanmaktadır.



Şekil 3.25. Rotasyon işlemi sırasında İHA'ların izlediği yörünge.



Şekil 3.26. Rotasyon işlemi sonrası İHA'ların konumları.

Şekil 3.24'de İHA'ların baş gezdirme açısı y+ yönünde, 90° rotasyon işlemi sonrası Şekil 3.26'da baş gezdirme açısı y- yönüne bakmaktadır.

3.2.8. Navigasyon – Ara Nokta Takibi

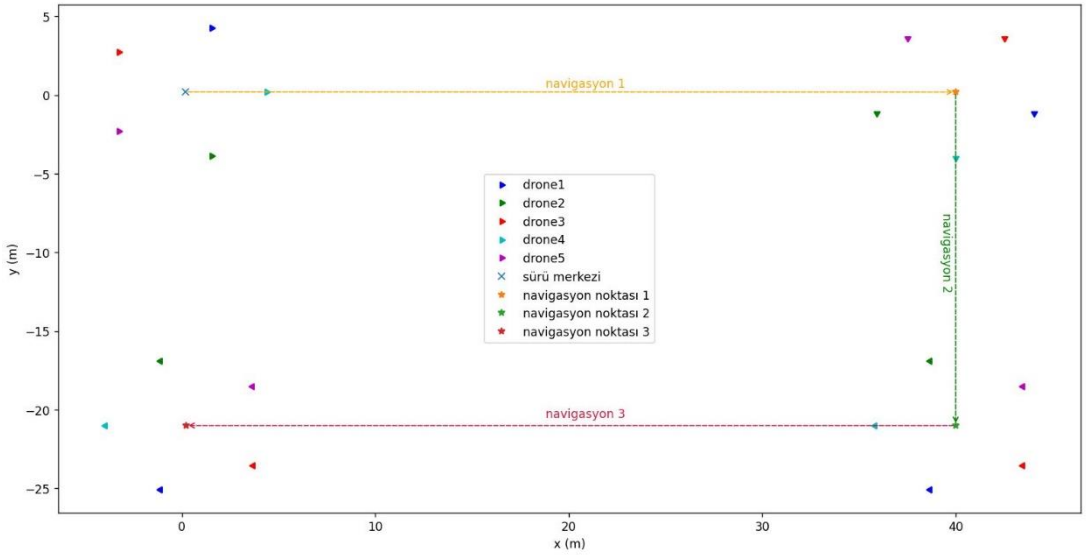
Sürü hedef noktası (P^*) ile arasında noktalar oluşturur ve bu noktalara uğrayarak ilerlemesi beklenir. Bunun için öncelikle sürünün ağırlık merkezi ($\bar{P}$) hesaplanır. Hedef ile (P^*) ile $\bar{P}$ arasındaki fark her İHA'ya eklenir. Limit sonsuza giderken sıfır oluyorsa sürünün o noktaya gitmesi beklenir.

$$\bar{p} = (\frac{1}{N}\sum_{J=1}^{N} p_j) - \ g\ , \tag{3.76}$$

$$p^*{}_i = p^*{}_i + \bar{P}\ , \tag{3.77}$$

$$u^t = \ Kp\big(p_j - \ p^*\big) + Ki\int_0^t (p_j - \ p^*)dt + Kd\ \frac{d(p_j - p^*)}{dt} + \gamma\ , \tag{3.78}$$

Sürünün merkez noktası ile hedef nokta arasındaki fark $\bar{p}$, her İHA'nın mevcut konumuna eklenerek $p^*{}_i$ İHA'nın hedef noktası PID kontrol yöntemine γ formason kontrol matrisi eklenerek navigasyon işlemi tamamlanır.



Şekil 3.27. Beşgen formasyonda olan sürü İHA'ların navigasyon ve rotasyonu.

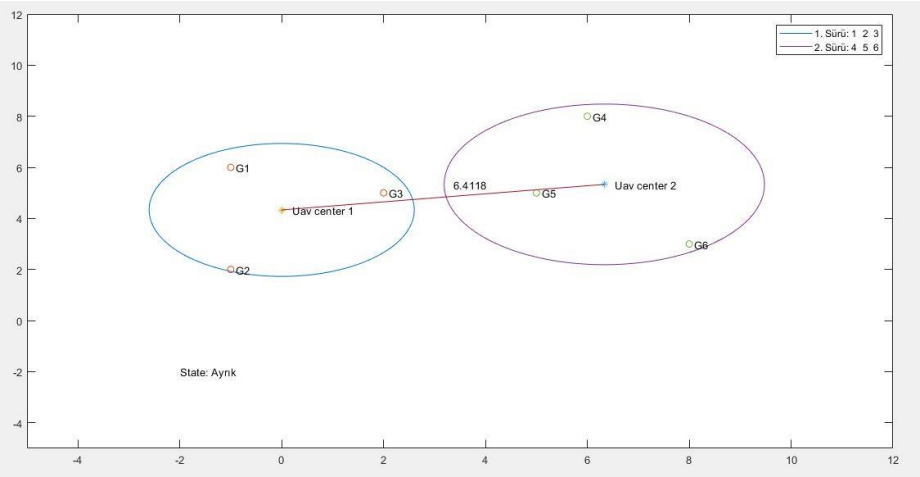
Şekil 3.27'de başlangıç konumları olarak Çizelge 3.10'da oluşturulan beşgen formasyonu kullanılmıştır. Eşitlik 3.77'de verilen denklem ile sürü İHA'ların merkezleri navigasyon noktalarına taşınmaktadır. Oluşturulan navigasyon noktaları arasındaki geçiş işleminde İHA'ların baş gezdirme açısı Eşitlik 3.24'te verilen denklem ile rotasyon açısı hesaplanmıştır. Rotasyon açısı Eşitlik 3.75'te verilen denklem ile navigasyon noktaları arası geçişte rotasyon işlemi sağlanmıştır.

3.2.9. Sürü Ayrılması

Sürü üyelerinin merkez veya lider tarafından birden fazlaya sürüye ayrılma işleminde, her sürü üyesi bir kontrol matrisinde saklanır. Bu matris her bir sürü gurubunu ve üyelerini temsil etmektedir. İlk sırada olan üye her zaman lider olarak tanımlanmaktadır. Bu bilgiler Wi-Fi ağına aktarılır ve merkezi kontrol yazılımında saklanır.

$$U_g = \begin{bmatrix} 2 & 1 & 3 \\ 6 & 4 & 5 \end{bmatrix}, \tag{3.79}$$

Eşitlik 3.79'da gösterilen 2xN/2 boyutundaki İHA numaralarının olduğu örnek sürü matrisi, 1. satır 1. sürüyü ve ilk sütununda olan 2 numaralı İHA'nın o sürünün lideri olduğunu belirtmektedir. Satırdaki kalan diğer İHA numaraları sürünün bireyini temsil etmektedir.



Şekil 3.28. MATLAB ortamında sürü ayrılma işlemi.

Şekil 3.28'de iki farklı sürüye ayrılan İHA'ların MATLAB ortamında modellemesi görülmektedir. Sağ üst köşeden sürü numaraları, sürünün sınırlarını temsil eden çizgiyi ve o süreye ait üyeler gözükmektedir. Tek boyutta olmayan sürü matrisi olduğundan sürü durumu ayrık olarak belirtilmiştir.

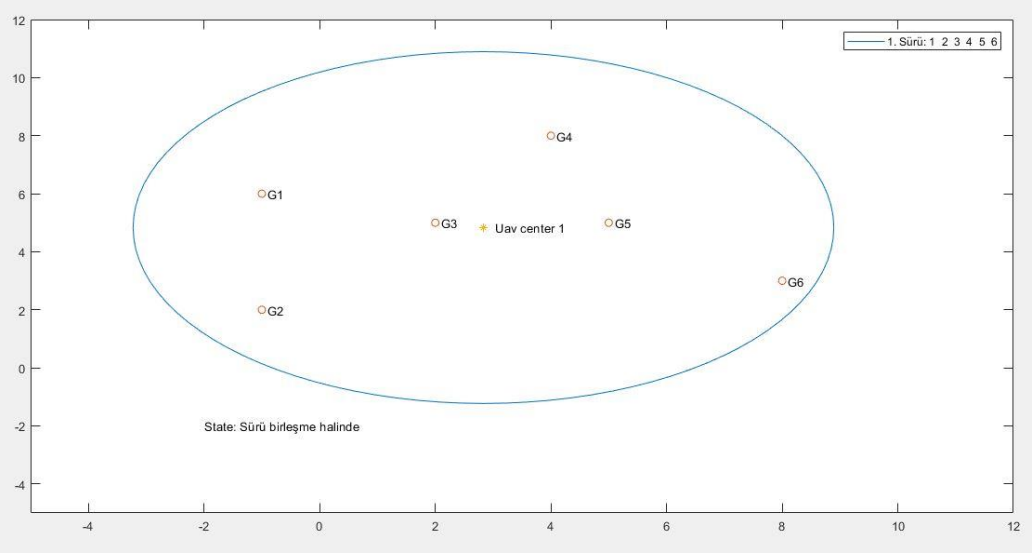
3.2.10. Sürü Birleşmesi

Birden fazla sürüye ayrılan İHA'ların tek bir sürü gurubuna dahil edilme işlemine sürü birleştirme işlemi için Eşitlik 3.80'de gösterilen sürü durum matrisini tek boyuta indirerek matris düzenlenmelidir.

$$U_g = [2 \quad 1 \quad 3 \quad 6 \quad 4 \quad 5], \tag{3.80}$$

Sürü birleştirme işleminde oluşan 1xN boyundaki matris oluşturulurken 1. satırdaki sürü üyeleri sabit kalarak diğer üyeler buna dahil edilir. Bu şekilde birleşen sürelerin lideri 1. sıra da olan sürü lideri olarak devam etmektedir.

Güncellenen matris sonrası kontrol algoritması çalıştırıldığında 2 ayrı sürünün tek sürü haline geldiği görülmektedir.



Şekil 3.29. Sürü birleşmesi simülasyon sonucu.

Birleşme işlemi sonrası MATLAB ortamında uygulanan algoritma, yeni oluşan tek boyutlu sürü sınırlarını ve sürüye ait bireyleri temsil etmektedir. Sürü matrisinin tek boyutta olmasıyla sürü durumu (state) birleşme durumunu bildirmektedir.

3.2.11. Yörünge Planlama (Trajectory)

Yörünge planlama işlemi istenilen zaman aralığında hedef noktaya ulaşmak için limitler dahilinde birim zamandaki hız değerinin belirlenmesi ile hesaplanır. Hızdaki değişimlerin sert olmaması için MJT (Minimum Sarsıntılı Yörünge) [117] algoritması kullanılmıştır.

$$M = \int_0^{tf} ||\dddot{x}||^2 dt, \tag{3.81}$$

Eşitlik 3.81'de ifade edilen tf, hareketin tamamlanacağı süredir.

$$x(t) = x_0 + (x_f - x_0)(6\tau^5 - 15\tau^4 + 10\tau^3), \tag{3.82}$$

$$y(t) = y_0 + (y_f - x_0)(6\tau^5 - 15\tau^4 + 10\tau^3), \tag{3.83}$$

$$\tau = \frac{t}{tf}, \quad 0 \leq \tau \leq 1, \tag{3.84}$$

$$V(jerkx) = \dot{x(t)} = x_f(30\tau^4 - 60\tau^3 + 30\tau^2), \tag{3.85}$$

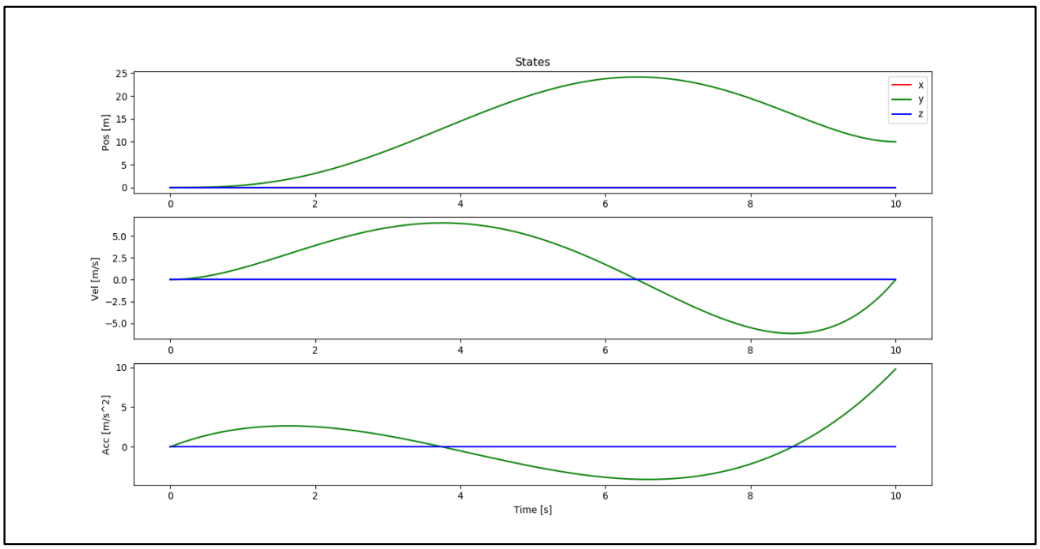
$$V(jerky) = \dot{y(t)} = y(30\tau^4 - 60\tau^3 + 30\tau^2), \tag{3.86}$$

Yörünge işlemi için giriş parametreleri:

Başlangıç pozisyonu : [0, 0, 0]
Hedef pozisyonu : [0, 10, 0]
Hedefe ulaşılması gereken süre (Tf) : 10
Fmin = 5 [m/s**2] IHA kinematik modeli minimum kuvvet
Fmax = 25 [m/s**2] IHA kinematik modeli maksimum kuvvet

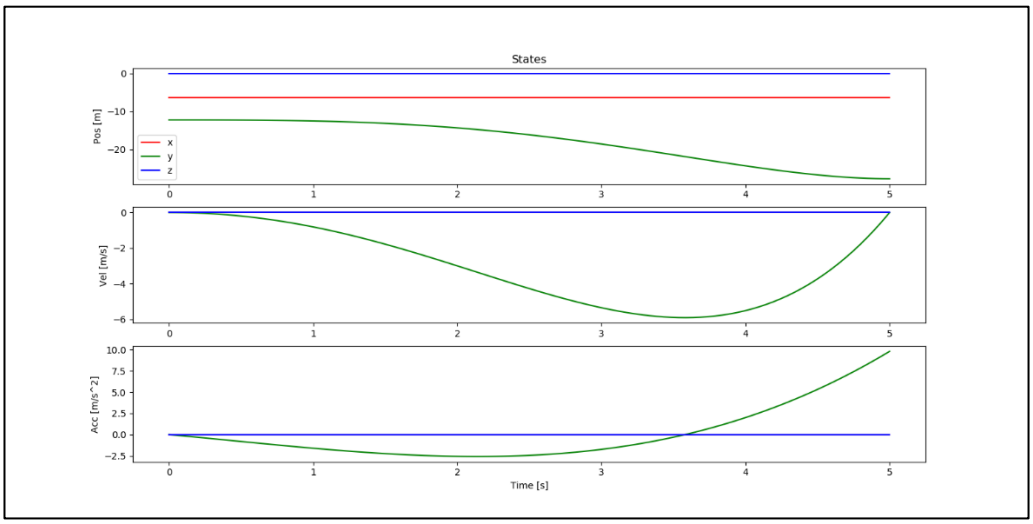
Wmax = 20 [rad/s] maksimum açısal hız

minTimeSec = 0,02 [s] birim zaman aralığı (hız değerleri IHA ya bu zaman aralığında uygulanacak)



Şekil 3.30. MJT algoritmasının simülasyon ortamında uygulanması.

Başlangıç ve hedef pozisyonları değiştirildiğinde yeni yörünge Şekil 3.31'de gösterilmiştir.



Şekil 3.31. Trajectory parameter değişikliği sonrası.

Bezier eğrileri [118] ile formasyon noktalarında navigasyon işlemi ve kontrol noktaları belirlenmiş ara noktaların takip işleminde kullanılmıştır.

