GÖK-TEK SİHA

1. Uçuş Kontrol Kartı : Pixhawk The Orange Cube

Uçuş kontrol kartı (Pixhawk The Orange Cube), hava aracının otonom görevlerini yerine getirmekten sorumludur ve temel işlevleri şunlardır:

Otonom Görev Yönetimi**:** Otonom kalkış, uçuş, kilitlenme ve iniş görevlerini gerçekleştirmesini sağlar.

Sensör Veri Okuma ve Kontrol**:** Kendi içinde bulunan barometre, altimetre, pusula gibi sensörlerden ve harici olarak sisteme eklenen hava hızı sensörü gibi bileşenlerden gelen verileri okur. Bu verileri işleyerek uçağı kontrol etmek için gerekli sinyalleri gönderir

Haberleşme**:** MAVLink protokolü ve MAVROS paketi aracılığıyla görev bilgisayarı (Nvidia Jetson Xavier NX) ile iletişim kurar. Bu iletişim sayesinde kilitlenme algoritması gibi yapay zeka kararları otopilota aktarılır

Telemetri Veri İletimi**:** Hava aracının irtifa, hız, koordinat, batarya durumu ve yönlendirici açıları gibi önemli parametreleri yer istasyonuna iletir

Manuel/Otonom Geçiş Kontrolü**:** Bir röle aracılığıyla, kumandadan veya yapay zeka bilgisayarından gelen PPM verisi ile uçuş kontrolcüsü arasındaki bağlantıyı kontrol eder, böylece manuel veya otonom uçuş modları arasında geçiş yapılmasını sağlar.

Uçuş Kontrol Kartı Seçimi: Görev isterleri için yapılan çalışmalar sonucunda uçuş kontrol kartı seçerken işlemci, bellek, uyumluluk, sensörler gibi temel kriterler baz almışlar. Yapılan araştırmalar sonucu en öne çıkan 3 uçuş kontrol kartı birçok açıdan karşılaştırmışlar. Özellikle uçuş performansı bakımından ön plana çıkmayı başaran Pixhawk The Orange Cube kullanılmasına karar vermişler ama ilk olarak Pixhawk 2.4.8 seçmişler fakat maddi kaynaklarda yaşanan değişimler sebebiyle Pixhawk The Orange Cube kullanmışlar.

UÇUŞ KONTROL BILGISAYARI	NAVIO 2	PIXHAWK 2.4.8	PIXHAWK ORANGE CUB
01 Ağırlık	01 75 gr	01 38 gr	01 76 gr
02 Barometre sayısı	02 1 adet	02 1 adet	02 2 adet
03 RAM	03 2 adet	03 1 adet	03 3 adet
04 İşlemci	04 BCM283780	04 STM32F427	04 STM32H753
05 Fiyat	05 5602 TL	05 3064 TL	05 6741 TL

Kamikaze görevi: Hava aracı arama modundayken görüntüde yer hedefi tespit edilince Yapay Zeka bilgisayarı röle ile kontrolü devralır; hedef konumu sunucudan alınır ve elevator/rudder/aileron ile motor hızı PPM üzerinden uçuş kontrolcüsüne gönderilirmiş. Hedef görüntüde belirledikleri sınır büyüklüğüne ulaşana kadar bu döngü devam eder; sınırı aşınca hız azaltılır, kamikaze kilitlenme algoritması ile hedef kilitlenip irtifa hesaplanır ve dalışa geçilir. Kilitlenme başarılıysa QR kod kamerayla okunup sunucuya iletilir; görev tamamlandığında yatay seyrin ardından tırmanış komutları yine Yapay Zeka tarafından Pixhawk Orange Cube uçuş kontrolcüsüne aktarılırmış ve böylece yeni hedef konumları alınır ve arama/kilitlenme döngüsü tekrarlanırmış.

