

# DRONLAR İÇİN OTOMATİK ROTA TAYİNİ VE TAKİBİ

Şehmus GÖKÇE, Aydın ÇETİN

<sup>1</sup>Teknoloji Fakültesi/Bilgisayar Mühendisliği, Gazi Üniversitesi, Türkiye  
<sup>2</sup>acetin@gazi.edu.tr

<sup>\*</sup>sehmus.gokce@gazi.edu.tr

**Abstract** – Günümüze kadar dronlar insan yönetimine ihtiyaç duyan uzaktan kumandalı cihazlar olarak gündeme gelmişlerdir. Dronlar, yaygın olarak insan desteğiyle çalışmakta olup ek iş gücüne, maliyet artışına sebep olmaktadır. Hatalı kullanım sonucu dronlar hasar görebilmektedir. Bu bildiride dronların otomatik rota tayini ve takibi önerilmektedir. Bu amaçla dronların takibi GPS tabanlı olarak yapılmakta ve belirlenen rotayı takip etmesi sağlanmaktadır. Bu şekilde kullanıcı devre dışı bırakılarak dronların belirlenen görevi yerine getirmesi ve insan kaynaklı kullanım hatalarının ortadan kaldırılması sağlanmaktadır. Uygulamada güvenlik amaçlı olarak öncelikle dronekit kütüphanesi kullanılarak simülasyon ortamında dronun çalışması modellenmiş ve performansı değerlendirilmiştir. Daha sonra modellenen dronun başarı testi Gazi Üniversitesi Merkez kampüsünde atanan rota üzerinden gerçekleştirilmiştir. Dronun kendisine verilen rotayı sorunsuz bir şekilde takip ettiği gözlemlenmiştir.

**Keywords** – Drone, rota takibi, otonom

## I. GİRİŞ

Günümüze kadar dronlar insan yönetimine ihtiyaç duyan uzaktan kumandalı cihazlar olarak gündeme gelmişlerdir. Askeri amaç dışında hobi olarak da kullanılmaya başlanan dronların kullanım oranları günden güne artmıştır. Dron kullanımındaki başlıca problemler, deneyimli ve eğitimli operatör gereksinimi, operatör ile cihaz arasındaki iletişimin kopma olasılığı olarak tanımlanabilir. Bu durumda operatör maliyetleri artmakta ve aynı zamanda gerek operatör kaynaklı, gerekse iletişim kopması kaynaklı kazalar oldukça yüksek maliyete sebep olabilmektedir. Dronun hatasız çalışması büyük bir önem teşkil etmektedir. Bu cihazlar yerleşim yerlerinde uçabilmekte, ticari amaçla örneğin bir yerden başka bir yere kargo taşımacılığı amacıyla kullanılabilir. Herhangi bir hata veya kırım yaşanması insanlar için büyük ölçüde tehlike yaratacaktır.

Dronların bir yerden başka bir yere kazasız ve doğru bir şekilde olarak yönlendirilmesi, aynı şekilde başlangıç noktasına sorunsuz bir şekilde iniş yapmasını sağlamak, kullanım esnasında herhangi bir tehdit oluşturmaması temel hedeftir. Günümüz teknolojisinde dronlara ilave edilecek algılayıcı ve modüllerle bu hedeflerin yerine getirilmesi mümkündür. Ancak bu durum sisteme ek maliyet getirmektedir. Ortaya çıkan ek maliyet operatör maliyeti ve iletişimin kopması sonucu ortaya çıkabilecek hasar kaynaklı maliyetlerle kıyaslandığında ihtimal edilebilecek düzeydedir.

Bu bildiride herhangi bir operatöre gerek kalmadan dronun atanan rotayı izlemesi ve sorunsuz bir şekilde başlangıç noktasına dönmesi hedeflenmiştir. Bu amaçla cihaza eklenecek GPS modülü ve sensörler aracılığıyla otomatik olarak işletilmesi hedeflenmiştir.

## II. MATERYAL VE METOD

Dronun çalışması için temel fiziksel bileşenler; enerji kaynağı, denetleyici, algılayıcılar ve modüller olarak tanımlanmıştır.

Bu çalışmada denetleyici olarak yaygın olarak kullanılan PixHawk uçuş denetleyicisi, enerji kaynağı olarak 5200mAh 3S Li-Po bir batarya tercih edilmiştir. Dronun konumunu belirlemek üzere GPS modülü, engelleri algılamak için ultrasonik sensörler, yüksekliği belirlemek için barometre, dronun havada stabilitesini sağlamak amacıyla akış sensörü tercih edilmiştir.