Analitik yöntem ile kontrol noktaları P_0, P_1, P_2, P_3 verildiğinde

$$C(t) = n\sum_{i=0}^{n}\binom{n}{i}t^i(1-t)^{n-i}P_i, \quad 0 \le t \le 1, \tag{3.87}$$

Eğrilerin yapısının belirlenmesi, n. derece Bezier eğrilerinin n+1 kontrol noktası olmalıdır. Burada n derecesi eğrinin yapısını temsil etmektedir.

Doğrusal Bezier eğrisi için 1. dereceden 2 kontrol noktasının oluşturulması,

$$C(t) = (1-t)P_0 + tP_1\,, \tag{3.88}$$

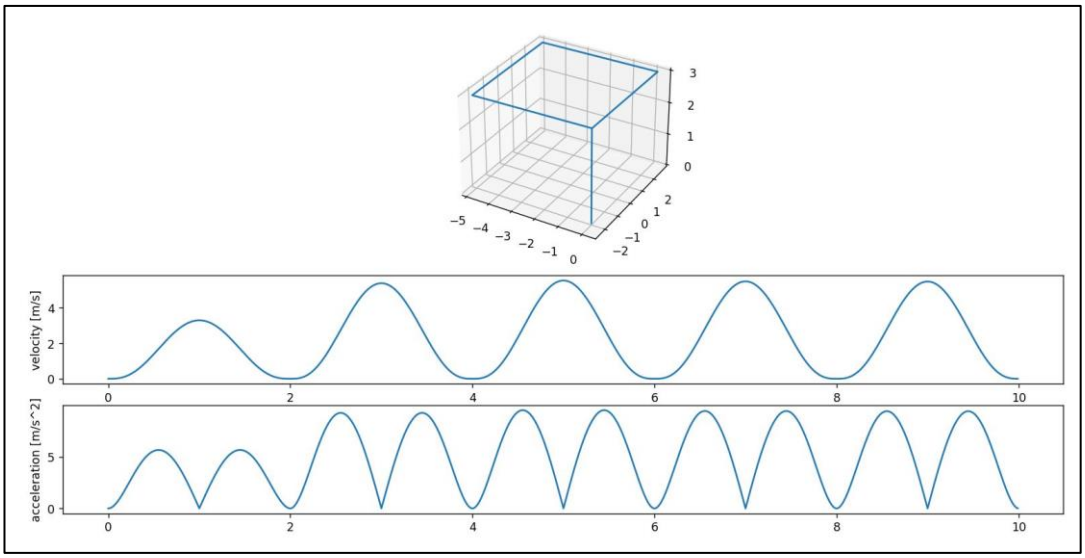
Kübik Bezier eğrisi için 3. dereceden 4 kontrol noktasının oluşturulması.

$$C(t) = (1-t)^3P_0 + 3t(1-t)^2P_1 + 3t^2(1-t)^2P_2 + t^3P_3\,, \tag{3.89}$$

Çizelge 3.5'te başlangıç konumları verilen İHA 1'in Çizelge 3.9'da kare formasyon noktaları için 1. dereceden doğrusal Bezier eğrileri ile yörünge planlaması yapılmıştır.

Çizelge 3.21. Bezier eğrileri ile kare formasyonun kontrol noktaları.

Kontrol Noktası	x	y	z
P0	0	-2	0
P1	0,1291	-2,2740	3
P2	-4,8001	-2,3425	3
P3	-4,8255	2,7255	3
P4	0,1994	2,7413	3



Şekil 3.32. Kare formasyonun Bezier eğrileri ile yörünge planlaması.

3.2.12. Sürü Kontrol

Sanal gerçek entegre çalışmasının, sürü kontrol işleminde, sınırları belirlenmiş arama uzayında sürü uçuş görevinde PSO yöntemi kullanılmıştır. Diğer sürü kontrol algoritmalarının yerel ve küresel minimum performansı simülasyon çalışmalarında karşılaştırılmıştır. Belirlenmiş hedef noktaya, formasyon ile navigasyon görevlerinde kontrol çıktıları ile oluşturulan amaç fonksiyonu kullanılmıştır. Eşitlik 2.17'de gösterilen PSO algoritmasına hedef referansı eklemek için;

Küresel minimum konumu (G_{best}) bulmak için;

$$G_{best} = \min(p_i), \tag{3.90}$$

Hedef referansına göre G_{best} güncellemesi için;

$$G_{best} = \min(P_{target} - p_i), \tag{3.91}$$

Global (küresel) minimum arama ve yerel minimum (P_{best}) güncellemek için, Yerel minimuma sahip çok modlu MICHALEWICZ fonksiyonu Eşitlik 3.92'de kullanılmıştır.

$$f(p)_t = -\sum_{i=1}^{d} \sin(p_i) \times \sin\left(\frac{(i+1)\times p_i^2}{\pi}\right)^{2m}, \tag{3.92}$$

m parametresi, fonksiyon çıktısındaki vadilerin ve sırtların dikliği, d boyutu (x,y) temsil etmektedir.

Eşitlik 3.93'de küresel minimum aramak için EASOM fonksiyonu kullanılmıştır. İki fonksiyonun arama performansı test edilmiştir.

$$f(x)_t = -\cos(x_1)\cos(x_2)\exp\left(-(x_1-\pi)^2-(x_2-\pi)^2\right), \tag{3.93}$$

Her iterasyonda p_{best} değeri Eşitlik 3.94'de gösterildiği şekilde her iterasyonda güncellenir.

$$p_{best} = \begin{cases} p, & f(p)_{t+1} < f(p)_t \\ p_{best}, & f(p)_{t+1} \geq f(p)_t \end{cases}, \tag{3.94}$$

Eşitlik 2.17'de verilen parçacık hızlarının pozisyon güncellemeleri için;

$$x(t+1) = x(t) + v(t+1), \tag{3.95}$$

Tüm sistem durum kontrol işleminde, sürü kontrolü, formasyon kontrolü ve çarpışma engelleme çıktıları tek durumda Eşitlik 3.96'de gösterilmiştir.

$$f(u)_s = V_i^{t+1} + V_c + V_f, \tag{3.96}$$

Eşitlik 3.96'da V_i^{t+1} PSO sürü kontrol çıktısı, V_c çarpışma engelleme, V_f formasyon kontrol çıktısı. Son durumda sürü kontrol çıktısının parçacıklara uygulanmadan önce Eşitlik 3.25'de verilen normalizasyon denklemi ile kuvvet etkileri oranlanır. Bu

işlemin amacı, sürü kontrol çıktısının vektör büyüklüğü, formasyon ve çarpışma engelleme çıktılarından büyük olması durumunda parçacığın çarpışmadan kaçınma veya formasyonu korumak için yeterli itki kuvvetini oluşturamaz.

Algoritma 3 PSO Algorithm

```
1   procedure pso()
2       c1,c2 = 2
3       w = 0.5
4       for j < uavs
5           rand1 = random(-1,1)
6           rand2 = random(-1,1)
7           c_vel = c1*rand1*(pbest – uav_pos[j])
8           s_vel = c2*rand2*(gbest – uav_post[j])
9           vel(t+1) = w*uav_vel[j] + c_vel + s_vel
10          pos(t+1) = pos(t) + vel(t+1)
11          update_pbest()
12      update_gbest()
```

Eşitlik 3.96'da matematiksel modeli verilen PSO algoritması Çizelge 3.22'de verilen başlangıç konumlarına göre uygulanmıştır. Kontrol çıktısı 1. iterasyon için hesaplanan parametre çıktıları Çizelge 3.23 ve Çizelge 3.24'de gösterilmiştir.

Çizelge 3.22. Parçacıkların başlangıç konumları.

Parçacık (İHA)	X	Y
1	2,6783	2,8149
2	0,7715	0,6251
3	3,6488	2,2958
4	1,3153	1,9173
5	0,4767	1,9172

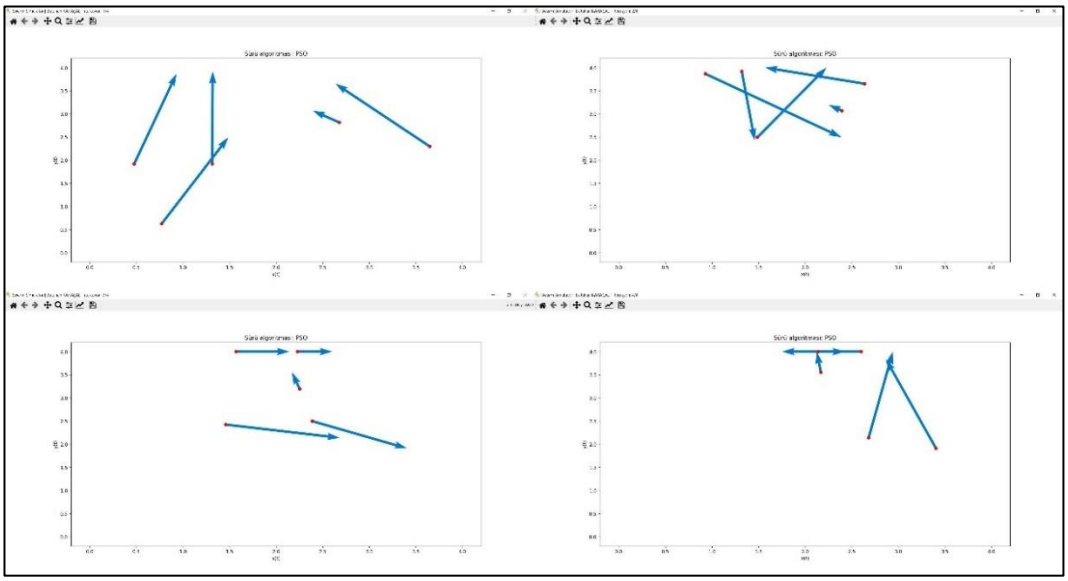
Çizelge 3.22'de ilk sütunda parçacık ismiyle olarak her bir İHA'nın numarası yer almaktadır.

Çizelge 3.23. Parçacıkların 4 iterasyon sonucu Gbest değerleri.

İterasyon	Gbest (x,y)
1	x:2,3949 y:3,0671
2	x:2,2532 y:3,1932
3	x:2,6012 y:4
4	x:2,1412 y:4

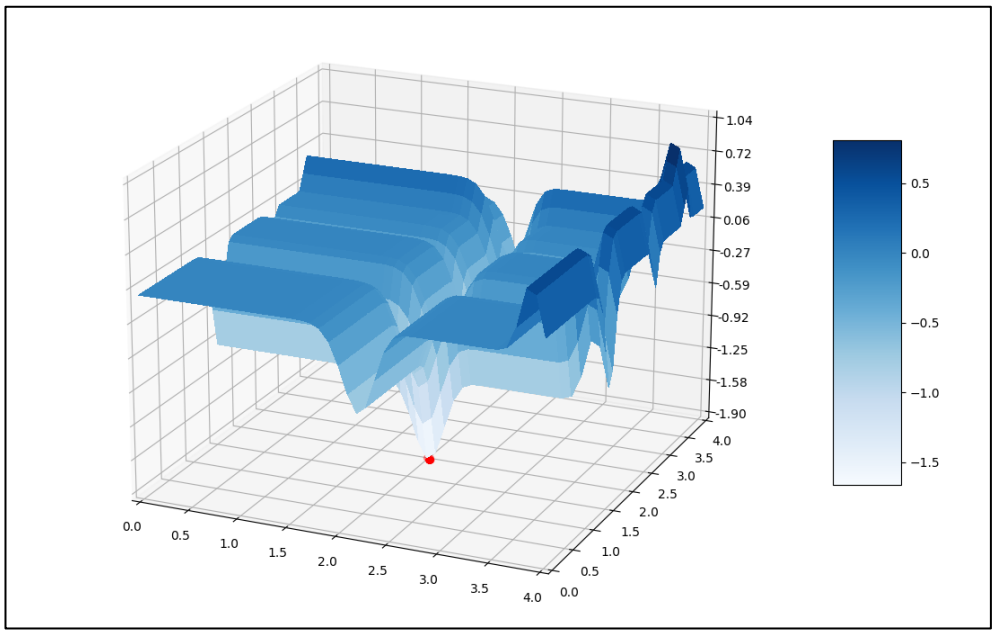
Çizelge 3.24. PSO algoritmasında her parçacığın 1. iterasyon parametre çıktıları.

Parçacık	Pbest (x,y)
1	x: 2,3949 y: 3,0671
2	x: 1,4868 y: 2,4928
3	x: 2,6378 y: 3,6516
4	x: 1,3153 y: 1,9173
5	x: 0,4767 y: 1,9172

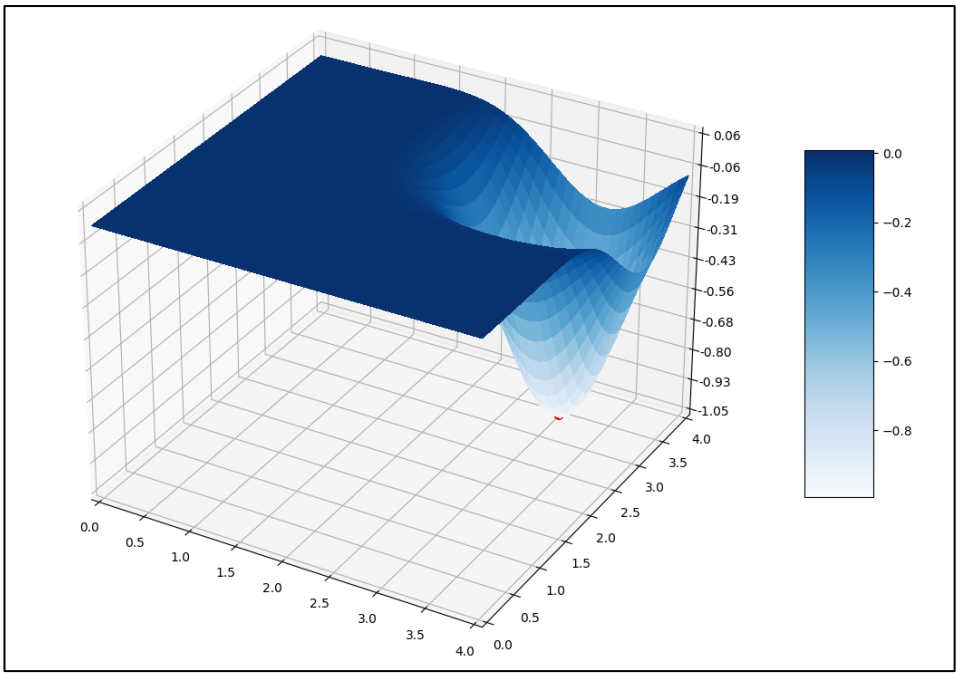


Şekil 3.33. PSO algoritmasının 4 iterasyon ile vektörel simülasyonu.

PSO algoritmasının küresel ve yerel minimum tespiti için kullanılan MICHALEWICZ ve EASOM fonksiyonlarının çıktıları sınır değerleri [0,4] ve 20 iterasyon sonuçları sırası ile Şekil 3.34 ve Şekil 3.35'te gösterilmiştir.

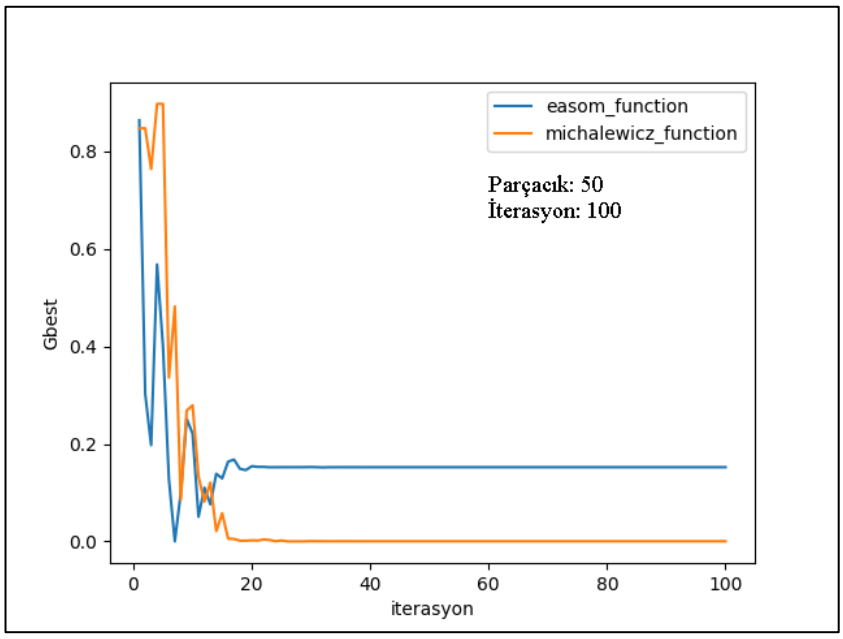


Şekil 3.34. PSO minimumu aramak için MICHALEWICZ fonksiyon çıktısı.

