

Multi Rotor Robotlar İçin Oto Pilot Yer Kontrol Birimi

Auto Pilot Ground Control Unit For Multi Rotor Robots

Haluk DİLMEN¹, Yunus SANTUR¹, Hakan BAYSAL², M. Fatih TALU¹

¹ Bilgisayar Bölümü Mühendislik Fakültesi
İnönü Üniversitesi

hdilmen@firat.edu.tr, ysantur@firat.edu.tr, mfatih.talu@inonu.edu.tr

² Bilgisayar Bölümü Mühendislik Fakültesi
Fırat Üniversitesi
hbaysal@firat.edu.tr

Özet

Elektronik entegre ve cihaz fiyatlarının son yıllarda giderek yaygınlaşması ve maliyetlerinin düşmesi kopter olarak bilinen dikey olarak iniş ve kalkış yapabilen sistemlerin daha sıklıkla kullanılır olmasına neden olmuştur. Bu makalede kopterinin iniş, kalkış, hareket ve otonom nesne takibi fonksiyonlarını barındıran Oto Pilot olarak adlandırılan yazılım modülünün tanıtılması hedeflenecektir. Oto Pilot modülü Python yazılım dili ve Robot Operating System (ROS) kullanılarak geliştirilmiştir. Oto Pilot Yazılımında otonom uçuş iki bölüm altında incelenebilir. Bunlardan ilki görüntü temelli olup nesne takibi sırasında gerçekleştirilen otonom uçuş bir diğeri ise GPS verilerini kullanarak gerçekleştirilen otonom uçuş. Geliştirilen yazılım ile kopterinin uçuşu esnasında kullanıcılara kontrol fonksiyonlarının sunulmasının yanında aynı zamanda kullanıcıların izlemek ve değerlendirmek istedikleri önemli bilgilere erişimleri amaçlanmıştır. Kopterinin hareketi sırasında kamerasından elde edilen görüntüler Oto Pilot'a aktarılarak kullanıcıların uçuşu görüntülemesi imkânı sunulmuştur.

Abstract

The price and cost decrease on integrated circuit and devices in recent years made copters more commonly used known for the ability to hover and take off vertically. In this article it is aimed to introduce autopilot software module that harbour copter functions to take off, land, move and autonomously track object. Autopilot module has been developed using Python programming language and the Robot Operating System (ROS). Autonomous flight can be examined in two parts in the Autopilot Software. The first autonomous flight based on image-based object tracking other autonomous flight performed by using GPS data. Besides offering users control functions Auto Pilot allows users to monitor and evaluate important information they want to monitor and evaluate. Kopter video images obtained from it's camera are transmitted to autopilot flight users given the opportunity to view the flight.

1. Giriş

İnsansız Hava Aracı (Unmanned Aerial Vehicle-UAV), uzaktan kontrol veya kendi başına hareket edebilen birimler olarak adlandırılmaktadır [1]. Bu tür hava araçları en yaygın olarak askeri uygulamalarda kullanılmaktadır. Birkaç yüz kilogram ağırlığında yük taşıyan sistemler fiyat olarak yüksek maliyetlerde inşa edilebilmesine karşın araştırmacılar ve sivil organizasyonlar daha çok mini boyutlardaki UAV'ler ile çalışma yolunu seçmişlerdir. Genel olarak UAV'ler uzaktan kumanda ile kontrol edilmelerine rağmen son zamanlarda otonom kontrol sistemi kullanan yapılar giderek daha fazla kullanılmaya başlanmıştır. UAV'ler askeri uygulamalarla birlikte son zamanlarda itfaiye, boru hatlarının gözetimi, askeri olmayan güvenlik çalışmaları, ulaşım gibi diğer sivil alanlarda da kullanılmaya başlanmıştır.

Gün geçtikçe sivil ve askeri alanlarda gelişen kullanım alanları UAV'leri kritik işlerde önemli kılan sebepler arasındadır. Kendi başına uçabilen otonom yapılar ve kolaylıkla kullanılan sistemler olması bakımından UAV'ler her geçen gün kritik uygulama ve işlerin önemli bir parçası olma yolunda hızla ilerlemektedirler. Bununla beraber geliştirilen çoğu uygulama daha çok yerden kontrol edilen ve belirli bir mesafede kullanılan sistemler olarak tasarlanmaktadır. Kullanılan araçlar ve bunlarla bağlantı, karmaşık görev ve uygulamalar için en hayati nokta olarak karşımıza çıkmaktadır. UAV ile kurulacak bağlantıda uplink kritik kontrol bilgilerinin iletilmesi ve downlink ise telemetri bilgilerinin operatör ve kullanıcılara aktarılması amacıyla kullanılmaktadır. UAV ile yer istasyonu arasında kurulacak bağlantının ve iletilecek verilerin miktarı UAV'nin otonom veya yarı otonom uçuş yapısı ile yakından ilgilidir. Tam otonom sistemlerde UAV ile bağlantı ve iletilen komutlar minimum seviyede gerçekleştiğinden bağlantı gereksinimi yarı otonom sistemlere göre daha