Uygulamada dronun denetimini sağlayacak gerekli yazılım geliştirmek amacıyla DroneKit kütüphanesi, cihazın anlık sensör ve durum bilgilerini gözlemlemeyi sağlamak amacıyla uçuş denetleyicisi ile birlikte çalışmak üzere Mission Planner kullanılmıştır. Otonom uçuşlarının başarılı bir şekilde çalışıp çalışmadığını test etmek ve uçuşu gerçekleştirmek için iki farklı çalışma ortamı, simülasyon ve gerçek sistem, tasarlanmıştır.

Simülasyon ortamında açık kaynak kodlu ArduCopter sanal dronu kullanılmıştır. Simülasyon Windows 10 ortamına kurularak Mavlink altyapısında MavProxy aracılığıyla ArduCopter derlenenek sanal simülasyon dronu oluşturulmuştur. Oluşturulan sanal simülasyon dronu Mission Planner üzerinden izlenmektedir.

Test sahası olarak başlangıçta gerçek sistemin de test edileceği düşünülerek, güvenlik amacıyla Gazi Üniversitesi kampüsünde yer alan futbol sahası seçilmiştir. Böylece planlanan koordinatlarda dronun davranışı test edilmiştir.

İkinci test fiziki bir sistem üzerinde test edilmektedir. Dronun anlık bilgilerini uzak bir bilgisayara iletebilmek amacıyla 455 Mhz'lik telemetry modülü sisteme ilave edilmiştir. Türkiyedeki 900 Mhz GSM sinyalleri ile karışmaması için 900 Mhz'lik bir telemetry modülü tercih

edilmemiştir. Telemetri modülü Mission Planner ile uyumlu çalışacak şekilde konfigüre edilmiştir. Geliştirilen yazılım uygulaması ile drona görev tanımlaması koordinat olarak verilmekte ve testlerin durumu Missin Planner üzerinden telemetri modülü aracılığı ile gözlemlenmiştir.

### III. SİMÜLASYON

Yapılan çalışma drona verilen görevlerin başarı ile otomatik olarak gerçekleştirilmesini esas almaktadır. Çalışma, ileride eklenebilecek OpenCV gibi yeni açık kaynak koduna ve kütüphanelere uyumlu olarak tasarlanmıştır. Bu amaçla Python yazılım dili tercih edilmiştir. Böylece OpenCV gibi bir görüntü işleme kütüphanesi kullanılmak istendiğinde sisteme entegrasyonu sorunsuz bir şekilde gerçekleştirilebilir.

#### A. Drona Verilecek Görevin ve Koordinatların Belirlenmesi

Kesin ve bilgilendirici sonuçları almak amacıyla materyal ve metod bölümünde ele alınmış olan iki adet simülasyon ortamı göz önünde bulundurulmuş ve koordinatlar belirlenmiştir.

Simülasyon ortamı olan ilk ortam için Gazi Üniversitesi'nin futbol sahasının etrafı koordinatlar olarak belirlenmiştir. Dronun sensörlerinin sorunsuz bir şekilde veri iletimini gerçekleştirdiğini test etmek amacıyla kalkış noktasının engelli bir yapıda olması önem teşkil etmektedir. Kalkış testini gerçekleştirmek amacıyla futbol sahasının kale arkası kalkış noktası olarak belirlenmiştir. Şekil 1'de Mission Planner'da simülasyon ortamındaki dronun başlangıç koordinatlarının futbol sahası içerisinde belirlenmesi görülmektedir. Seçilen kalkış noktasına ek olarak sahanın dört köşesi belirlenmiştir. Böylelikle dron, verilen bütün noktalara yönlendirildikten sonra başlangıç noktasına sorunsuz olarak inebilmesi gözlemlenip test edilebilecektir. Testlerin başarılı bir şekilde simülasyon ortamında gerçekleştirilmesi güvenlik baz alındığında fiziksel ortama geçişi kolaylaştıracaktır.



Şekil.1. Simülasyon ortamındaki dronun başlangıç koordinatlarının futbol sahası içerisinde belirlenmesi.

#### B. Simülasyon yazılımı

Oluşturulan Simülasyon yazılımının temeli DroneKit Framework üzerine kurulduğundan kütüphanenin sunduğu çok sayıda temel komut kullanılabilir. Bu temel

komutların çalıştırılabileceği araç kütüphane üzerinden Vehicle adıyla yazılım geliştiricilere sunulmaktadır. Böylelikle geliştirilen Vehicle objesi sayesinde anlık bilgiler yazılım çalışma zamanında kullanıcıya iletilebilmektedir. Vehicle objesi tarafından kullanıcıya aktarılacak anlık bilgiler doğrultusunda otomatik olarak verilecek komutların tayini sağlanabilmektedir. Örneğin pil seviyesinin belirlenen seviyeden düşük olması durumunda sistemin nasıl hareket edeceği veya anlık GPS sensör koordinatlarının belirlenememesi sonucu uçuşun başlayıp başlayamayacağı gibi görev tayinleri bu obje aracılığı ile programcı tarafından gerçekleştirilmektedir. Görev mekanizmaları simülasyon yazılım katmanında belirlendiği gibi fiziksel kontrolcü katmanında da tanımlanmaktadır.