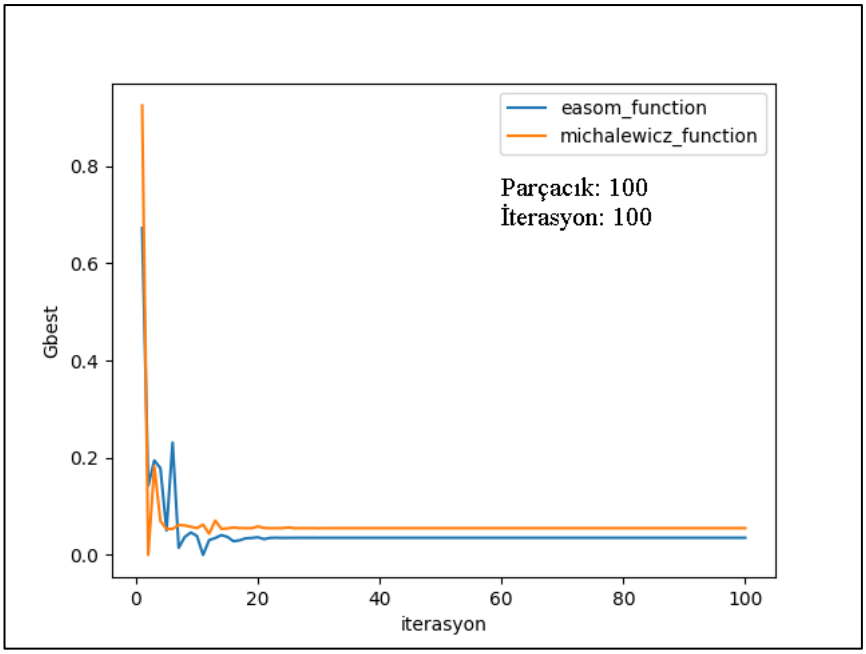


Şekil 3.35. PSO minimumu aramak için EASOM fonksiyon çıktısı.

Şekil 3.34 ve Şekil 3.35'te PSO algoritmasının minimum arama fonksiyonlarının çıktılarında tepe dikliklerinin (minimum) noktalarında kırmızı ile temsil edilen parçacıkların toplandığı görülmektedir.

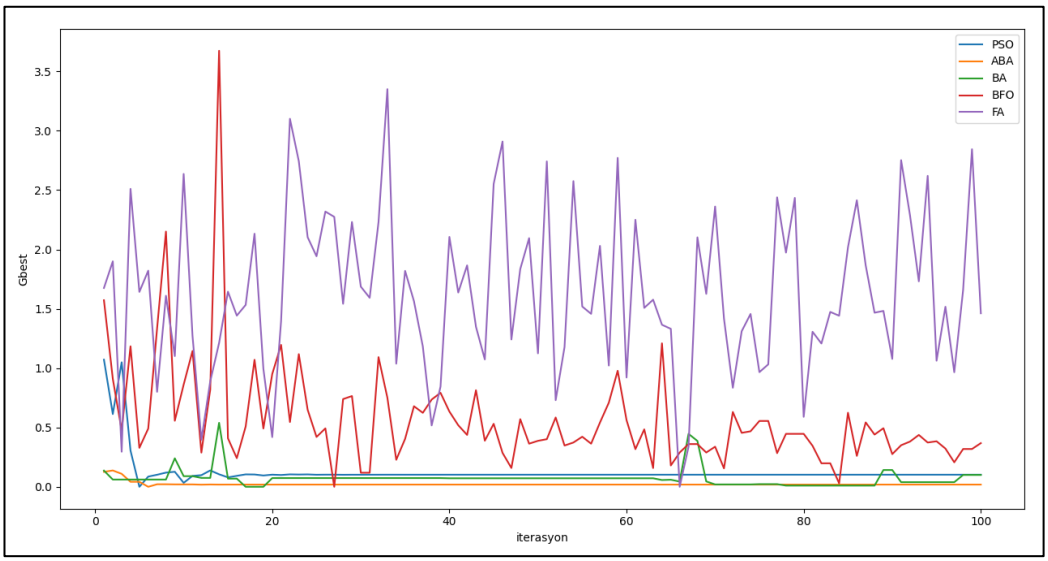


Şekil 3.36. Minimum bulmada iki fonksiyonun performans karşılaştırması.



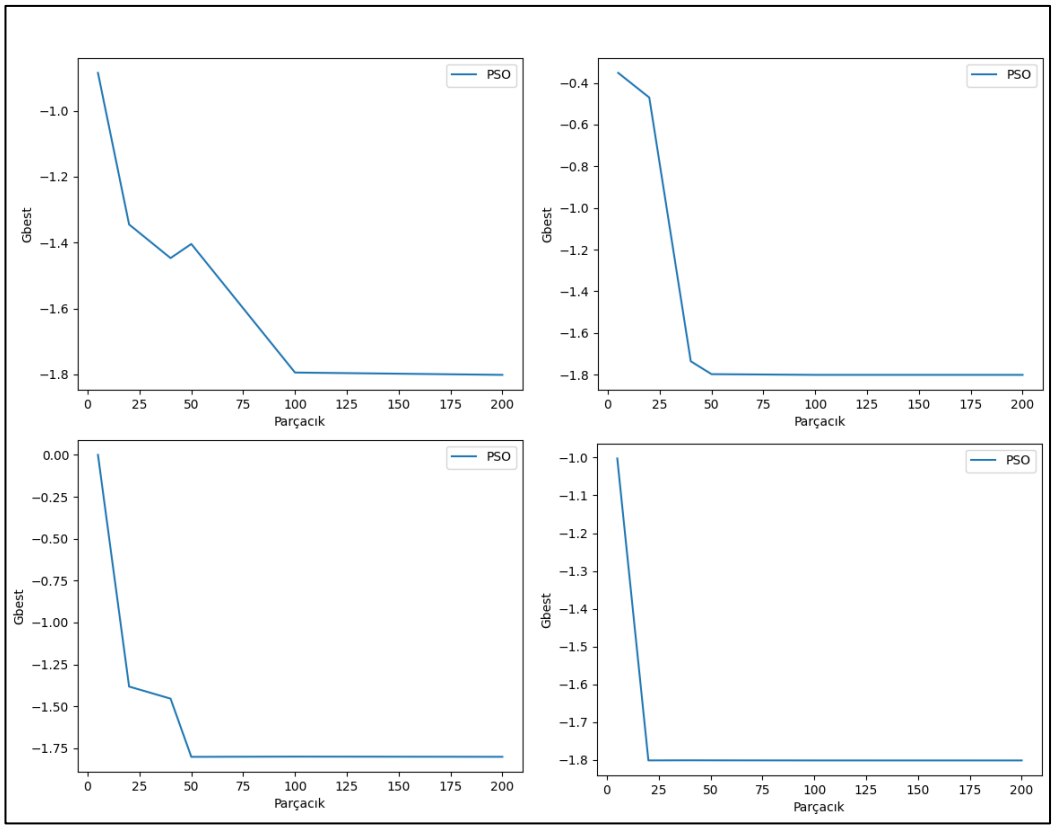
Şekil 3.37. Minimum bulmak için farklı iterasyonve parçacık sayısı testi.

Diğer sürü kontrol algoritmaları ABA, BA, BFO ve FA (Yapay Arı Kolonisi, Yarasa Algoritması, Kuş Sürüsü Algoritması ve Ateş Böceği Algoritması) MICHALEWICZ fonksiyonu ile modellenmiş simülasyon performansları değerlendirilmiştir.



Şekil 3.38. Sürü algoritmaları xy bileşke vektörü test sonuçları.

Karşılaştırma sonuçları iterasyon sayısı sabit tutularak PSO algoritmasının parçacık sayısı değiştirilerek, 100 iterasyon için toplam 200 parçacık başlangıçta oluşturularak konumları sabit, katsayılar her iterasyon ve parçacıkta rastgele oluşturulmuştur. Başlangıçta oluşturulan konumlardan 6 farklı parçacık (5,20,40,50,100,200) sırası ile pozisyonlar alındığı için bir sonraki parçacık sayısı içerisinde, parçacıkların birbirlerine göre en iyi konumu değişmektedir. Minimum aramada MICHALEWICZ fonksiyonu ile sırası ile ABA ve PSO algoritmasının küresel en iyi konuma ulaştığı gözlemlenmiştir.

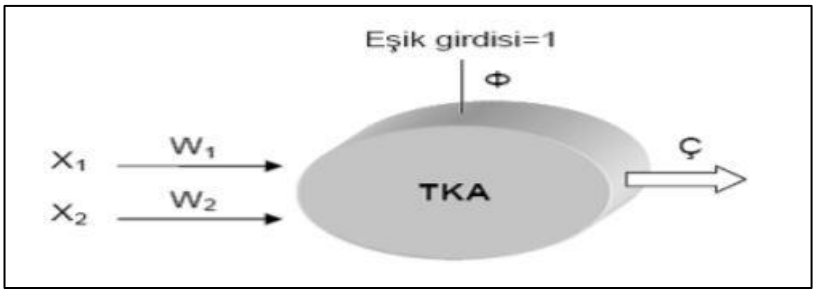


Şekil 3.39. Parçacık sayısının küresel minimum etkisi.

Sonuçlar değerlendirildiğinde arama uzayında sabit hedef konuma yaklaşım parçacık sayısı arttıkça azalmaktadır. Sürü uçuş işleminde PSO algoritması küresel ve yerel minimum değerleri parçacık konumu sürekli değiştiği için yeni minimum noktaları üretmektedir. Bu sebeple karşılaştırma da parçacık sayısı arttıkça önceki pozisyonlar sabit olarak atandığında amaca daha hızlı ulaşmaktadır.

3.2.13. Tek Katmanlı Yapay Sinir Ağı

Doğrusal problemlerin çözümünde kullanılan, tek katmanlı yapay sinir ağı (YSA) sadece girdi ve çıktı katmanından oluşmaktadır. Kullanılan aktivasyon fonksiyonuna göre çıkış değeri [-1,1] aralığında ve eşik değeri ϕ ile ifade edilmiştir ve çıktı değerinin 0 olmasını önler [119].



Şekil 3.40. Tek katmanlı yapay sinir ağı modeli [119].

$$Ç = f(\sum_{i=1}^{N} w_i x_i + \phi), \tag{3.97}$$

$$f(g) = \begin{cases} 1, & Ç > 0 \\ -1, & Ç \leq 0' \end{cases} \tag{3.98}$$

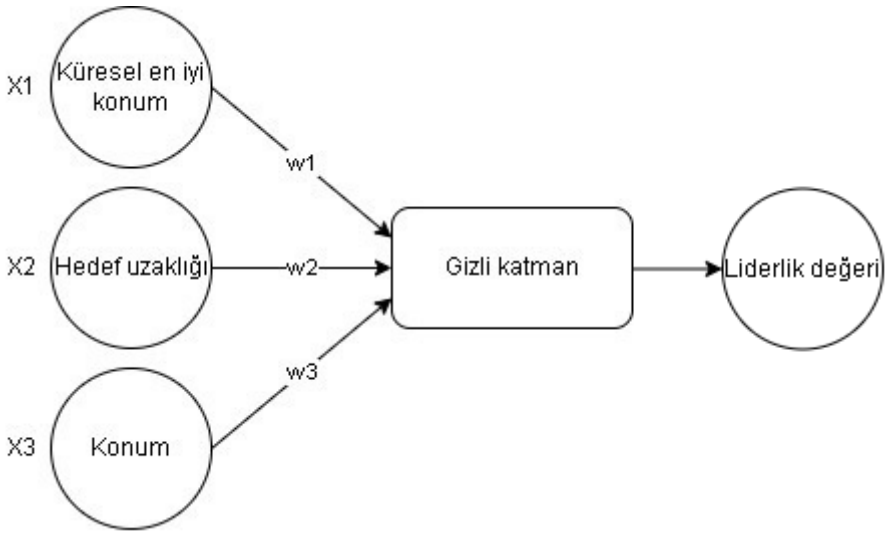
Ağırlık güncellemesi her iterasyonda gerçekleştirilir:

$$w_i(t+1) = w_i(t) + \Delta w_i(t), \tag{3.99}$$

3.2.14. Sürü Liderinin Belirlenmesi

Sürü liderini belirlenmesinde temel "if-then" yapısı yerine tek katmanlı yapay sinir ağı modeli kullanılmıştır. Yapay sinir ağının giriş parametreleri; küresel en iyi konum, hedef var ise uzaklığı ve konum değeri ağırlıkları ile uygulandığında 1'e yakınsayan en iyi sonuç lider olarak seçilmektedir.

Bu tez çalışması kapsamında sürü liderinin belirlenmesi, İHA sayısının az olması ve ağ eğitiminde kullanılan giriş parametrelerinin sınırlı alan içerisinde olduğundan doğrusal bir problemdir. Giriş parametrelerinin fazlam olması durumunda giriş ve çıktı arasına gizli katman ve geri yayılım işlemi ile ağ çok katmanlı sinir ağına evrilir. Bu nedenle simülasyon testlerinde, sürü kontrol çıktısının küresel iyi konumuna göre belirlenen eğitim veri seti oluşturulmuştur. Yapay sinir ağında eğitim için kullanılan veri setinin çeşitliliği ve sayısı tahmin yeteneğini doğrudan etkilemektedir.



Şekil 3.41. Sürü lideri seçim işlemi ağ yapısı.

Şekil 3.41'de gösterilen ağ yapısının gizli katmanında, kullanılacak aktivasyon fonksiyonunun seçim işleminde, sınıflandırma yapılmaması, ara değerlerin kullanılmadığı çıktığının [-1,1] aralığında çalışan tanh (tanjant hiperbolik) fonksiyonu kullanılmıştır.

İleri yayılım işleminde giriş ve ağırlık çarpımları aktivasyon fonksiyonuna gönderilir.

$$f(x) = \frac{2}{1+e^{-x}} - 1, \tag{3.100}$$

Geri yayılım işleminde hata değerini azaltmak ve pozisyonunu değiştirmek için tanh fonksiyonunun türevi ve hata çarpımı yeni ağırlık değerini oluşturur. Her iterasyonda ağırlıklar Eşitlik 3.103'de gösterildiği şekilde güncellenir.

$$error = y - f(x), \tag{3.101}$$

$$f(\dot{x}) = 1 - f(x)^2, \tag{3.102}$$

$$w_{i+1} = w_i + y * error * f(x), \tag{3.103}$$

Çizelge 3.25. Lider seçiminde kullanılan 1 iterasyon örnek eğitim veri seti.

Parçacık	Konum (x y)		Gbest (x y)		Pbest (x y)		Liderlik değeri
1	2,6783	2,8145	2,6012	4	2,3949	3,0671	0
2	0,7715	0,6251	2,6012	4	1,4868	2,4928	0
3	3,6488	2,2958	2,6012	4	2,6378	3,6516	1
4	1,3153	1,9173	2,6012	4	1,3153	1,9173	0
5	0,4767	1,9172	2,6012	4	0,4767	1,9172	0

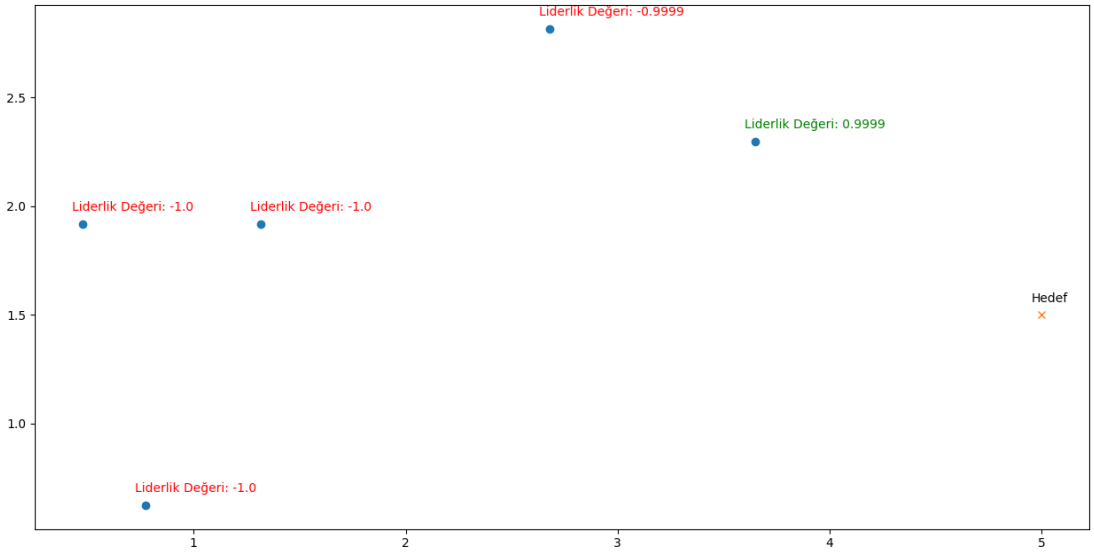
Çizelge 3.25'te x ve y konum bilgileri arasına boşluk eklenerek gösterilmiştir. Oluşturulan veri seti konum bilgileri işleme alınırken konum bilgilerinin Eşitlik 3.104'teki bileşkesi gönderilir.

$$p = \sqrt{x^2 + y^2 + z^2}, \tag{3.104}$$

Çizelge 3.26. Öğrenilmiş veri seti 10 000 iterasyon lider tahmini.