MAVLink: İHA içerisinde bulunan yapay zekâ bilgisayarı ve uçuş kontrol kartının haberleşmesini, aynı zamanda gerektiği yerde yer istasyonu ve uçuş kontrol kartının haberleşmesini sağlamak amacıyla, hava araçları için geliştirilmiş ve Pixhawk Cube ile uyumlu MAVLink protokolü kullanmışlar:
Hava aracındaki Yapay Zeka bilgisayarının bir GPIO portu, basit bir röle üzerinden Pixhawk Cube uçuş kontrolcüsüne PPM ile komut göndermek için bağlı. Diğer bir GPIO portu da bu röleye bağlı. Röle, Pixhawk'a kumanda sinyali veya Yapay Zeka'dan gelen PPM sinyalini kontrol etmek için kullanılıyor.

Yapay Zeka bilgisayarındaki kamera görüntüyü alıyor ve 5G Wifi ile yer istasyonuna gönderiyor. Yer istasyonu da bu görüntüyü hem yarışma sunucusuna iletiyor hem de analiz edip rakip İHA'yı arıyor. Bulduğu rakibin koordinat ve boyut bilgilerini 5G Wifi üzerinden tekrar hava aracına gönderiyor.

Hava aracı, gelen bu bilgileri kullanarak yapay sinir ağı modeline göre elevator, rudder, aileron açılarını ve motor hızını ayarlıyor ve PPM ile Pixhawk Cube uçuş kontrolcüsüne iletiyor. Bu işlem devam ederek hedefe kilitlenmeye çalışılıyor.

Manuel uçuşta ise röle, Yapay Zeka bilgisayarının GPIO piniyle tetiklenip PPM bağlantısı kesiliyor ve kumanda alıcısından gelen sinyal doğrudan Pixhawk uçuş kontrolcüsüne aktarılıyor.

Rakip İHA arama modunda yarışma sunucusundan alınan koordinatlar MAVLink protokolüyle telemetri modülünden Pixhawk uçuş kontrolcüsüne iletiliyor. Aynı şekilde, hava aracının irtifa, hız, koordinat, batarya durumu ve yönlendirici açıları gibi kritik bilgiler de yer istasyonuna aktarılıyor.

2.Otonom Sistemler: İHA'nın ön kısmına kamera ve görev bilgisayarı yerleştirmişler bu sayede İHA'ların otonom gerçek zamanlı görsel tespit ve takibini yüksek başarı oranı ile yapmışlar. Bu projede geliştirdikleri algoritmalar, hareket halindeki bir İHA'yı otonom olarak tespit edip takip ederek Nvidia Xavier NX üzerinde gerçek zamanlı olarak çalıştırmış.

Seçtikleri algoritmalar ve onların geliştirilmesi: Otonom sistemi 2 ayrı bölüme ayırmışlar algılama ve takip. Algılama, YOLOv3 algoritmasını ve kayan pencere (sliding window method) yöntemini temel almış. Takip ise GOTURN (Regresyon Ağlarını Kullanarak Genel Nesne İzleme) algoritmasını temel almış.Bu nesneleri yüksek hızda izlemeyi sağlıyor. Otonom izlemeyi gerçekleştirmek ve doğruluğu artırmak için GOTURN (Generic Object Tracking Using Regression Networks) ve YOLOv3 birlikte kullanılarak hibrit bir "Algılama Yoluyla Takip" kombinasyonu geliştirmişler.

2.1. Nesne Tespiti (Algılama) YOLO: Redmon ve arkadaşları, giriş olarak ham görüntü piksellerini kullanıp nesneyi çevreleyen pencerenin koordinatlarını ve sınıf olasılıklarını tahmin eden baştan-başa eğitilebilir tek aşamalı bir yapay sinir ağı tasarlamışlar. Bundan dolayı, YOLO aşırı miktarda hızlı nesne konumlandırma algoritmasıdır. Takip amacıyla YOLO kullanıldığında karşılaşılan sorunlara bakılırsa, takip görevi için gönderilen imge, hedef nesneden birden fazla içeriyorsa, algoritma doğası gereği hepsini tespit etmeye çalışacaktır bu da genel görsel takip uygulamalarının usulüne uygun degil. Ayrıca ekran kartı üzerinde çalıştığında ne kadar hızlı sayılsa da, Jetson gibi işlemciler üzerinde 100+ fps hıza ulaşabilen bir yöntem değildir ve fazla hesaplama kaynağı gerektirir. Birde takip edeceği nesneyi bir sonraki framede tahmin edememesi olasılık dahilinde olduğundan sürekli sonuç üretmesi kesin değildir.