düşüktür. Tam otonom sistemlerde iletilmesi gereken bilgiler genel olarak belirlenen noktaya uçuş, uçuş başlama, inme, binayı gözetleme veya belirlenen kişiyi takip etme gibi sınırlı sayıda komuttan oluşmaktadır. Aynı şekilde UAV'den yer kontrol istasyonuna iletilen bilgiler de görev yapısına göre değişmektedir. Genelde UAV'den yer kontrol istasyonuna UAV'nin pozisyonu, yüksekliği gibi telemetri bilgileri iletilmektedir.

UAV'ler arasında Dört Rotor (Quadrotor) olarak adlandırılan yapı en fazla yaygın olanıdır. Quadrotor, UAV bünyesinde bulunan dört ayrı rotor sisteminden ismini alır. Quadrotorlarla birlikte ayrıca 6 ve 8 rotorlu yapılarda bulunmaktadır. UAV'ler rotor sayısı ile doğru orantılı olarak daha fazla yük ve sensör taşıma imkânına sahip olurlar. Örneğin sekiz rotorlu bir yapı yaklaşık 1 kilograma yakın ekstra yük taşıma seçeneğini sunabilmektedir. Quadrotorların ve diğerlerinin yaygın olmasındaki başlıca sebepler basit mekanik yapısı ve ileri kontrol algoritmalarının uygulanabilirliğine izin veriyor olmasıdır [2]. Quadrotorları ticari olarak sunan Avrupa ve Amerika'da çeşitli kurumlar bulunmasına karşın ana pazarları araştırma geliştirme merkezleri olarak gösterilebilir [3-4].

Genel olarak uçaklarda jiroskop ve ivmeölçerler navigasyon için vazgeçilmez sensörlerdir. Jiroskop uçağın hangi ekseninde (pitch, yaw, roll) ne kadar döndüğünü bulmamıza yardımcı olurken ivmeölçerler de hangi eksen yönünde ne kadar gittiğimizi bize gösterir. Bu sensörler eskiden çok büyük boyutlarda olmalarını karşın, gelişen MEMS (Mikro Elektro Mekanik Sistemler) teknoloji sayesinde bir entegre boyutuna kadar küçüldüler. Navigasyon için çok kullanılan bir diğer sensör ise elektronik pusula ya da manyetik yön sensörüdür. Bu sensör sayesinde platformumuzun ne yöne doğru gittiği çok hassas bir şekilde ölçülebilir. Uçağın hız ve yükseklik bilgisi de genellikle basınç sensörleri ile ölçülür ve hesaplanır. Bütün bu sensörler doğru ölçüm için çok iyi ölçümleme (kalibrasyon) isterler.

Uçuş koşullarındaki istikrarlı olmayan durumlardan dolayı bu tür araçların kontrolü oldukça zordur. UAV'lerin görsel olarak kontrolleri konusunda birçok farklı çalışma yapılmıştır. Genel yapı UAV üzerine monte edilmiş bir kamera yardımıyla ortamda tespit edilen nesne ve objelere ait görüntülerin direk olarak sunulması esasına dayanır [5-7]. Bir diğer yöntem de, [8]'da melez(hibrit) kamera yapısıdır. Bir kamera araç üzerinde kullanılırken bir diğeri ise yerde kullanılarak aracın uçuş güzergâhını izleme ve varış noktasındaki durum ve koşullar hakkında bilgi almak amaçlanır. Yerden kontrol edilen araçlarda uçuş esnasında deneyimli operatörlere ihtiyaç vardır. Operatörler bu cihazları uçuş sırasında farklı görevleri aynı anda izleme ve değerlendirme durumuyla karşı karşıya kalırlar. Bunun için üniversiteler, şirketler, ve bu işi hobi olarak yapanlar, UAV'leri elektronik olarak stabilize edecek atalet ölçüm sistemlerini (IMU) geliştirmişlerdir. Bu sistemler sayesinde

araçlar konumlarını GPS sistemini kullanarak yere göre belirlerler [9].

Üç eksenli Kartezyen koordinat sistemi için entegre edilmiş açışal hız sensörleri ve doğrusal ivmeölçerler stabil bir uçuşu kontrol etmek için bir mikroişlemci tarafından sürekli örneklendirilip işlenmektedir. UAV'ler çeşitli teknik sistemler örneğin GPS, ultrasonic algılama, açışal hız sensörleri ve doğrusal ivmeölçerler ile 3 boyutlu hareket takibi gereksinimlerini yerine getirebilirler [4].