Parçacık	Liderlik değeri
1	-0,9999
2	-1
3	0,9999
4	-1
5	-1

Sürü liderinin seçim işlemi sadece merkezi kontrol yazılımından talep geldiğinde, lider ile bağlantının zaman aşımına uğradığında veya lider sürüden ayrıldığında gerçekleşmektedir. Simülasyon testlerinde lider değişim işlemi sürünün mevcut konumlarında beklemesi ile gerçekleştirilmiştir. Seçim sonrası lider etiketi haberleşme formatında güncellenerek işlem sonlandırılmıştır. Sürü lideri değişiminde, mevcut İHA sayısına göre en uygun formasyon ataması gerçekleştirilmektedir. Örnek olarak beşgen formasyonunda 5 İHA'nın lider bağlantısı kopması ile yeni lider seçimi sonrası formasyon oluşumu kare olarak belirlenmektedir.



Şekil 3.42. Lider seçimi öğrenilmiş örnek veri seti sonucu.

3.2.15. Mavlink Haberleşme Protokolü

İHA'ların otonom kontrolü için uçuş kontrolcü ile haberleşme protokolü olarak MAVLINK haberleşme protokolü kullanılmaktadır. Çift yönlü çalışan protokol ile İHA'ların pozisyon ve diğer tüm bilgilerini okunabilmekte ve komut gönderilebilmektedir. Sürü kontrolü ve diğer kontrol algoritmalarının bu temel bilgileri ile oluşturulmakta ve kontrol çıktıları hız veya pozisyon bilgisi olarak gönderilmektedir.

COMP ID : Araç kimliği, sistemdeki diğer bileşenleri ayırt etmek için

MSG ID : Mesaj türüne göre çözmek için her mesaj için ayrı kimlik bulunur

PAYLOAD : Mesaj verileri (mesaj türüne göre

BÖLÜM 4

DENEYSEL SONUÇLAR

Bu tez çalışması kapsamında oluşturulan matematiksel denklemler MATLAB ortamında test edilerek platformdan bağımsız bir alt yapı oluşturmak için Python ve QT5 kütüphanesi ile sürü kontrol arayüzü geliştirilmiştir. Çalışmanın sanal-gerçek bölümü GAZEBO-ROS alt yapısında deneysel çalışmalarda kullanılan telemetri bilgileri kullanılarak modellenmiştir. Deneysel uçuş testleri Karabük Üniversitesi ay-yıldız stadyumunda 41,204941 enleminde, 32,656360 boylamında 17 832 m² yeşil saha alanı içerisinde gerçekleştirilmiştir. Olası GPS yenileme hızı, konum hassasiyeti (RTK veya yerel konumlanama sistemleri kullanılmamasından kaynaklı oluşan hassasiyet) ve rüzgar etkisi sebebiyle çarpışma engelleme algoritmalarının tepki süresinde oluşabilecek gecikmeden dolayı İHA'lar arası güvenli mesafe 6 m olarak belirlenmiştir.

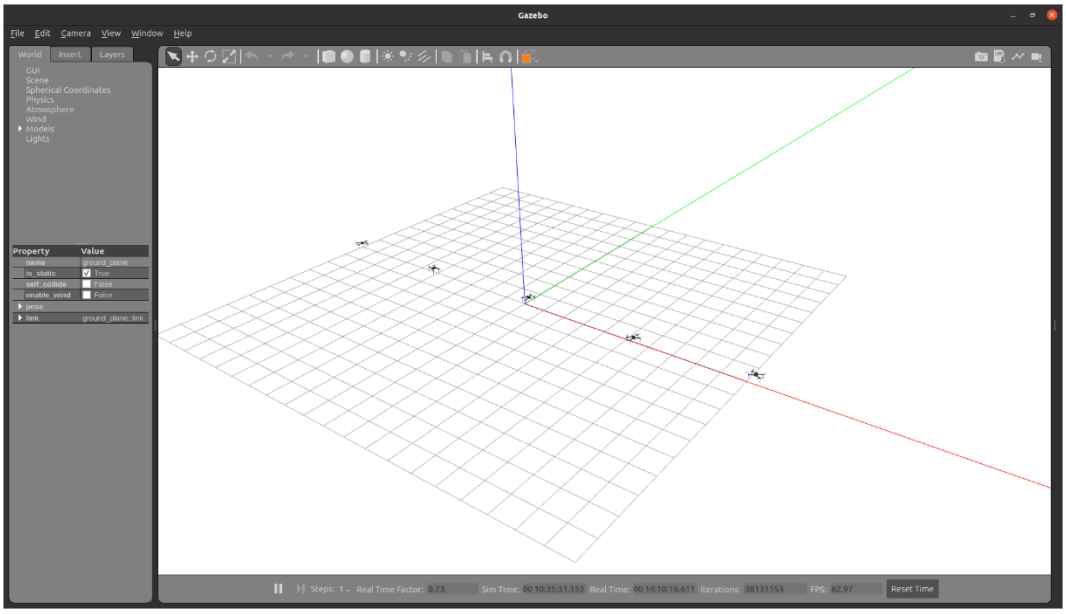
Şekil 4.1. Deneysel çalışmaların gerçekleştirildiği alan.

Şekil 4.2. Sanal-gerçek entegre deney alanı.

4.1. SANAL GERÇEK ENTEGE SİMÜLASYON ÇALIŞMASI

Çalışmasının 3 boyutlu simülasyon çalışmalarında kullanılan İHA'ların fizik motoru ile uyumlu çalışan ve gerçek verilerin sağlanması amacıyla GAZEBO ve ROS alt yapısı kullanılmıştır. Sağlanan MAVLINK mesajları ve GPS küresel konum verileri deneysel olarak gerçek ortamda kullanılacak İHA'lar ile birebir örtüşmektedir. Bu şekilde tasarlanan ve testleri yapılan tüm algoritmalar gerçek ortamda yapılan testlere direkt olarak uygulanabilmektedir. Simülasyon ortamında hava şartları ve iletişim kısıtlaması (olası gecikme veya bağlantı hataları) olmadan gerçekleştirilmiştir.

Sanal gerçek entegre simülasyon çalışmasında, deneysel çalışmalarda kullanılan İHA'nın telemetri bilgileri, oluşturulan Wi-Fi ağında her İHA için atanan TCP ve UDP portlarında paylaşılmaktadır. Bu bağlantı, GAZEBO-ROS modeline bağlanarak gerçek zamanlı kontrol işlemi gerçekleştirilmiştir.



Şekil 4.3. GAZEBO ortamında oluşturulan 5 drone modeli.

4.2. MERKEZİ KONTROL YAZILIMI

Sürü kontrol merkezi kontrol ve arayüz yazılımı, İHA'ların belirlenmiş görevlerinin atanması, bireysel kontrol işlemleri, formasyon belirlenmesi vb. görevlerin sürü lideri tarafından veya direkt İHA'lara gönderilerek işlenmesini sağlamaktadır. Merkezi kontrol ve izleme için iki farklı programlama dilinde (Python ve Go) geliştirilmiştir.

Geliştirilen sürü kontrol yazılımının öne çıkan özellikleri:

- Sürü formasyonlarının belirlenmesi
- Sürü uçuş algoritmasının seçilmesi (PSO, YAK, KSO vb.)
- Her İHA'nın bireysel bağlantı, irtifa, iniş işlemlerinin gerçekleştirilmesi
- Sürüyü ayırma ve birleştirme
- Formasyon işleminde her İHA'nın diğer İHA ile olan mesafesi
- Sürüler arası mesafe
- Görev atama işlemi (oluşturulan ayar dosyasında daha önceden belirlenmiş görevler atanabilmektedir.) örneğin, irtifa alması, formasyon türünün belirlenmesi, sürü uçuşu, navigasyon (gezinme) işlemleri.

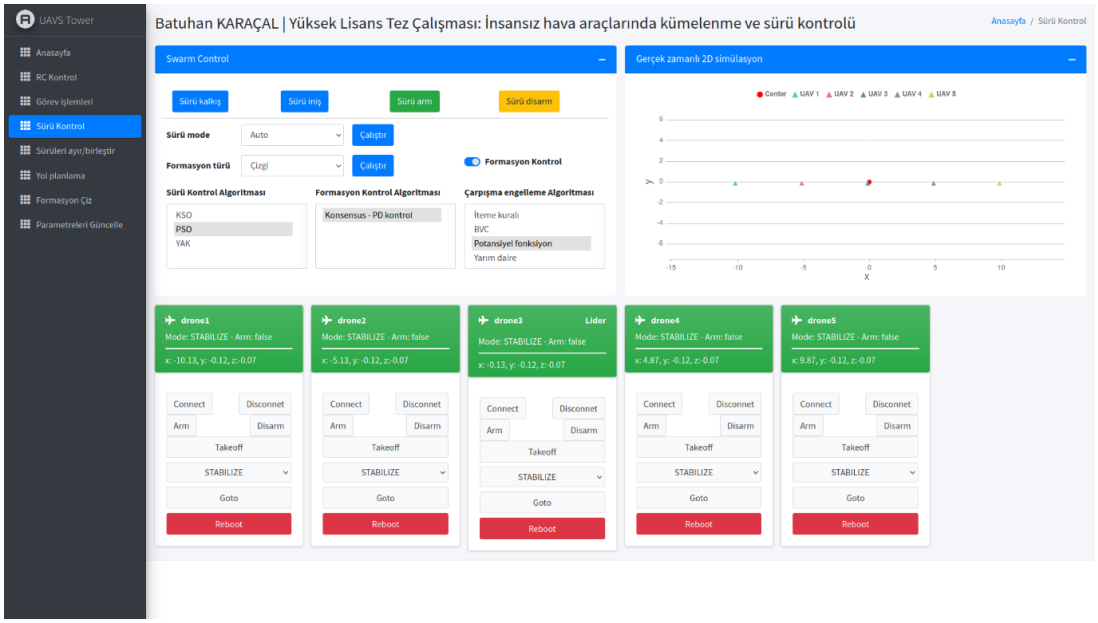
Kontrol merkezi, yarı dağıtık sürü kontrol işlemlerinde, liderin görevleri okuması için belirlenen lidere direkt veya Wi-Fi ağına görev listesi aktarılarak gerçekleştirilmektedir. Merkezi kontrol işleminde, İHA'lar hız ve konum kontrolleri merkezi kontrol bilgisayarı tarafından gerçekleştirilmektedir.

Tüm sürü kontrol işlem süreçlerinde, merkezi sürü kontrol yazılımından gelen mesajlar dış ortam çarpışma engelleme haricinde override (geçersiz kılma) işlemine tabi tutularak kesme işlemi gerçekleştirilir.

Sürülerin tamamen yer istasyonundan kontrol işleminin dezavantajı; her parçacık (İHA) için eş zamanlı çalışma işlemi (multi threading) yapıldığından kullanılan sunucu özelliklerine göre ve sürü üyelerinin sayısına göre performans kaybı ve dar boğaz oluşmaktadır.

Şekil 4.4. Tasarlanan sürü kontrol merkezi yazılımı.

Eş zamanlılık testleri için Şekil 4.4'de geliştirilen sürü kontrol arayüz yazılımı, eş zamanlılık testleri için Şekil 4.5'de Go programlama dili ile geliştirilmiştir.

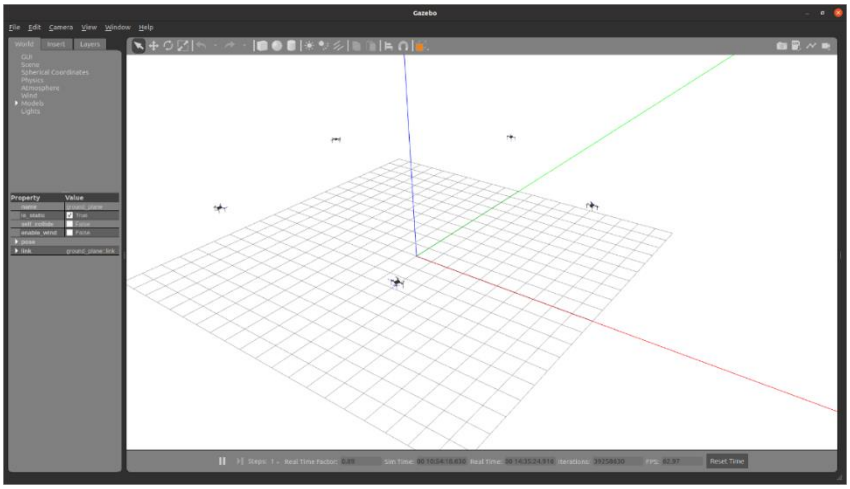


Şekil 4.5. Go progralama dili ile geliştirilen arayüz.

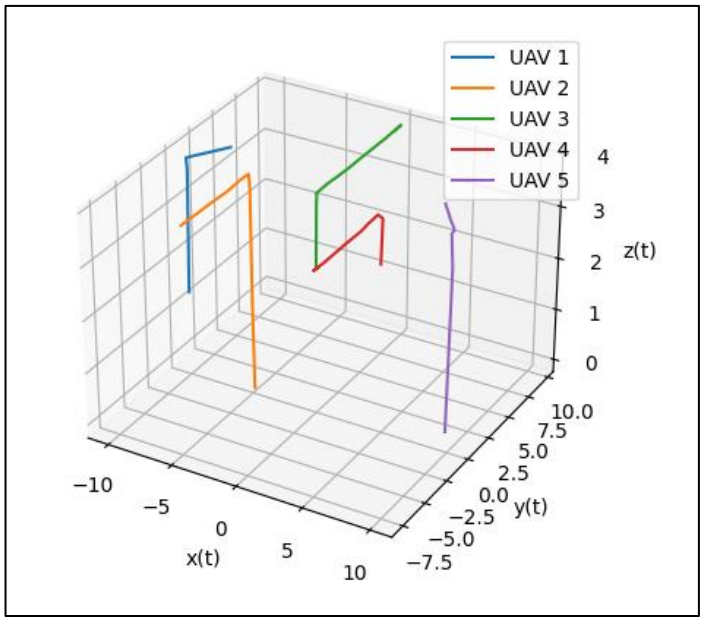
Şekil 4.5'te 5 İHA'nın gerçek zamanlı telemetri verilerine dayanan, derlenebilen ve statik programlama dili olan Go ile geliştirilen sürü kontrol arayüzü gösterilmiştir.