2.2. Nesne Takibi: GOTURN ve GHT:
GOTURN (Generic Object Tracking Using Regression Networks): Çevrimiçi izleme algoritmaları, nesnenin görünümünü çalışma sırasında öğrenir. Bu yüzden, gerçek zamanlı izleyiciler genellikle derin öğrenmeye göre daha hızlı olan çevrimiçi öğrenme yöntemlerini tercih eder.

GOTURN, genellikle çok yavaş çalışan ve gerçek zamanlı uygulamalar için pratik olmayan, nesne izleme için sinir ağları kullanan önceki yöntemlere göre çok daha hızlıdır. GOTURN, çevrimiçi eğitim gerektirmeden basit bir ileri besleme ağı kullanır, nesne hareketi ile görünüm arasındaki genel ilişkiyi öğrenir ve eğitim setinde olmayan yeni tanımlanabilir nesneleri de izleyebilir. İzleme için sinir ağlarını kullanmaya yönelik önceki denemelerin yürütülmesi çok yavaştır ve gerçek zamanlı uygulamalar için pratik değildir. Buna karşılık GOTURN, çevrimiçi eğitim gerektirmeyen basit bir ileri beslemeli ağ kullanır ve izleyicimizin test süresi boyunca 100 fps'de çalışmasına olanak tanır.

Geliştirilen Hibrit Takipçi (GHT): Yukarıda bahsedilen yöntemlerin avantajları ve dezavantajları ele alındığında birbirlerinin kusurlarını örttüğü gözlemlenmiş. GOTURN yönteminin kötü durum kurtarıcısına ihtiyacı varken, YOLO algoritmasının da sadece takip edilecek hedefe odaklanmaya ve tek kartlı bilgisayarlarda daha hızlı çalışması için optimize edilmeye ihtiyacı vardır. Bu sebeple Geliştirdikleri Hibrit Takipçide (GHT), YOLO ilk framede nesnenin konumunu belirlemede ve takip edicinin başarısız olduğu durumlarda kullanılırken, GOTURN yöntemi ise diğer durumlarda nesneyi takip amaçlı kullanılmış. Bu şekilde hem YOLO konumlandırıcısını hem de GOTURN takipçisini kullanan yeni bir hibrit takip edici sistem oluşturmuşlar. Bunu gerçekleştirebilmek için öncelikle YOLO ağı, İHA imgelerinden oluşturulan veri seti ile eğitmişler.

Hız kaybı olmaması için, YOLO'nun C dilinde yazılmış orijinal sürümü derlenerek dinamik bir kütüphane haline getirilmiş ve bu kütüphanedeki fonksiyonlar Python'da kullanılmış. Bu kısımdan sonra GOTURN ve YOLO yöntemleri için sırasıyla, takipçi ve algılayıcı sözcükleri kullanılmış. Sonuç olarak, Geliştirilen Hibrit Yöntem GOTURN'den hız, tek hedefe kilitlenme ve sürekli sonuç üretme avantajlarını alırken, YOLO'yu da takipçide oluşacak kötü durumları kurtarması ve en-boy oranını dinamik olarak güncellemesi için kullanılmış.

Videodan nesne takibi için OTB (Object tracking benchmark), VOT (Visual object tracking) gibi birçok veri seti bulunmasına rağmen sırf İHA lardan oluşan veri seti bulunmamaktadır. Bu problem için yaklaşık 7500 imge etiketlendirilerek veri seti oluşturulmuş. Bu veri setinin yaklaşık 2000 imgelik kısmı YOLOV3'ün eğitimi için, 15 videodan oluşan yaklaşık 5500 imgelik kısmı ise test için ayrılmış.