Kütüphanenin yazılımcıya sunduğu diğer bir komut ise LocationGlobalRelative olarak adlandırılmaktadır. Temel altyapıda LocationGlobalRelative komutu oluşturularak drona istenilen sayıda görev koordinatları tayini ataması gerçekleştirilebilmektedir. Sistem bir döngü altyapısında çalıştığı için anlık olarak her komut sonrası Vehicle objesinin durumu sorgulanması gerekmektedir.

Sistemin çalışmasında Simple\_goto komutu önemli bir yere sahiptir. Simple\_goto komutu temelde bir LocationGlobalRelative olarak koordinatlara gerçek veya simülasyonda tanımlanmış dronu koordinatlara ulaştırmayı hedef alır. Python dilinin sıralı olarak çalışma özelliğinden dolayı belirlenen komutların birer birer gerçekleştirilip gerçekleştirilmediğinin gözlenmesi hayati önem taşımaktadır. Örnek vermek gerekirse Simle\_goto komutu sonrası ikinci Simple\_goto komutunun verilmeden önce drone'un gerçekten o koordinatlara ulaşıp ulaşmadığı bir yazılım döngüsü içerisinde test edilmektedir. Dron planlanan koordinata ulaştığı takdirde ikinci simple\_goto komutu kullanılmaktadır. Güvenli ve iyi düzeyde uçuşun temelleri göz önüne alındığında döngü sisteminin hatasız olarak kullanılması büyük önem arz etmektedir.

Simülasyon yazılımının eksiksiz olarak çalıştırılıp test edilebilmesi için içerisinde uzaktan kontrol edilebilir olarak tanımlanan bir çok aracın tanımlanmış olduğu ArduPilot projesi GitHub açık kaynak kod paylaşım platformu tarafından test sistemine yüklenmiştir. Proje sisteme derlenmemiş açık kaynak kodlu olarak yüklendiği için test sistemi üzerinde derlenmesi önem arz etmektedir. ArduPilot içerisindeki hava aracı olarak ArduCopter seçilerek amaca uygun bir testin gerçekleştirilmesi amaçlanmaktadır. Simülasyon uygulamasının başlangıç koordinatları locations.txt dosyası içerisinde Gazi Üniversitesi merkez kampüsü futbol sahasının etrafı olarak seçilmiştir.

Hazırlanmış olan yazılımın drone üzerinde çalıştırılması için Linux komutlarını Windows 10 test sistemi üzerinde çalıştırmak için kullanılan Cygwin girdi olarak \$sim\_vehicle.py -map -console komutunu almaktadır.

Cygwin tarafından sim\_vehicle.py dosyası çalıştırıldıktan sonra map ve console adlı iki adet parametre simülasyon sistemine gönderilmektedir. Konsol ve harita uygulamalarını anlık olarak izlemeyi kolaylaştıran komut uygulamaları kullanıcıya sunmak için oluşturulmuşlardır. Test sisteminde

Mission Planner gözlem uygulaması kullanıldığından bu parametrelerin kullanımı önem teşkil etmemektedir.

Şekil 2'de belirtildiği gibi Missin Planner Uygulamasının ve yazılan python uygulamasının birbiriyle bağlanması için MAVProxy üzerinden iki adet portun sistem üzerinde ayrılmış olması gerekmektedir. Öngörülen olarak bir adet 14550 numaralı port kullanıma sunulmuştur. İkinci port ise MAVProxy komut satırı üzerinden açılmaktadır.

```
C:\Program Files (x86)\MAVProxy\mavproxy.exe
Connect tcp:127.0.0.1:5760 source_system=255
Loaded module console
Loaded module map
Running script (C:\Users\sehmu\AppData\Local\MAVProxy\mavinit.scr)
Loaded module help
Unknown command 'graph timespan 30'
Log Directory:
Telemetry log: mav.tlog
MAV> output add 127.0.0.1:14552
STABILIZE> Adding output 127.0.0.1:14552
GUIDED>
```

Şekil.2. Bir adet çıkış portunun açılması.