4.3. SANAL GERÇEK ENTEGRE SONUÇLARI

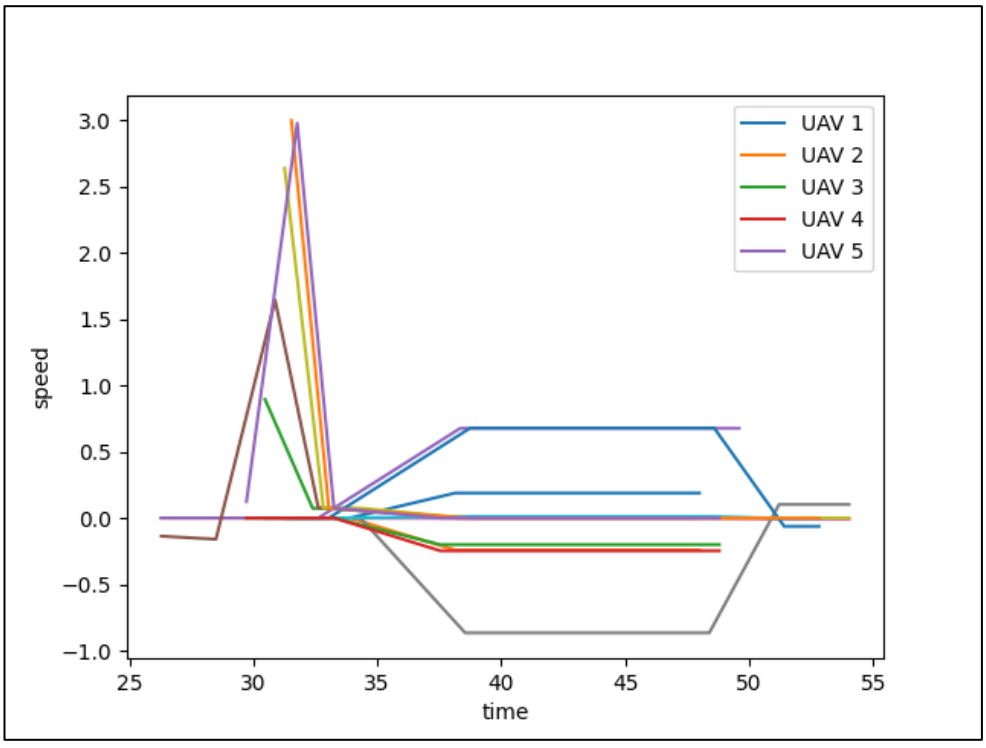
GAZEBO ortamında oluşturulan modellerin, ROS alt yapısı ile MAVLINK mesajları oluşturularak kontrol algoritmaları çalıştırılmıştır.



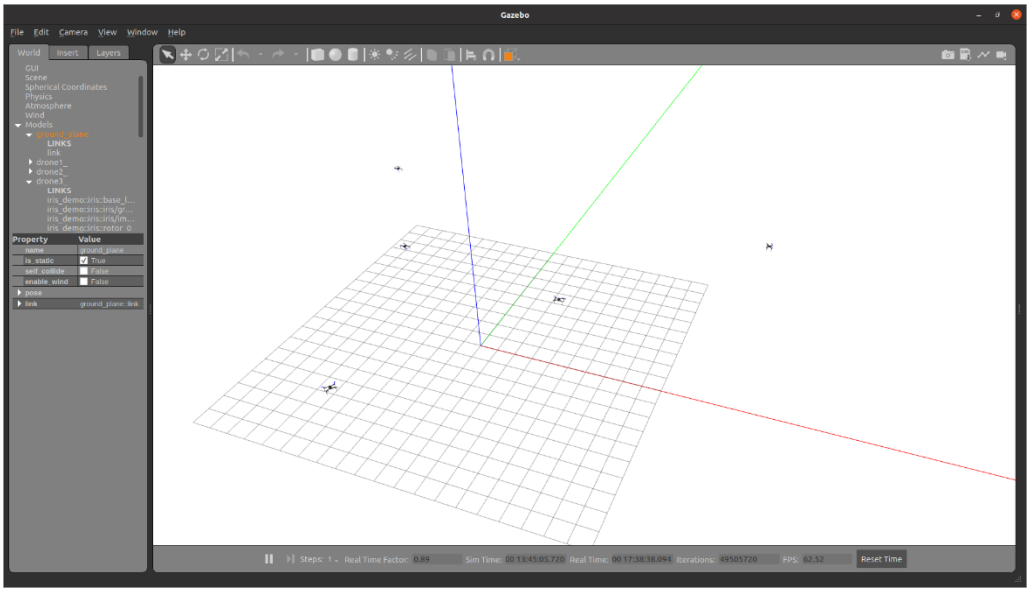
Şekil 4.6. GAZEBO sürü İHA'ların beşgen formasyonu



Şekil 4.7. Beşgen formasyonun xyz kartezyen grafiği.

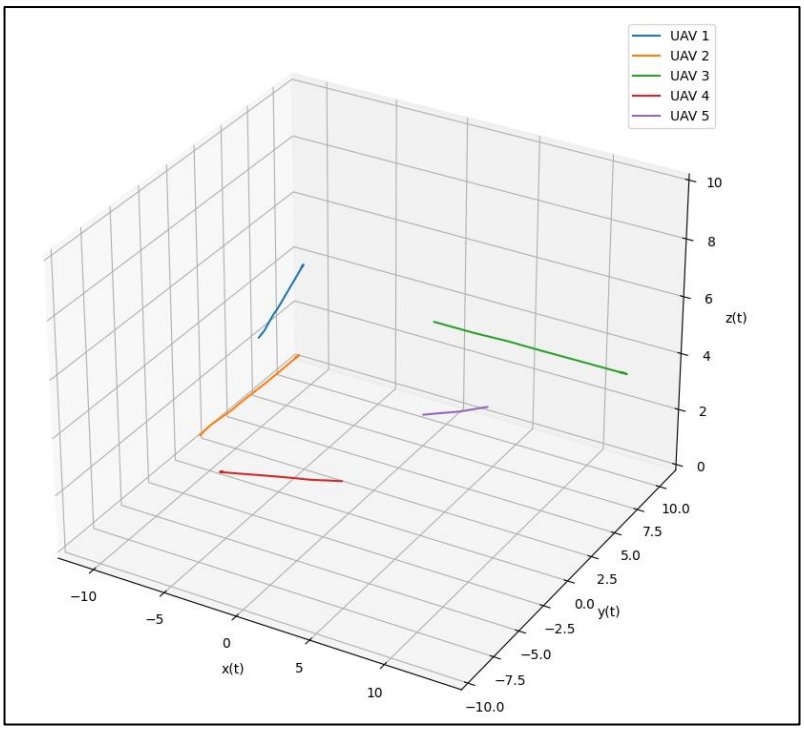


Şekil 4.8. Beşgen formasyon noktalarına hareketin hız-zaman sonucu.



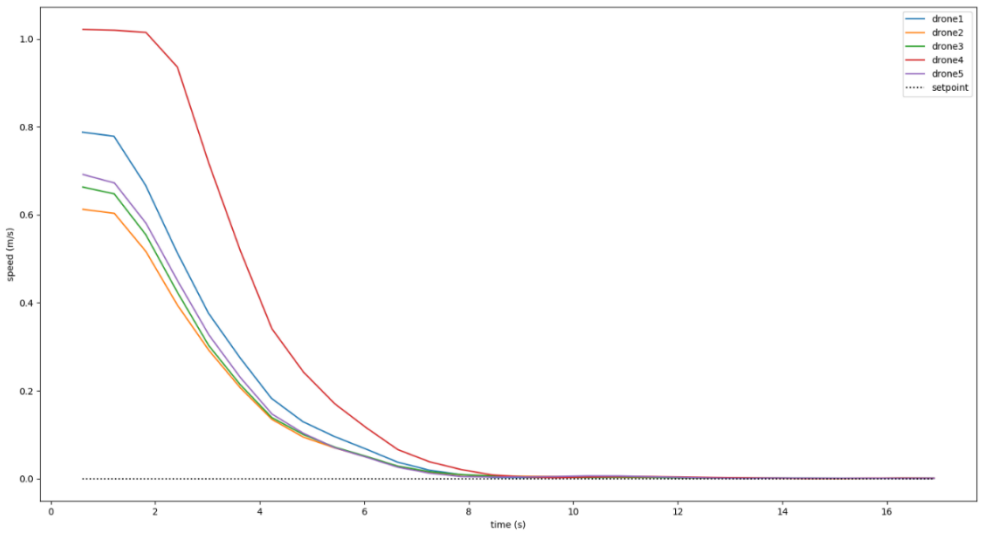
Şekil 4.9. V formasyonu perspektif görüntüsü.

Beşgen geometrik formasyonundan v geometrik formasyonuna geçiş işlemi sürekli zamanlı gerçekleştirilerek uçuş kayıt bilgileri beşgen formasyonundan v geçişi göstermektedir.



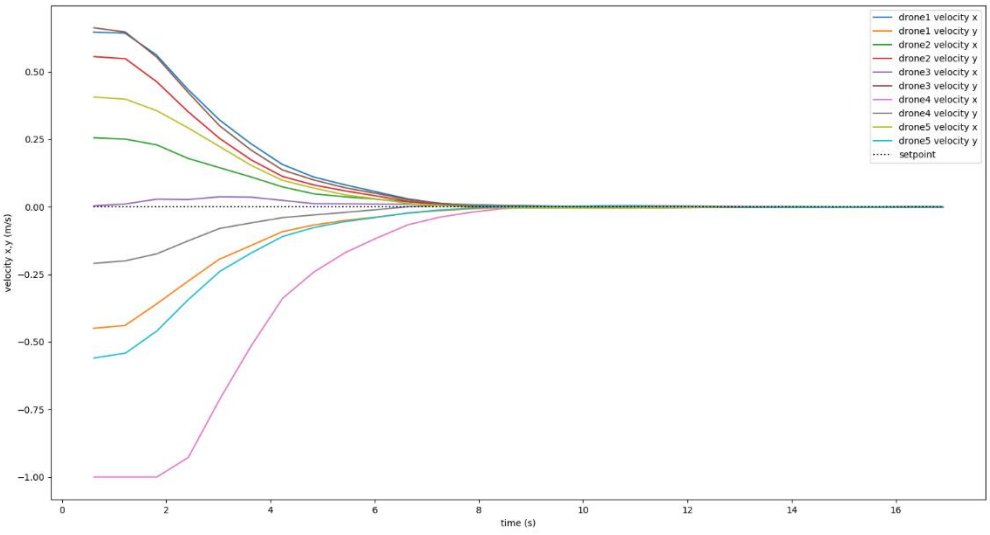
Şekil 4.10. Formasyon geçişinde izlenen xyz konumları.

Şekil 4.10'da beşgen formasyonunda olan İHA'ların v formasyonuna geçiş işleminde Eşitlik 3.65'te verilen formasyon kontrol matrisi ile salınım olmadan doğrusal geçiş işlemi gerçekleştirilmiştir.



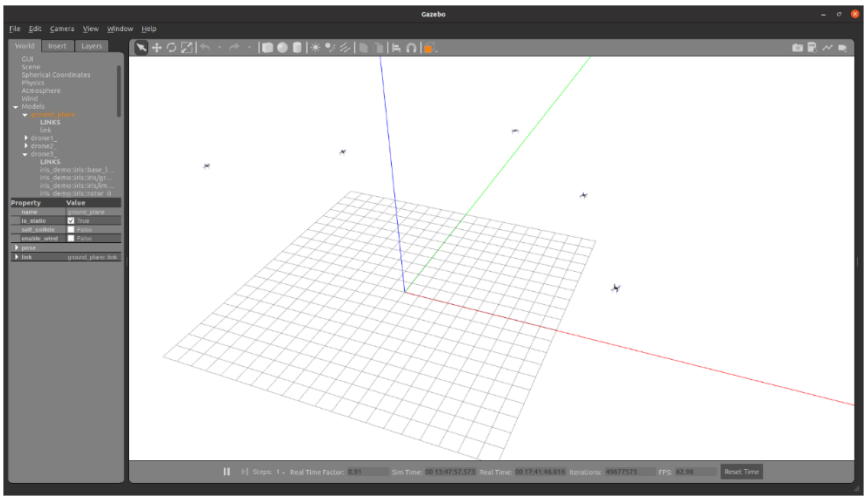
Şekil 4.11. Beşgen – V formasyon geçişi, PD kontrolcü bileşke hızların grafiği.

Şekil 4.11'de formasyon geçiş işleminde, sadece PD tipi formasyon kontrolcüsünün x, y ve z hız vektörlerinin Eşitlik 3.5'te verilen Öklid uzaklığı ile hız vektörünün bileşkesi gösterilmiştir.

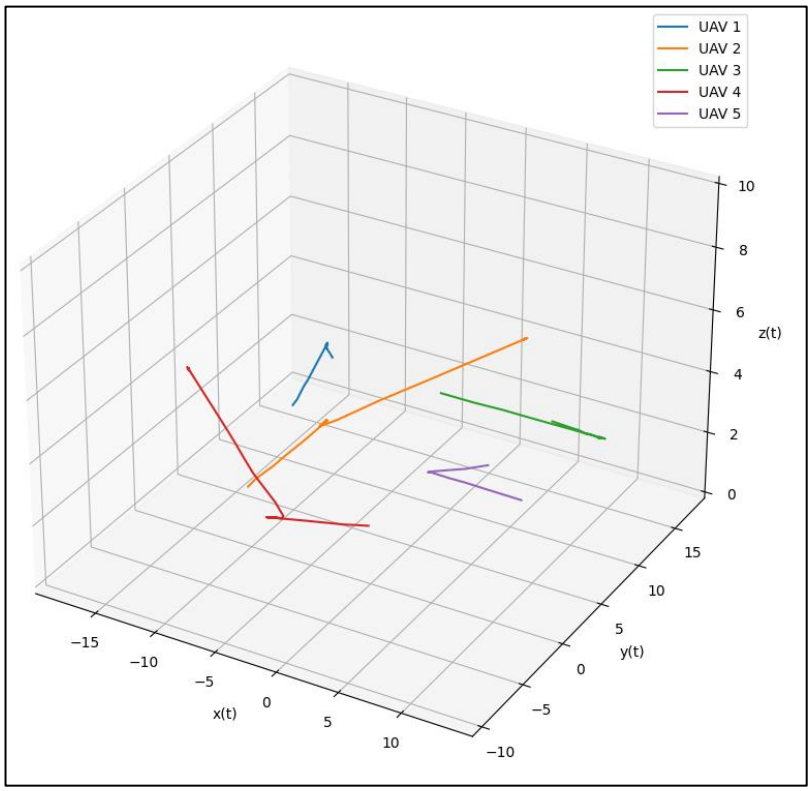


Şekil 4.12. Beşgen – V formasyon geçişi, PD kontrolcü vektörel hızlar.

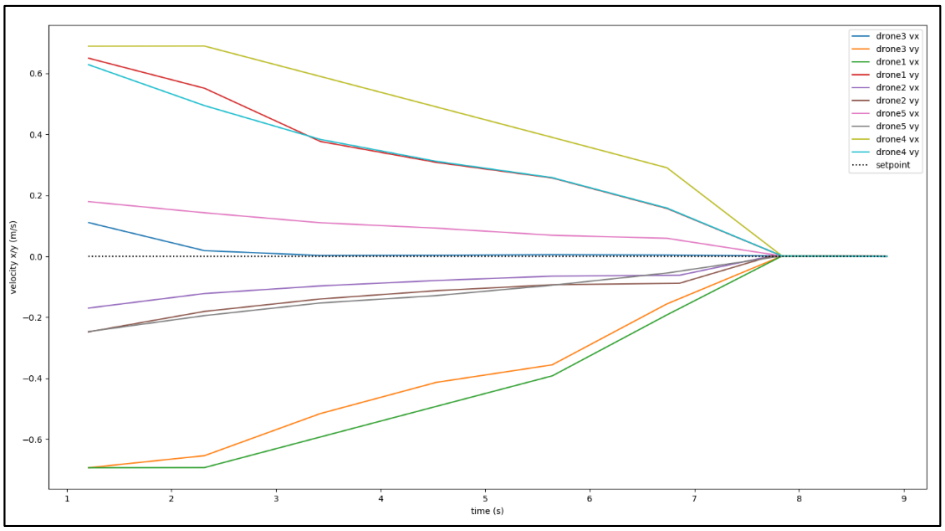
Beşgen formasyonunu başlangıç noktalarından sırası ile v ve hilal formasyon noktalarının oluşturulması ve bu noktalara ilerlemesi sürekli zamanda gerçekleştirilmiştir.



Şekil 4.13. Hilal formasyonu perspektif görüntüsü.