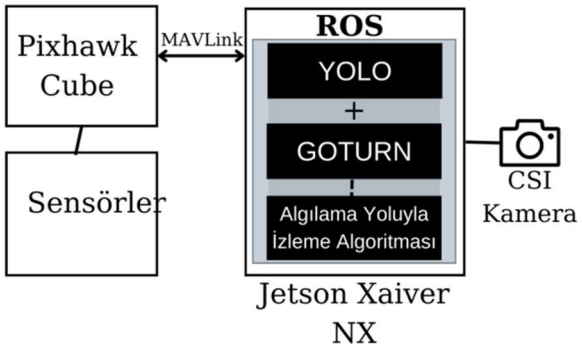
Geliştirdikleri Yöntem ile Literatürdeki Yöntemlerin İHA Veri Seti Üzerinde Performanslarının Karşılaştırılması:

Yöntem	Başarı	Hassasiyet	Hız (fps)
GHT	0.561	0.773	53.5
GHT-Tiny	0.524	0.737	69.2
YOLOv3	0.485	0.653	16.5
YOLOv3Tiny	0.461	0.630	47.1
CSRT	0.232	0.402	111
GOTURN	0.205	0.265	20.1
MIL	0.215	0.379	12.3
KCF	0.126	0.196	115
BACF	0.213	0.339	25.8

Bu projede, İHA'ları tespit ve takip edebilen, YOLOv3 ve GOTURN tabanlı hibrit bir yöntem geliştirilmiş ve bu yöntem, gerçek zamanlı çalışan diğer yöntemlerle karşılaştırılmış. Eğitim ve test için İHA içeren özel bir veri seti oluşturulmuş, ayrıca hız açısından YOLOv3-Tiny modeli de değerlendirilmiş. Geliştirilen Hibrit Yöntem (GHT), diğer yöntemlere göre daha yüksek başarı sağlasa da GHT-Tiny'ye kıyasla daha yavaş. Yöntem, yalnızca İHA takibi için özelleştirilmiş olduğundan genel nesne takibi yapan yöntemlerden bu yönüyle ayrılıyor.

3.Otonom Görev Algoritmaları

3.1 Otonom Kilitlenme Algoritması: Kamera görüntüsünde hedef bir İHA tespit edilmesi durumunda, hedefin kilitlenme kamerasının görüntüsünün tam ortasına gelmesi için geliştirilen hibrit takipçi (GHT) algoritması yardımıyla gerekli olan İHA yönelimi hesaplanır. Bu hedef yönelime gelinerek aracın görüntünün tam ortasında olması sağlanır ve İHA, hedef İHA'ya otonom olarak kilitlenme görevini gerçekleştirir. Bu hedef yönelimlerin otopilota aktarılması için Robot Operating System (ROS) yazılımı kullanılıyormuş. ROS ile birlikte kullanılan MAVROS paketi sayesinde otopilot ile görev bilgisayarı arasındaki iletişim sanal bir ağ yardımıyla kuruluyor ve bu iletişim kurulduktan sonra tüm veri alışverişi daha kolay.



İHA, takip sırasında hedefi belirli bir kilitlenme alanına getirmeye çalışıyor. Bu kilitlenme durumu, görev bilgisayarında görüntü işleme ile kontrol ediliyor ve yer istasyonuna aktarılıyor. Eğer hedef 4 saniye boyunca bu alanda kalırsa, sistem bunu başarılı bir kilitlenme olarak kabul ediyor ve başka bir hedef aramaya başlıyor. Bu döngü ya görev tamamlanana ya da 15 dakikalık süre dolana kadar devam ediyor. Görev sona erdiğinde ise İHA, otonom bir şekilde inişini gerçekleştiriyor