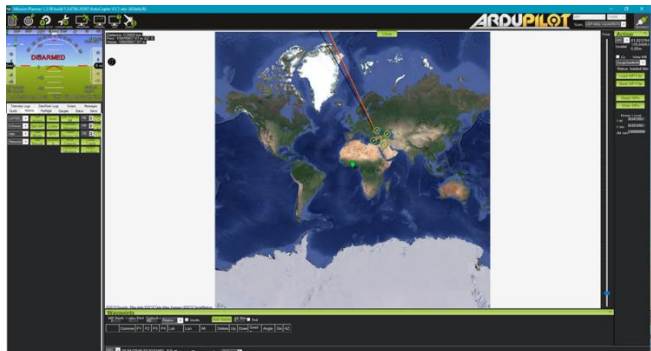
Çıkış portunun açılması sonucu Mission Planner üzerinden UDP 14552 bağlantısı yapılarak (Şekil 3) çalıştırılması sağlanmıştır. Tanımlanan ortamlarda data rate altyapının tasarımı açısından farklılık göstermektedir. Simülasyon ortamı için 115200 olarak tanımlanmıştır. Gerçek ortamda ise telemetry modülü farklı bir data rate kullandığından 57600 olarak seçilmiştir.



Şekil 3: Missin Planner bağlantısı

Mission Planner üzerinden belirtilen UDP portuna bağlantı sağlandıktan sonra sanki gerçek bir drone'a bağlanmışçasına bütün bilgiler ve koordinatlar kullanıcıya sunulabilmektedir. Böylelikle hazırlanan proje çalıştırıldığında anlık durum bilgisi yazılımcılar ve simülasyon ekibi tarafından gözlemlenebilecektir.

Başarılı bir bağlantı sonucu cihaz bilgileri ve görevlerin tanımlanabileceği Şekil 4'teki ekran gözlemleyiciyi karşılamaktadır. Sol taraf drone genel bilgilerinin olduğu kısımdır.



Şekil.4. Başarılı bağlantı sonucu drone bilgileri ve harita konumu.

Harita konumu üzerinde dron simülasyon veya gerçek ortamda disarmed(emniyet) konumunda hazır olarak beklemektedir. Hazırlanan uygulamanın çalıştırılması sonucu Şekil 5'te belirtilen sahanın 4 köşesinin görev olarak tanımlanarak kalkış yapılan konuma güvenli olarak iniş yapması sağlanmaktadır.



Şekil.5. Uygulamanın çalıştırılması sonucu rota izi.

#### IV. SONUÇ

Bu çalışmada dronlar için otomatik rota tayini ve takibi önerilmektedir. Yapılan testler neticesinde dronun GPS modülü ve sensörler aracılığı ile otomatik olarak verilen görevleri altyapı üzerine geliştirilen uygulama sayesinde sorunsuz bir şekilde izlediği görülmüştür. Uçuşların başarılı olarak gerçekleştirilip gerçekleştirilmediğini izlemek için Mission Planner üzerinden anlık olarak verilerin ve logların takibi sağlanmıştır.

Bu çalışmada güvenlik en öncelikli kriter olarak tespit edilerek gerçekleşmesi muhtemel her görev öncelikle güvenlik için simülasyon ortamında gerçekleştirilmiş olup başarılı sonuçlar elde edilmiştir. Sonraki safhada uygulamanın fiziksel ortamına aktarımı sağlanmaktadır.

Seçilen mevcut donanım hali hazırda yüksek karmaşıklık arz eden görevler için yeterli işlem gücüne sahip olmayabilir. Bu amaçla karmaşık görev tanımlamaları için gelişmiş işlem gücü ve belleğe sahip donanıma ihtiyaç vardır. Bu tür karmaşık görevlerin başarılı ve güvenli bir şekilde yerine getirilebilmesi için Raspberry Pi gibi yüksek işlem hacmine sahip küçük bilgisayarlar ile çalışma yetisinin olması gerekmektedir.

#### REFERANSLAR

- [1] Li, F., Zlatanova, S., Koopman, M., Bai, X., & Diakité, A. (2018). Universal path planning for an indoor drone. *Automation in Construction*, 95, 275-283.
- [2] Yanmaz, E., Rinner B. (2018). Universal path planning for an indoor drone. *Automation in Construction*, 95, 275-283, 95, 275-283
- [3] Koubâa, A., Qureshi, B., Sriti, M., Allouch, A., Javed, Y., Alajlan, M., . . . Tovar, E. (2018). Dronemap Planner: A Service-Oriented Cloud-Based Management System for the Internet-of-Drones. *Ad Hoc Networks*.
- [4] Carlson, D. F., & Rysgaard, S. (2018). Adapting open-source drone autopilots for real-time iceberg observations. *MethodsX*, 5, 1059-1072.