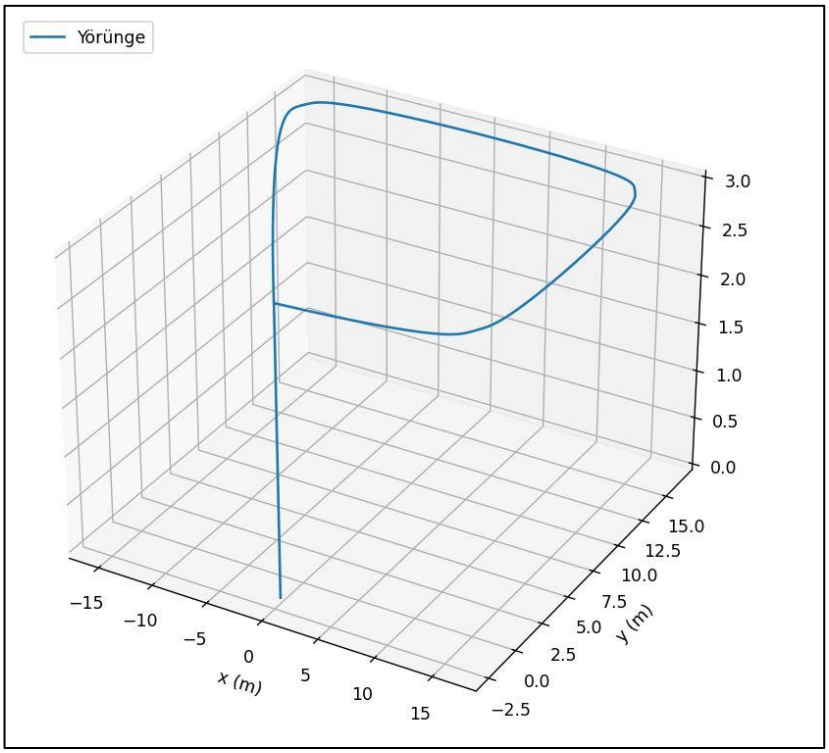


Şekil 4.14. Formasyon değiştirme sırasında xyz konumları.

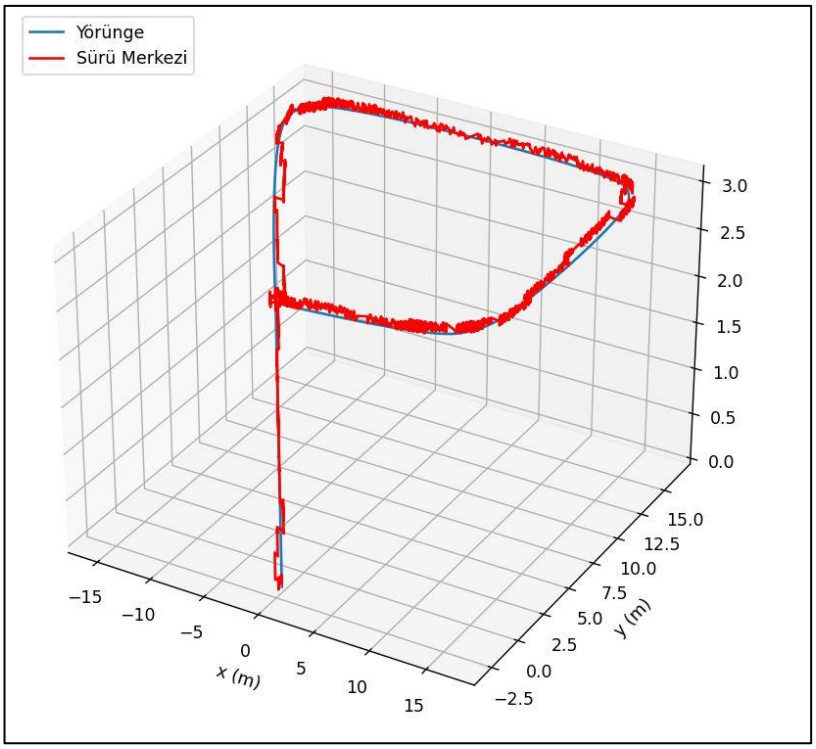


Şekil 4.15. Beşgen-çizgi formasyon geçiş işleminde PID çıktısı.

Şekil 4.15'de formasyon geçiş işleminde, sürü İHA'ların hız eşitlemesi olmadan PID kontrol ile formasyon geçişini tamamlamaktadır.



Şekil 4.16. Navigasyon işlemi için Bezier eğirisi ile yörünge oluşturma.

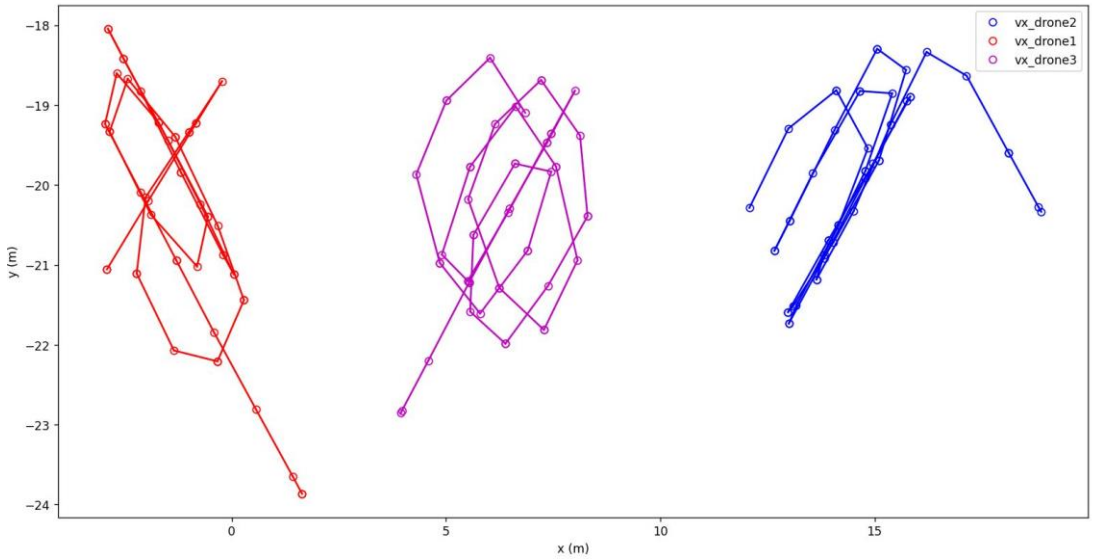


Şekil 4.17. Navigasyon işleminde sürü merkezinin yörünge takibi.

Eşitlik 3.89'da verilen Bezier eşitliği ile navigasyon yörüngesi Şekil 4.16'da oluşturulmuştur. Şekil 4.17'de oluşturulan yörüngenin (mavi) sürü İHA'ların merkezinin (kırmızı) takip ederek sürünün oluşturulan yörüngeyi takip etmesi sağlanmıştır.

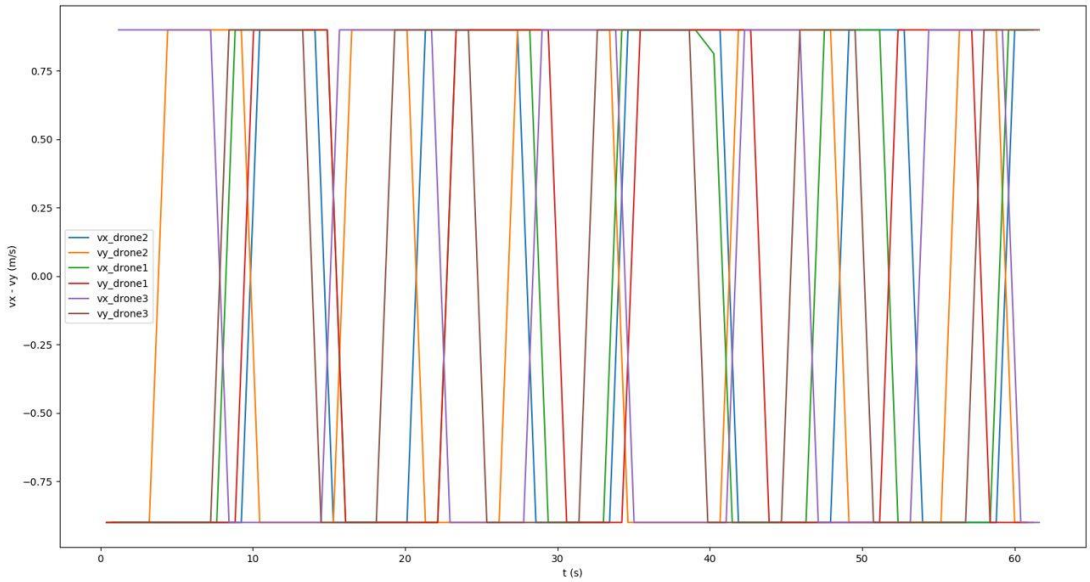
Çizelge 4.1. İkinci dereceden 5 noktalı yörünge değerleri.

	q0			q1	
i	x	y	z	x	y
1	0	-2	0	0	-2
2	0,1291	-2,2740	3	0,1291	-2,2740
3	15	0	3	15,1	0,1
4	15	15	3	14,9	15,1
5	-15	15	3	-15,1	14,9

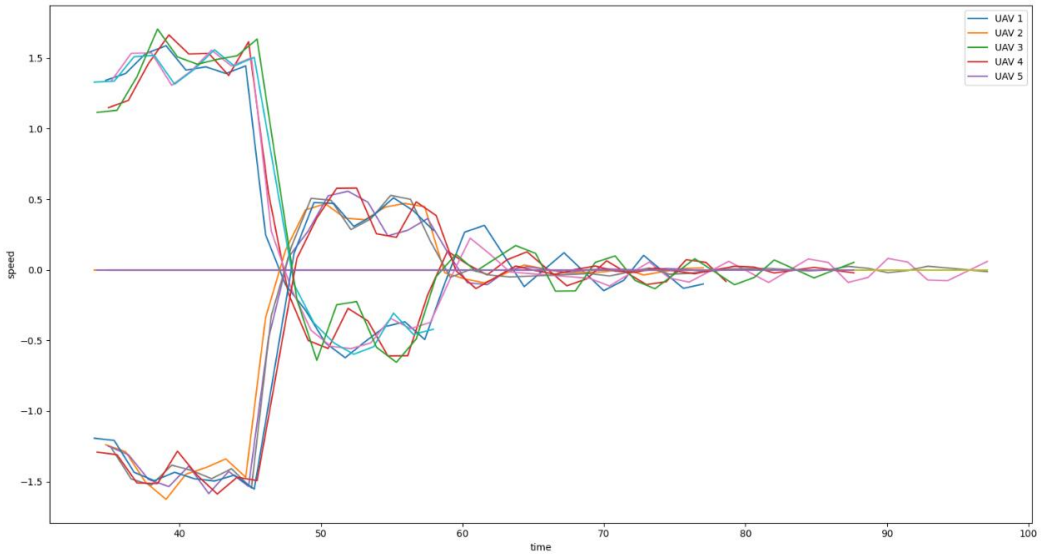


Şekil 4.18. Sürü uçuş işleminde xy konumları.

Şekil 4.18'de Eşitlik 3.91'de verilen PSO algoritmasına hedef ekleme ve çarpışma engelleme algoritması ile oluşan 3 İHA'nın x ve y konumları gösterilmiştir. İHA'ların istenilen bölgede çarpışmadan kaçınarak sınırlandırılmış alanda küme oluşturduğu gözlemlenmiştir. Şekil 4.19'da 3 İHA'nın PSO algoritmasının alan uçuşu işleminde x ve y eksenindeki hızlarının ±0,9 m/s olarak sınırlandırılmış grafiği.

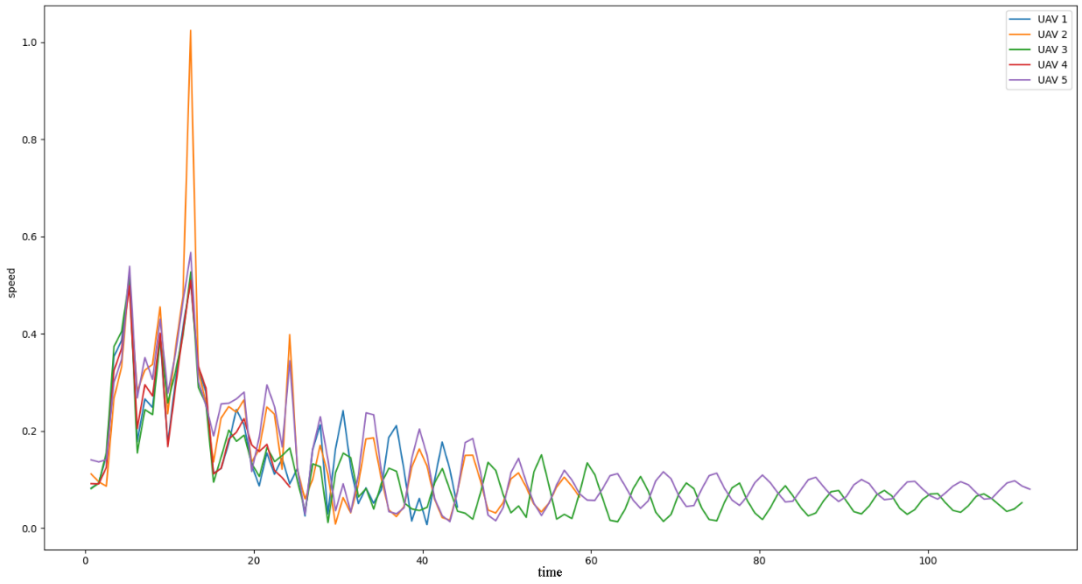


Şekil 4.19. Sürü uçuşu sırasında x ve y eksen hızlarının ayrı gösterimi.



Şekil 4.20. Sürü uçuşu hız zaman grafiği.

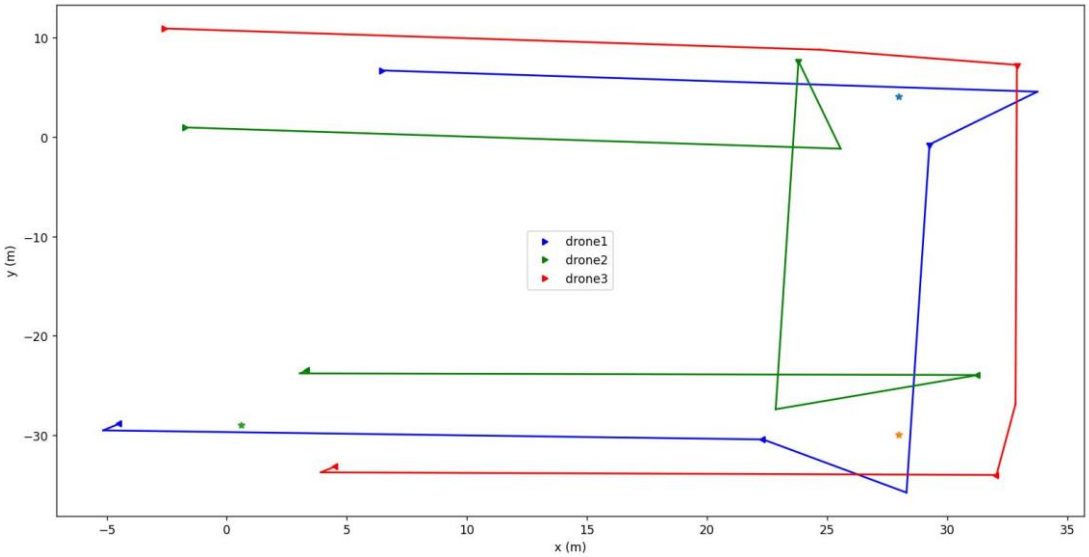
Şekil 4.20'de 5 İHA'nın sürü uçuş işleminde istenilen sınıra yakınsaması ile oluşan x ve y eksenlerindeki vektörel hızların zamanla değişimi gösterilmiştir.



Şekil 4.21. Serbest alan uçuşu x ve y eksen hızları.

Şekil 4.21'de 5 İHA'nın alan uçuşu sırasında, çarpışma engelleme algoritmasının eksen hızına etkisi (turuncu) gösterilmiştir.