3.2 Otonom Yönelim Algoritması: İHA'lar, Nvidia Jetson Xavier NX görev bilgisayarıyla yarışma sunucusundan gelen telemetri ve GPS verilerini işleyerek rakip İHA'ların gerçek konumlarını tahmin ediyor. Görsel algılama yapılamadığı durumlarda, ekip tarafından geliştirilen 3 boyutlu haritalama algoritması sayesinde konum verileriyle yönlendirme yapılıyor ve hedef tespiti kolaylaştırılıyor. Ayrıca, alınan GPS verileri Nvidia Jetson Xavier NX üzerinde Kartezyen koordinatlara dönüştürülerek spektral kümeleme uygulanıyor. Bu sayede, en fazla İHA'nın bulunduğu kümenin merkezi uçuş rotası olarak belirleniyor. Kümedeki en yakın İHA hedef alınıyor, takip tamamlandığında bir sonraki en yakın İHA'ya yönelerek döngü devam ettiriliyor. Veri gecikmesi veya güvensizlik durumunda ise, yardımcı pilot manuel müdahalede bulunarak İHA'yı hedefi kameraya alabilecek uygun pozisyona getiriyor.

3.3 Otonom Kaçış Algoritması: İHA'nın rakip araçlar tarafından takip edilip edilmediğini belirlemek ve bu duruma karşılık otonom kaçış manevrası gerçekleştirmek amacıyla bir takip tespiti ve kaçış algoritması geliştirilmiştir. Bu algoritma, yarışma sunucusundan sağlanan rakip İHA'ların telemetri verilerine dayalı olarak çalışmaktadır.

1. Takip Tespit Mekanizması

Aday Takipçi Tespiti: Gök-Tek İha'nın bulunduğu konumu merkez alan ve yarıçapı 100 metre olan bir küre tanımlanır. Bu küreye giren diğer İHA'lar "aday takipçi" olarak değerlendirilir.

Muhtemel Takipçi Tespiti: Aday takipçilerin Gök-Tek İha'yla olan öklit mesafesi sürekli izlenir. Bu mesafe 50 metrenin altına düştüğünde, ilgili İHA "muhtemel takipçi" olarak işaretlenir.

Takipçi Onayı: Eğer mesafe azalmaya devam eder ve 20 metrenin altına düşerse, bu İHA "takipçi" olarak tanımlanır ve kaçış algoritması devreye alınır.

2. Kaçış Algoritması

Yoğunluk Tabanlı Kaçış: Takip tespiti sonrası görev bilgisayarı (Nvidia Jetson Xavier NX) tarafından yarışma sunucusundan alınan konum verileri analiz edilir. Spektral kümeleme algoritması ile en fazla rakip İHA'nın bulunduğu bölge belirlenir. Bu bölgeye doğru rota oluşturularak kaçış manevrası gerçekleştirilir.