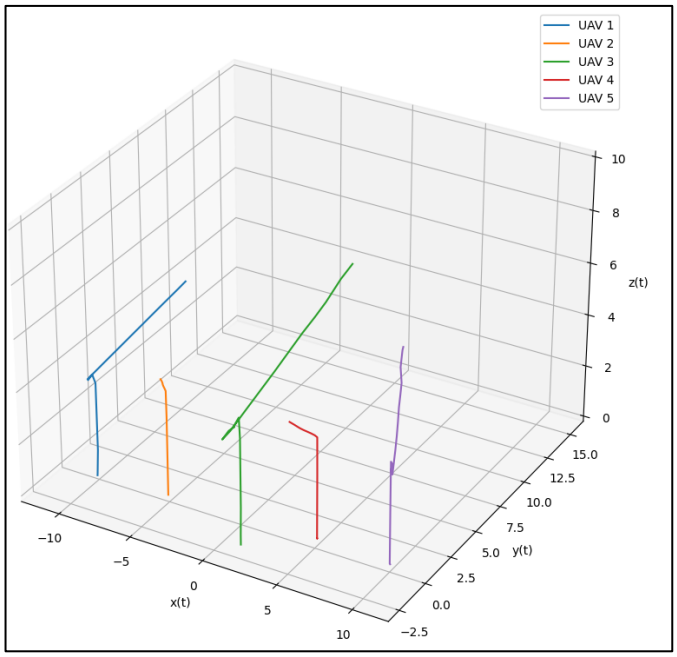
Sürü uçuş işlemi sırasında, her İHA'nın formasyonu korumak ve çarpışmadan kaçınmak için birbirlerine göre hızlarını ayarladığı ve yörüngelerini takip ettiği gözlemlenmiştir.



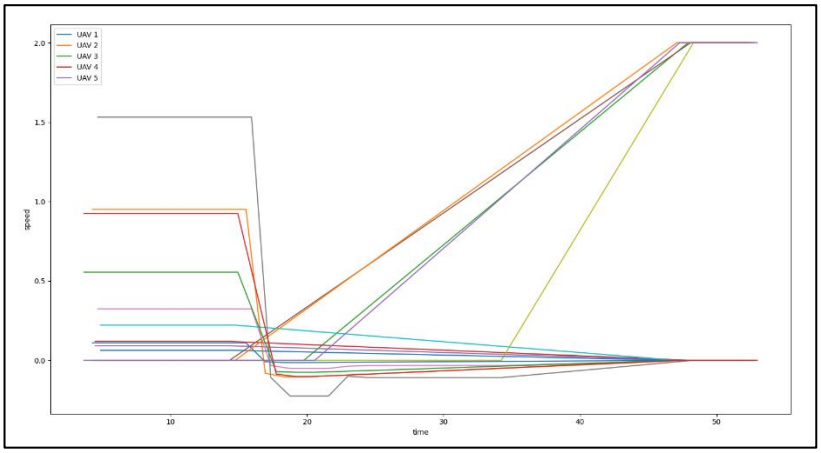
Şekil 4.22. 3 İHA'nın navigasyon ve rotasyon işlemi.

Şekil 4.22'de navigasyon noktaları arası geçiş işleminde ±2° ve konum takibinde Eşitlik 3.78'de verilen PID tabanlı ara nokta takip işleminde ±1,2778 metre hata ile üçgen formasyonundaki 3 İHA'nın 3 farklı ara nokta takibi gerçekleştirilmiştir. Konum kontrol işlemi PID kontrol işlemine ek olarak ivme kontrolü ve son ara noktada oluşan ivmenin sebep olduğu hatayı azaltmak için sadece Eşitlik 3.65'te verilen formasyon kontrol matrisi işleme alınmıştır. Kontrol çıktısının hata payı sürü merkezi ile navigasyon noktası arasında oluşan fark alınarak ölçülmüştür. Şekil 4.22'de deneysel testler de oluşturulan üçgen formasyonunda, GPS konum hatasından kaynaklı formasyonda bozulmalar oluşmaktadır.

Şekil 4.23 ve Şekil 4.24'de V formasyonu sırasında analiz sonuçlarının sadeleştirilmesi ve anlaşılması için simülasyon zamanı sıfırlanmıştır. İHA'ların V formasyonunda uçuş görevlerinin tamamlanması ve başlangıç pozisyonlarına gitme komutu ile izlediği konum, hız, zaman grafikleri incelenmiştir.



Şekil 4.23. V formasyonundan başlangıç konumlarına geçiş xyz grafiği.



Şekil 4.24. V formasyonundan başlangıç konumlarıan geçiş hız-zaman grafiği.

Şekil 4.23 ve Şekil 4.24'de 5 İHA'V formasyonunda uçuş yapan 5 İHA'nın formasyon görevinin bitmesi ile başlangıç konumlarına, çarpışma olmadan ilerlediği gözlemlenmiştir.

Bu tez çalışmasında, kümelenme işleminde 8 farklı geometrik şeklin matematiksel modeli çıkarılarak formasyon işlemleri başarılı bir şekilde tamamlanmıştır. Formasyon kontrol işleminde PD tipi kontrol çıktısı dalgalanmayı azaltmak için Kp ve Kd katsayılarının parçacık sayılarına bölünerek ve sürü kontrol hız çıktısıyla eşitlenerek başarılı bir şekilde formasyonu koruyarak rotayı takip etmiştir.

Kontrol yöntemlerinde İHA'lar arası mesafeye ve iletişim yöntemine bağlı olarak birim zamanın (hesaplama işlemlerinde oluşan zaman farkının) doğru hesaplanması için İHA'ların iletişiminde konum vb. bilgiler alınırken zaman aktarılmıştır. Konum, hız ve ivme tahmin işlerinde ortalama zaman faktörü hesaplanmıştır. Kontrol işlemlerinde kullanılan birim zamanda oluşan farklılıklar, İHA'lara gönderilen istenilen hız ile İHA'nın uyguladığı hız da farklılıklara sebep olacağından hata değerinin artmasına ve istenilen kontrol çıktısına daha geç ulaşmasına sebep olmaktadır.

Merkezi kontrol, tüm İHA'ların yer istasyonu ile iletişim kurması ve komutları işlemesi ile gerçekleştirilmektedir. Bu ağ yapısı İHA sürüsünün kapsama alanı ve boyutu az olduğundan sürü kontrol arayüzü geliştirilerek kullanılmıştır. Sürü üyelerinin merkezi olmayan kontrolünde (sadece birbirleri ile etkileşim halinde), oluşturulan ağ geçidi ile her İHA bir düğüme bağlanarak, diğer İHA'ların bilgilerine ulaşmaktadır. Bu şekilde menzil dışında kalan üyelerin konum bilgileri farklı düğüm noktaları ile aktarılması sağlanmıştır.

Merkezi olmayan kontrol yönteminde, görev gereksinimlerinin uygulanması ve yönetimi için her bireye görev talimatları göndermek sistem performansını düşürmektedir. Bu sebeple sürü lideri konsepti geliştirilmiştir. Liderin belirlenmesi her İHA'nın sürüdeki en iyi konumuna, hedef ile olan uzaklığına, konumuna ve farklı formasyon oluşumlarına göre oluşturulan veri seti ile tek katmanlı yapay sinir ağı modeli uygulanmıştır. Tek noktalı merkezi kontrol yöntemine göre görev gereksinim talimatlarını alan sürü liderinin, ağ içerisindeki diğer sürü bireylerine yükleme işlemi daha hızlı ve kararlı olmaktadır.

BÖLÜM 5

SONUÇLAR

Bu çalışmada, İHA'ların sürü kontrolü PSO algoritması ile uygulanmıştır. Formasyon-kümelenme işlemleri için gerekli algoritmalar oluşturulmuş ve gerçek zamanlı atama işlemi ile formasyon noktalarına atama sonrasında PD tipi kontrolcü ile formasyon kontrolü sağlanmıştır. Simülasyon testlerinin ve sanal gerçek entegre uygulamalarının sonucunda;

1. Sürü kontrolü ve kümelenme işlemlerinde, mutlak veya göreli pozisyon hassasiyeti sistem performansına doğrudan etki etmektedir. Fiziksel sistemde kullanılan GPS modülü 5m hassasiyete sahip olduğundan, hassasiyeti arttırmak için RTK (Gerçek Zamanlı Kinematik Uydu Navigasyonu) kullanılabilir. RTK maliyeti ve sabit olması sebebiyle, konum yerelleştirme, haritalama (SLAM) veya konum tahmini işlemleri ile konum hassasiyeti arttırabilir.
2. Rastgele dağılım için sadece sürü kontrol ve çarpışma engelleme algoritmaları çalıştırılarak parçacıkların sürü merkezi odaklı kümelenme işlemi gerçekleştirdiği gözlemlenmiştir.
3. Formasyon noktalarına atama problemi, her noktanın, her İHA ile eşleştirilmesi seçim işlemi olduğundan parçacık sayısı arttıkça atama performansı azalmaktadır.
4. Sürü kontrol işleminde kullanılan PSO algoritmasının, en iyileme problemine çözümünde diğer sürü üyeleriyle etkileşimi gerektiğinden harici iletişim kurallarının kullanılması ve merkezi kontrol işlemlerinde atama ve sürü kontrol arama uzayında n^2 problemi oluşturduğundan (her parçacık sayısı arttıkça her parçacık için oluşturulan matrisin büyüme hızı) kullanılan sunucu özelliklerine göre dezavantaj oluşturmaktadır.
5. Sürülerin yer istasyonundan kontrolü için geliştirilen sürü kontrol arayüz yazılımı Python dilinde oluşturularak her parçacık için sistem belleği ayrılarak,

görev işlemleri eş zamanlı gerçekleştirilmektedir. Merkezi kontrol yazılımında, paralel ve eş zamanlı işlemlerin gereksinimi sebebiyle eş zamanlılık problemine çözüm olarak Python programlama dilinde gerçekleştirilen sürü yer kontrol yazılımı, Go programlama dilinde karşılaştırılmıştır. Yapılan testler de Goroutine sürü kontrolünde esneklik ve performans sağladığı gözlemlenmiştir.

6. Geliştirilen sistemde merkezi kontrol etkisini arttırılması amacıyla sürü İHA'ların birbirleri ile iletişim kurması için oluşturulan ağ geçidinin mobil internet bağlantısı sağlanarak eş zamanlı olarak bulut sunucusuna yüklenmesi ile Wi-Fi veya radyo iletişimindeki etki mesafesi sorunu çözülebilir.
7. Çalışmada, lider kontrolündeki sürü üyelerinin, etkileşimleri oluşturulan Wi-Fi ağının bağlantı kalitesi ve menziline bağlı olduğundan, bireyler arası mesafe ağ menziline bağlıdır. Kullanılan Nodemcu modülü dahili antene sahip olması etki mesafesini azaltmaktadır. Bu sebeple, harici anten veya farklı Wi-Fi modemleri ile etki alanı arttırılabilir.

Simülasyon çalışmaları ve deneysel sonuçlar, merkezi kontrol işleminin tek noktaya (yer kontrol istasyonu) bağlı olması, bağlantının zaman aşımına uğraması durumunda tüm sürü üyelerini etkilediği gözlemlenmiştir. Dağıtık model benzetimi olarak, lider kontrolü (yarı dağıtık) sistemde, görev paketlerini aktarma işlemi sırasında oluşan gecikme ve kayıplar sürü uçuşunu etkilememektedir. Sonuçlar, merkezi olmayan sistemin etkinlik ve performansının daha üstün olduğunu göstermektedir.

KAYNAKLAR

1. Kahveci, M., and Can, N., "İnsansız hava araçları: tarihçesi, tanımı, dünyada ve Türkiye'deki yasal durumu", ***Selcuk University Journal Of Engineering, Science And Technology***, 5 (4): 511–535 (2017).

2. Bonebau, E., Dorigo, M., and Theraulaz, G., "Swarm intelligence: From natural to artificial systems", ***Swarm Intelligence: From Natural To Artificial Systems***, 15 (1): (1999).

3. Boskovic, J. D., Prasanth, R., and Mehra, R. K., "A multi-layer control architecture for unmanned aerial vehicles", ***Proceedings Of The American Control Conference***, 3: 1825–1830 (2002).

4. Garnier, S., Gautrais, J., and Theraulaz, G., "The biological principles of swarm intelligence", ***Swarm Intelligence***, 1 (1): 3–31 (2007).

5. Kennedy, J., and Eberhart, R., "Particle swarm optimization", (1995).

6. Dorigo, M., "Optimization, learning and natural algorithms", ***PhD Thesis, Politecnico Di Milano***, (1992).

7. Bayındır, L., and Şahin, E., "Modeling self-organized aggregation in swarm robotic systems", ***2009 IEEE Swarm Intelligence Symposium***, 88–95 (2009).

8. Mayya, S., Wilson, S., and Egerstedt, M., "Closed-loop task allocation in robot swarms using inter-robot encounters", ***Swarm Intelligence***, 13 (2): 115–143 (2019).

9. Prasetyo, J., de Masi, G., and Ferrante, E., "Collective decision making in dynamic environments", ***Swarm Intelligence***, 13 (3–4): 217–243 (2019).

10. Zhang, S., Liu, M., Lei, X., Huang, Y., and Zhang, F., "Multi-target trapping with swarm robots based on pattern formation", ***Robotics And Autonomous Systems***, 106: 1–13 (2018).

11. Oh, H., Shirazii, A. R., Sun, C., and Jin, Y., "Bio-inspired self-organising multi-robot pattern formation: A review", ***Robotics Auton. Syst.,*** 83-100. (2017).

12. Nedjah, N., and Junior, L. S., "Review of methodologies and tasks in swarm robotics towards standardization", ***Swarm And Evolutionary Computation***, 50 (100565): (2019).

13. Yang, H., Cao, S., Bai, L., Zhang, Z., and Kong, J., "A distributed and parallel self-assembly approach for swarm robotics", ***Robotics And Autonomous Systems***, 118 (6198): 80–92 (2019).

14. Vicsek, T., and Zafeiris, A., "Collective motion", ***Physics Reports***, 517 (3–4): 71–140 (2010).

15. Dollarhide, R. L., and Agah, A., "Simulation and control of distributed robot search teams", ***Computers And Electrical Engineering***, 29 (5): 625–642 (2003).

16. Edgeworth, F. Y., "Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences", 88–89 (1881).

17. Pareto, V., "Cours d'Economie Politique", 426–430 (1896).

18. Breder, C. M., "Equations descriptive of fish schools and other animal aggregations. Ecology", ***Ecology***, 35: 361–370 (1954).

19. Warburton, K., L. J., "Tendency-distance models of social cohesion in animal groups. Journal of theoretical biology", ***Journal Of Theoretical Biology***, 150 (4): 473–488 (1991).

20. Okubo, A., "Dynamical aspects of animal grouping: swarms, schools, flocks, and herds. Advances in biophysics", ***Advances In Biophysics***, 22: 1–94 (1991).

21. Grünbaum, D., and Okubo, A., "Modeling social animal aggregations: Frontiers in mathematical biology", 100: 296–332 (1994).

22. Grünbaum, D., "Schooling as a strategy for taxis in a noisy environment. Evolutionary ¨ ecology", 12: 503–522 (1998).

23. Parrish, J. K., Viscido, S. V., and Grünbaum, D., "Self-organized fish school: An examina- ¨ tion of emergent properties. Biol. bull.", 202: 296–305 (2002).

24. Gazi, V., and Passino, K. M., "Stability analysis of swarms. IEEE trans. on automatic control", 48 (4): 692–697 (2003).

25. Gazi, V., and Passino, K. M., "A class of attraction/repulsion functions for stable swarm aggregations. Int. J. Control", 77 (18): 1567–1579 (2004).

26. Gazi, V., and Passino, K. M., "Stability analysis of social foraging swarms. IEEE trans. on systems, man, and cybernetics", 34 (1): 539–557 (2004).

27. Liu, Y., and Passino, K. M., "Stable social foraging swarms in a noisy environment. IEEE transactions on automatic control", 49 (1): 30–44 (2004).

28. Reynolds, C. W., "Flocks, herds, and schools: A distributed behavioral model. Comp. Graph", (1987).

29. Balch, T., and Arkin, R. C., "Behavior-based formation control for multi-robot teams", (1999).

30. Vicsek, T., Czirok, A., Ben-Jacob, E., Cohen, I., and Shochet, O., "Novel type of phase transition in a system of self-driven particles", (1994).

31. Buhl, J., Sumpter, D. J. T., Couzin, I. D., Hale, J. J., Despland, E., Miller, E. R., and Simpson, S. J., "From disorder to order in marching locusts", ***Science***, 312 (5778): 1402–1406 (2006).

32. Ben-Jacob E., Cohen I., and Levine H., "Cooperative self-organization of microorganisms", 49: 395–554 (2000).

33. Afsharizand, B., Chaghoei, H., Kordbacheh, A. A., Trufanov, A., and Jafari, G., "Market of stocks during crisis looks like a flock of birds", (2020).

34. Grossman, D., Aranson, I. S., and ben Jacob, E., "Emergence of agent swarm migration and vortex formation through inelastic collisions", ***New Journal Of Physics***, 10: (2008).

35. Lien, J., Bayazit, O.B., Sowell, R.T., Rodríguez, S., and Amato, N.M., "Shepherding behaviors", 4159–4164 (2004).

36. Lien, J., Rodríguez, S., Malric, J., and Amato, N.M., "Shepherding behaviors with multiple shepherds", 3402-3407 (2005).

37. Aldana, M., and Huepe, C., "Phase transitions in self-driven many-particle systems and related non-equilibrium models: A network approach", (2003).

38. Mermin, N. D., and Wagner, H., "Absence of ferromagnetism or antiferromagnetism in one or two-dimensional isotropic heisenberg models", 17: 1133–1136 (1966).

39. Samiloglu, A. T., Gazi, V., and Bugra Koku, A., "Comparison of three orientation agreement strategies in self-propelled particle systems with turn angle restrictions in synchronous and asynchronous settings", ***Asian Journal Of Control***, 10 (2): 212–232 (2008).

40. Güzel, M. S., Ajabshir, V. B., Nattharith, P., Gezer, E. C., and Can, S., "A novel framework for multi-agent systems using a decentralized strategy", 37 (4): 691–707 (2019).

41. Schmickl, T., Thenius, R., Möslinger, C., Radspieler, G., Kernbach, S., Szymanski, M., and Crailsheim, K., "Get in touch: Cooperative decision making based on robot-to-robot collisions", ***Autonomous Agents And Multi-Agent Systems***, 18 (1): 133–155 (2008).

42. Schmickl, T., and Hamann, H., "BEECLUST: A swarm algorithm derived from honeybees. Derivation of the algorithm, analysis by mathematical models and implementation on a robot Swarm", (2010).

43. Collett, T. S., and Collett, M., "Memory use in insect visual navigation", ***Nature Reviews Neuroscience***, 3 (7): 542–552 (2002).

44. Arvin, F., Turgut, A. E., Krajník, T., and Yue, S., "Investigation of cue-based aggregation in static and dynamic environments with a mobile robot swarm", ***Adaptive Behavior***, 24 (2): 102–118 (2016).

45. Wahby, M., Petzold, J., Eschke, C., Schmickl, T., and Hamann, H., "Collective change detection: adaptivity to dynamic swarm densities and light conditions in robot swarms", (2019).

46. Vardy, A., "Aggregation in robot swarms using odometry. Artificial life and robotics", 21 (4): :443-450 (2016).

47. Amjadi, A. S., Raoufi, M., and Turgut, A. E., "A self-adaptive landmark-based aggregation method for robot swarms", ***Adaptive Behavior***, (2021).

48. Condliffe, J., "A 100-drone swarm, dropped from jets, plans its own moves", (2017).

49. Kelly K., "Out of control: The rise of neo-biological civilization", (1994).

50. Molina, B., "Drones from super bowl 51 halftime show", ***USA TODAY***, (2017).

51. Chung, S.-J., Paranjape, A., Dames, P., Shen, S., and Kumar, V., "A survey on aerial swarm robotics", (2018).

52. Bürkle, A., Segor, F., and Kollmann, M., "Towards autonomous micro UAV swarms", 61: . 339-353 (2011).

53. Worth, D. B., Woolley, B. G., and Hodson, D. D., "SwarmSim: A framework for modeling swarming unmanned aerial vehicles using Hardware-in-the-Loop", ***Journal Of Defense Modeling And Simulation***, 18 (2): 105–124 (2021).

54. Bekmezci, I., Sahingoz, O. K., and Temel, Ş., "Flying adHoc networks (FANETs): A survey", 11 (3): 1254–1270 (2013).

55. Sahingoz, O. K., "Networking models in flying Ad-hoc networks (FANETs): Concepts and challenges", ***Journal Of Intelligent And Robotic Systems: Theory And Applications***, 74 (1–2): 513–527 (2014).

56. Sivakumar, A., Keng, C., and Tan, Y., "UAV swarm coordination using cooperative control for establishing a wireless communications backbone", (2010).

57. Teague, E., and Kewley, R.H., "Swarming unmanned aircraft systems", (2008).

58. Zhou, Y., Li J., Lamont, L., and Rabbath, C., "Modeling of packet dropout for UAV wireless communications", ***International Conference on Computing, Networking and Communications (ICNC)***, 12: 677–682 (2012).

59. Internet: Ardupilot, " Swarming - Mission Planner documentation", **https://ardupilot.org/planner/docs/swarming.html** (2021).

60. Girolami, M., Bacco, M., Chessa, S., and Benedetto, M. di, "UAVs and UAV swarms for civilian applications: Communications and image processing in the sciadro project", (2017).

61. Gaudiano, P., Shargel, B., Bonabeau, E., and Clough, B. T., "Swarm intelligence: A new c2 paradigm with an application to control of swarms of UAVs", (2003).

62. Corner, J. J., and Lamont, G. B., "Parallel simulation of UAV swarm scenarios", (2004).

63. Lamont, G. B., Slear, J. N., and Melendez, K., "UAV swarm mission planning and routing using multi-objective evolutionary algorithms", (2007).

64. Chung, T. H., Jones, K. D., Dayc, M. A., Jones, M., and Clement, M., "Swarm UAV live-fly competition at the naval postgraduate school", (2013).

65. Campion, M., Ranganathan, P., and Faruque, S., "A review and future directions of UAV swarm communication architectures", (2018).

66. Chen, W., Liu, B., Huang, H., Guo, S., and Zheng, Z., "When UAV swarm meets edge-cloud computing: The QOS perspective", 33 (2): 36–43 (2019).

67. Zhu, X., Liu, Z., and Yang, J., "Model of collaborative UAV swarm toward coordination and control mechanisms study", (2015).

68. Demirci, R., and Uslu, E., "Resultant force approach for swarm UAV behaviors", ***2021 Innovations In Intelligent Systems And Applications Conference (ASYU)***, (2021).

69. Belkadi, A., Ciarletta, L., and Theilliol, D., "Particle swarm optimization method for the control of a fleet of unmanned aerial vehicles", (2015).

70. Chen, H., and Fei, J., "UAV path planning based on particle swarm optimization with global best path competition", ***International Journal Of Pattern Recognition And Artificial Intelligence***, 32 (1): (2017).

71. Zhou, X., Gao, F., Fang, X., and Lan, Z., "Improved bat algorithm for UAV path planning in three-dimensional space", ***IEEE Access***, 9: 20100–20116 (2021).

72. Ghamry, K. A., Kamel, M. A., and Zhang, Y., "Multiple UAVs in forest fire fighting mission using particle swarm optimization", (2017).

73. Aniket, G., Aman, V., Parth, M., and Raghava, N., "A particle swarm optimization-based cooperation method for multiple-target search by swarm UAVs in unknown environments", (2021).

74. Liu, J., He, M., Luo, L., Liu, Q., and Zou, M., "Pairwise control in unmanned aerial vehicle swarm flocking", 2020: 1 (2021).

75. Petráček, P., Walter, V., Báča, T., and Saska, M., "Bio-inspired compact swarms of unmanned aerial vehicles without communication and external localization", (2020).

76. Turgut, M. N., "Dört rotorlu insansız hava aracının modellenmesi ve simülasyonu", (2011).

77. Silva, D. H. C., Santos, M. F., Silva, M. F., Neto, A. F. S., and Mercorelli, P., "Design of controllers applied to autonomous unmanned aerial vehicles using software in the loop", (2019).

78. Pehlivan, K. and Aküner, M. C., "Quadrotor test düzeneği için PID kontrolör tasarımı ve uygulaması", ***Marmara University***, 2 (1): 15–24 (2020).

79. Bozkurt, E., Dandıl, B. and Ata, F., " Dört Rotorlu İnsansız Hava Aracının Kayan Kipli Denetleyici ve Geri Adımlamalı Denetleyici ile Yönelim ve Yükseklik Denetimi", 23-36 (2020).

80. Mutlu, İ. and Kahveci, M., "GNSS uydu dağılımının gerçek zamanlı kinematik gnss ve ağ-rtk ölçülerindeki önemi", ***Geomatik***, (2019).

81. Ebeid, E., Skriver, M., and Jin, J., "A survey on open-source flight control platforms of unmanned aerial vehicle", (2017).

82. Pranoto, I., Naiborhu, J., and Achmadi, S., "Formation control of multiple dubin’s car system with geometric approach", (2012).

83. Kuhn, H. W., "The hungarian method for the assignment problem", (1995).

84. Oh, K. K., Park, M. C., and Ahn, H. S., "A survey of multi-agent formation control", ***Automatica***, 53: 424–440 (2015).

85. Turpin, M., Michael, N., and Kumar, V., "CAPT: Concurrent assignment and planning of trajectories for multiple robots", ***The International Journal Of Robotics Research***, 33 (1): 98–112 (2014).

86. Gazi, V., Fİdan, B., Hanay, Y. S., and Köksal, İ., "Aggregation, foraging, and formation control of swarms with non-holonomic agents using potential functions and sliding mode techniques", 15(2): 149–168 (2007).

87. Yao, J., Ordóñez, R., and Gazi, V., "Swarm tracking using artificial potentials and sliding mode control", (2006).

88. Tanner, H. G., Pappas, G. J., and Kumar, V., "Leader-to-formation stability", (2004).

89. Miswanto, Pranoto, I., Muhammad, H., and Mahayana, D., "The control design of ship formation with the presence of a leader", ***International Journal Of Robotics And Automation (IJRA)***, 4 (1): 53–62 (2015).

90. Desai, J. P., Ostrowski, J. P., Vijay Kumar, R., Vijay, R., and Kumar, V., "Modeling and control of formations of nonholonomic mobile robots", (2001).

91. Xie, M., Laugier, C., Jiangzhou, L. U., and Sakhavat, S., "Sliding mode control for nonholonomic mobile robot", (2000).

92. Mancini, A., Zingaretti, P., Orlando, G., Frontoni, E., Mancini, A., and Zingaretti, P., "Sliding mode control for vision based leader following", (2007).

93. Mısır, O., and Gökrem, L., "Sürü robotları için esnek ve ölçeklenebilir toplanma davranışı metodu", ***European Journal Of Science And Technology***, 100–109 (2020).

94. Sial, M. B., Ding, W., Wang, S., Wang, X., Wyrwa, J., and Liao, Z., "Mission oriented flocking and distributed formation control of UAVs", (2021).

95. Mechali, O., Jamshed, I., Jingxiang, W., Xiaomei, X., and Limei, X., "Distributed leader-follower formation control of quadrotors swarm subjected to disturbances", ***2021 IEEE International Conference On Mechatronics And Automation (ICMA)***, (2021).

96. Gauci, M., Chen, J., Li, W., Dodd, T. J., and Groß, R., "Self-organized aggregation without computation", ***International Journal Of Robotics Research***, 33 (8): 1145–1161 (2014).

97. Mısır, O., Gökrem, L., and Can, M. S., "Fuzzy-based self organizing aggregation method for swarm robots", (2020).

98. Parhizkar, M., Di Marzo Serugendo, G., Nitschke, J., Hellequin, L., Wade, A., and Soldati, T., "First-order agent-based models of emergent behaviour of dictyostelium discoideum and their inspiration for swarm robotics: A selection of aggregation phase behaviour with biological illustrations", ***Artificial Life And Robotics***, 25 (4): 643–655 (2020).

99. Ali, Q., and Montenegro, S., "Decentralized Control for Scalable Quadcopter Formations", ***International Journal Of Aerospace Engineering***, 2016: (2016).

100. Toksöz, M. A., Oğuz, S., and Gazi, V., "Decentralized Formation Control of a Swarm of Quadrotor Helicopters", ***2019 IEEE 15th International Conference On Control And Automation (ICCA)***, (2019).

101. Waydo, S., and Murray, R. M., "Vehicle motion planning using stream functions", (2003).

102. Ye, G., and Wang, O., "Coordinated motion control of swarms with dynamic connectivity in potential flows", 38 (1): (2005).

103. Merheb, A. R., Gazi, V., and Sezer-Uzol, N., "Implementation studies of robot swarm navigation using potential functions and panel methods", ***IEEE/ASME Transactions On Mechatronics***, 21 (5): 2556–2567 (2016).

104. Huang, Y., Tang, J., and Lao, S., "Collision avoidance method for self-organizing unmanned aerial vehicle flights", (2019).

105. Xu, W., Xiang, L., Zhang, T., Pan, M., and Han, Z., "Cooperative control of physical collision and transmission power for UAV swarm: A dual-fields enabled approach", (2021).

106. Muntasha, G., Karna, N., and Shin, S. Y., "Performance analysis on artificial bee colony algorithm for path planning and collision avoidance in swarm unmanned aerial vehicle", (2021).

107. Zhou, D., Wang, Z., Bandyopadhyay, S., and Schwager, M., "Fast, on-line collision avoidance for dynamic vehicles using buffered voronoi cells", ***IEEE Robotics And Automation Letters***, 2 (2): 1047–1054 (2017).

108. Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U., and Stryk, O.V., "Comprehensive simulation of quadrotor UAVs using ROS and GAZEBO", (2012).

109. Yasin, J. N., Haghbayan, M.-H., Yasin, M. M., and Plosila, J., "Swarm Formation Morphing for Congestion Aware Collision Avoidance", (2020).

110. Bachmayer, R. L. E., "Vehicle networks for gradient descent in a sampled environment", (2002).

111. Tanner, H. G., Jadbabaie, A., and Pappas, G. J., "Stable flocking of mobile agents, part I: Fixed topology", (2003).

112. Tanner, H. G., Jadbabaie, A., and Pappas, G. J., "Stable flocking of mobile agents, part II: Dynamic topology", (2003).

113. Leonard, N. E., and Fiorelli E., "Virtual leaders, artificial potentials and coordinated control of groups,", ***Proceedings of the 40th IEEE Conference on Decision and Control***, 2968–2973 (2001).

114. Olfati-Saber, R., "Flocking for multi-agent dynamic systems: Algorithms and theory", ***IEEE Transactions On Automatic Control***, 51 (3): 401–420 (2006).

115. Czirók, A., Barabási, A. L., and Vicsek, T., "Collective motion of self-propelled particles: Kinetic phase transition in one dimension", ***Physical Review Letters***, 82 (1): 209–212 (1999).

116. Meng, X. B., Gao, X. Z., Lu, L., Liu, Y., and Zhang, H., "A new bio-inspired optimisation algorithm: Bird swarm algorithm", ***Journal Of Experimental And Theoretical Artificial Intelligence***, 28 (4): 673–687 (2016).

117. Freeman, P., "Minimum jerk trajectory planning for trajectory constrained redundant robots", (2012).

118. Joy, K. I., "Quadratic B´ezier Curves", (2000).

119. Arı, A., and Berberler, M. E., "Yapay sinir ağları ile tahmin ve sınıflandırma problemlerinin çözümü için arayüz tasarımı", (2017).

120. Winarno, E., Hadikurniawati, W., and Rosso, R. N., "Location based service for presence system using haversine method", ***In 2017 international conference on innovative and creative information technology (ICITech)***, 1–4 (2017).

121. Ligas, M., "Cartesian to geodetic coordinates conversion on a triaxial ellipsoid. Journal of Geodesy", 86(4): 249–256 (2012).

ÖZGEÇMİŞ

Batuhan KARAÇAL Yeşilkent Anadolu Lisesi'nden mezun oldu. 2013 yılında Karabük Üniversitesi Mühendislik Fakültesi Metalurji ve Malzeme Mühendisliği Bölümü'nde öğrenime başlayıp 2016 yılında Mekatronik Mühendisliği bölümüne yatay geçiş yaparak, 2018 yılında lisans eğitimini tamamladı. 2019 yılında Karabük Üniversitesi Lisansüstü Eğitim Enstitüsü Mekatronik Mühendisliği Anabilim Dalı'nda yüksek lisans programına başladı ve eğitimine devam etmektedir.