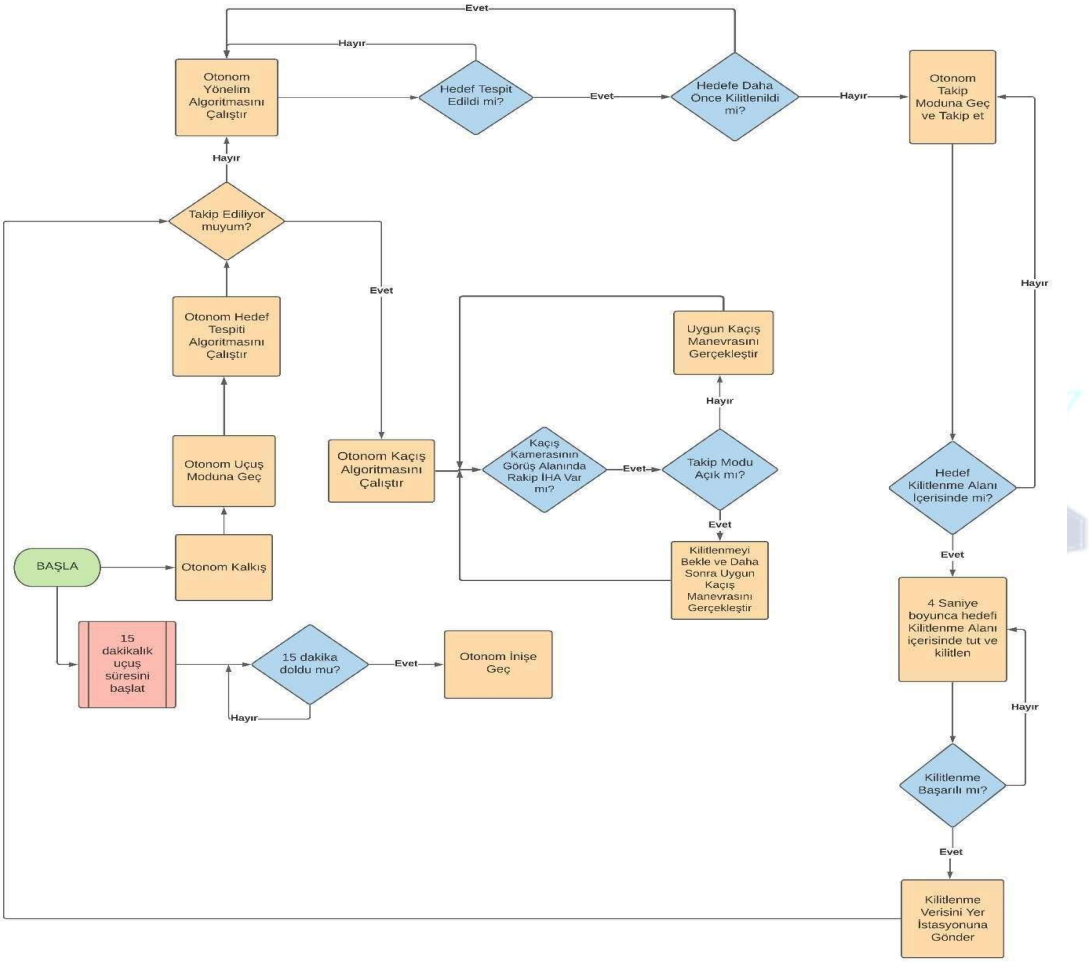
Amaç: Takipçi İHA'nın dikkatinin dağıtılması ve başka İHA'lara kilitlenmesinin sağlanması.

3. Manuel Müdahale Durumu

Otonom Tespit Başarısızlığı: Eğer sistem takip tespiti gerçekleştiremezse ve yer istasyonu veya yardımcı pilot tarafından İHA'nın takip edildiği fark edilirse:
Yer istasyonu, acil durum uçuş planına geçiş yapar veya yardımcı pilot manuel kontrol ile İHA'yı devralır.

Bu yapı sayesinde, İHA hem otonom hem de gerektiğinde manuel olarak takipten kaçabilecek şekilde tasarlanmış, yarışma şartlarında güvenli ve esnek bir uçuş stratejisi sağlanmıştır.

Otonom Görev Akış Şeması:



4. Kullanıcı Arayüz Tasarımı: Yer kontrol istasyonu, İHA ile kullanıcı arasında haberleşmeyi sağlamak, uçuş verilerini takip etmek ve aracı hazır komutlarla kolayca yönlendirmek amacıyla kullanılmaktadır. Savaşan İHA yarışması kapsamında; hız, dönüş açısı, yükseklik, görev modu gibi uçuş verilerinin izlenmesi, aracın konumunun gerçek zamanlı görüntülenmesi, RTL yüksekliği ve PID gibi ayarların yapılabilmesi gibi gereksinimleri karşılayacak bir arayüz tasarımı hedeflenmiş. Bu doğrultuda, açık kaynaklı ve kullanıcı deneyimi yüksek olan Mission Planner ile yerli geliştirilen Gök-Tek yazılımları olmak üzere iki farklı yer kontrol istasyonu tercih etmişler.

4.1 Mission Planner Yer Kontrol Yazılımı: Kullanılacak olan otopilot kartı (Pixhawk The Cube Orange) ve otopilot yazılımı (Ardupilot) ile uyumlu çalışan Mission Planner arayüzünün kullanılmasında karar verilmiş. Mission Planner'ın Yer Kontrol İstasyonu arayüz programı olarak kullanılmasındaki ana sebepler şu şekilde sıralanabilir:

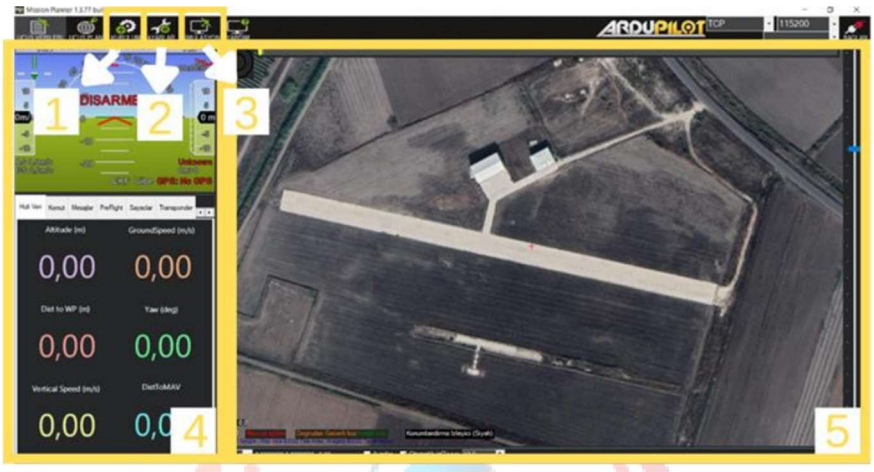
1-Uçağa ait birçok parametrenin ayarlanması ile gerçek uçuşlara yakın seviyede hazırlanan simülasyon ortamı (Software In The Loop (SITL)) üzerinde uçuş öncesi testler yapılabilmek.

2. Anlık uçuş değerleri (altitude, attitude (roll, yaw, pitch), speed (air, ground) vs.) gözlemlenebilir ve uçağa anlık komutlar iletilebilir.

3. Uçuş sonrasında uçuş kayıtlarının tutulduğu log dosyalarının yine Mission Planner tarafından sunulan kapsamlı arayüz yardımıyla ayrıntılı analizleri yapılabilmekte, araca ait aviyonik alt sistemlerin kalibrasyonları kolayca ayarlanabilmekte ve yine araca ait parametreler isterlere göre yeniden ayarlanabilir.

4. Birçok kişi ve takım tarafından kullanıldığı için güçlü bir komünitesi var, düzenli ve ayrıntılı hazırlanmış dokümantasyona sahip.

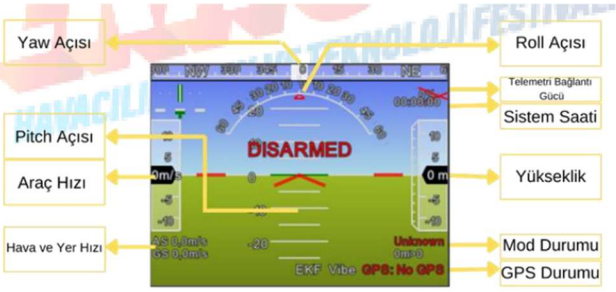
5. Açık kaynak kodlu olduğu için kendi isterlerimize göre yeni özellikler eklenebilmektedir



Mission Planner Arayüzü

1-) Konfigürasyon Ekranları 2-) Kalibrasyon Ekranı 3-)Simülasyon 4-) Takip Paneli 5-) Harita

4.2 Takip Paneli: Mission Planner arayüzünde olan takip panelinin kullanımı kolay. Takip Panelinde uçağın genel uçuş bilgileri, ARM/DISARM işlemleri gibi değişimlerin yapılabileceği iki farklı bölümden oluşutor. Ve uçak ile alakalı pekçok bilgi yer almaktadır. (Roll, Pitch ve yaw açıları, yükseklik, hız, mod türü, uçuş saati, telemetri bağlantısı, batarya yüzdesi, GPS durumu ve benzeri...

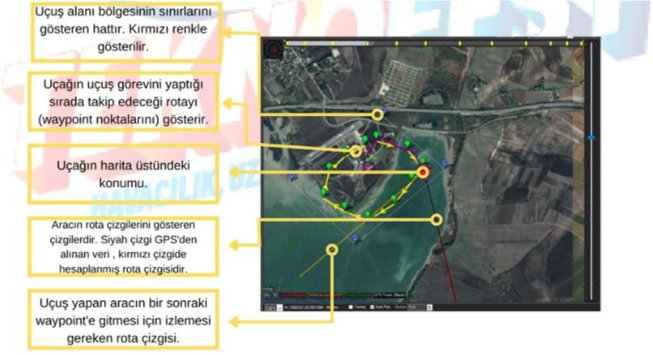


Takip Paneli

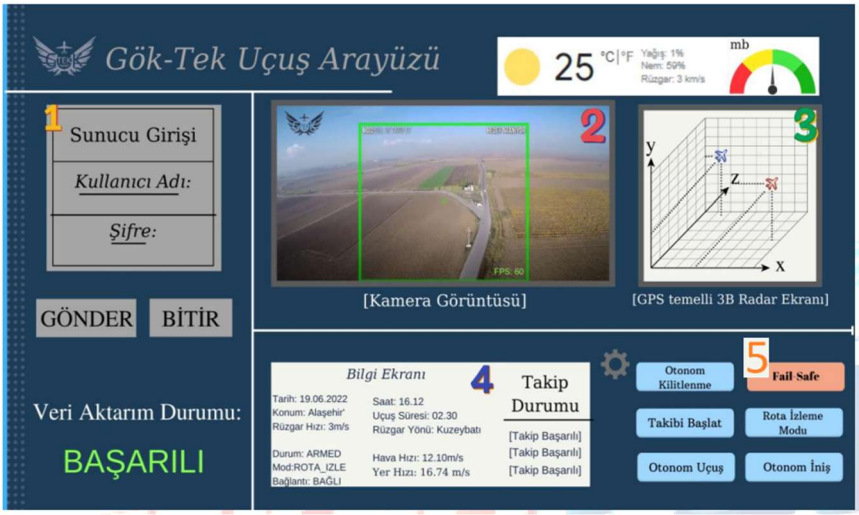
Uçuş Takip panelinin bir diğer elemanı ise uçuş esnasında yer kontrol istasyonu sorumlusunun; uçuş öncesinde genel sistem parametrelerini incelemesini, uçağın uçuştan önce genel uçuş kalibrasyonlarının yapılmasını ve ARM/DISARM işlemlerinin yapılmasını sağlayan birçok fonksiyonel menüden oluşur.

Komut Paneli

4.3 Uçuş Harita: Mission Planner arayüzünün önemli araçlarından biri de verilerini Google Map Data'dan çeken harita bölümü. Bu bölüm yer kontrol istasyon sorumlusunun en çok takip etmesi gereken bölümlerden biri. Aracın anlık olarak konumunun takip edilebildiği bu kısımda dönüş yarıçapı, rota hattı, waypoint noktaları, uçuş alanının sınırları gibi verilerde yer alır.



4.4 Görev Sunucu Arayüzü: GÖK-TEK takımı tarafından Savaşan İHA yarışması görevlerine özel olarak geliştirilen ikinci arayüz, İHA'dan gelen görüntülerin anlık izlenebildiği, kitlenme verilerinin görüntülenebildiği, sunucuya veri aktarımı yapılabilen ve sunucudan gelen verilerin takip edilebildiği özgün bir sistem olarak tasarlanmıştır. Görev ve konuma özel ihtiyaçlara yanıt verebilmesi için hazır sistemler tercih edilmemiş, bunun yerine Python'un Tkinter kütüphanesi kullanılarak tamamen özgün ve entegre edilebilir bir arayüz geliştirilmiştir. Bu arayüz; giriş ekranı, uçuş haritası, uçuş görüntü ekranı ve uçuş verileri ekranı olmak üzere toplam dört bölümden oluşmaktadır.



1-) Sunucu Giriş Ekranı 2-) Görüntü Ekranı 3-)Rakip İha Görüntüleme Sistemi
4-) Veri Ekranı 5-) Elle Müdahale Ekranı