Bu entegre sistemlerin ve sensörlerin kullanımına karşın quadrotorların kontrolü hala en zor konulardan biri olmakla beraber birçok araştırma merkezinde konuyla ilgili araştırma geliştirme faaliyetleri devam etmektedir [4, 10]. Araştırma faaliyetlerinin büyük çoğunluğu seyir sırasında kontrol sorunlarına değinmektedirler. Örnek olarak, görüntü temelli seyir, eş zamanlı yer ve haritalama uygulamaları gösterilebilir. Bahsedilen bu uygulamalar doğrusal veya doğrusal olmayan karmaşık kontrol algoritmaları kullanırlar.

UAV'lerin uçuşlarındaki nihai hedef, tamamen kendi halinde seyir gerçekleştirebilen sistemlere erişmektir [11]. Otonom kendi başına uçan helikopter tasarımı ortam değişkenlerinin ve sensörlerdeki verilerin çok hızlı değişimi dolayısı ile oldukça zor bir görevdir. Özerk navigasyon sistemi kullanarak hareket edebilen sistemlerin en sık kullandığı form GPS kullanarak gerçekleştirilen uçuşlardır [10].

GPS sistemi kullanılabilir alanlarda uçuş süresinde güvenilir ve doğru bilgileri sunarak UAV'lerin otonom uçuş işleminin gerçekleştirilmesinde yardımcı olmaktadır. Sonuç olarak GPS ve Atalet Navigasyon Sistemi(INS) in entegrasyonu ile ilgili pek çok çalışma yapılmıştır.

Bu konuda dikkat edilmesi gereken noktalardan bir diğeri ise teknolojik yenilikler sensör fiyatlarında maliyetleri aşağı çekerken sensörlerdeki kalite ve doğruluk oranlarını da düşürmüştür. Sistem tasarlanırken birlikte çalışacak sensörlerin entegrasyonu ve birbiriyle uyumlu çalışması hayati önem taşımaktadır [12]. Yüksek manevra gerektiren uçuş sırasında GPS sensörlerindeki hata, IMU ve diğer sensörleri örneğin barometreyi kritik duruma çekmektedir. Uçuş sırasında sensör ve sistemlerin birisinde oluşabilecek hata durumunda UAV'nin diğer sensör ve algoritmaların yardımıyla fonksiyonlarını devam ettirebilmesi büyük önem taşımaktadır.

GPS ve IMU entegrasyonu gevşek veya sıkı olarak birbirlerine bağıllık derecesine göre değişmektedir. Bu durum GPS takip bant genişliğini düşürme veya cihaz arama ve takip sistemlerindeki en uygun performansı yakalamada yardımcı olur [12]. GPS ve IMU'nun gevşek bağıllı yapıya sahip olmasının bir faydası, her iki sistemin birbirinden ayrı yapı olarak değerlendirilmesinin mümkün hale gelmesidir. Diğer farklı yapılardaki bağımsız

işlevselliklerin bütünleşme amaçlı kullanılabilmesine olanak sağlar.

1.2 ROS (Robot Operating System)

ROS (Robot İşletim Sistemi) yazılım geliştiricilere robot uygulamaları oluşturmak yardımcı olmak için kütüphaneleri ve araçları sağlar. ROS açık kaynak BSD lisansı altında lisanslanmıştır. ROS geliştiricilere aygıt sürücülerini, donanım soyutlama, mesaj aktarımı, paket yönetimi, görselleştiriciler ve kütüphaneler olmak üzere pek çok imkan sağlar. ROS sistem olarak iki kısımdan oluşur. İlki var olan işletim sistemine yüklenen ROS kütüphanelerini içeren yazılım paketidir. Bu kısım sistemin temelini oluşturur ve günümüzde stabil olarak Ubuntu işletim sistemini desteklemektedir, bir diğer kısım ise “ros-pkg” adı verilen ROS paketleri, ana sistemin üstüne yüklenen ve robotların yapmasını istediğiniz işleri içeren uygulamaları içeren kısımdır. Bu kısım genellikle kullanıcılar tarafından oluşturulan ROS uygulamalarıdır. Bu uygulamalar sayesinde robotlara obje ve yüz tanıma, derinlik algılama, hareket izleme ve iki boyutlu görüntüden 3 boyutlu görüntü elde etme gibi insansı özellikler kazandırılabilir.

ROS üniversite laboratuvarlarından çıkma bir teknoloji olduğu için daha çok robot araştırmalarında kullanılıyor olsa da bugün amatör ruhlu hobi severler arasında da yaygın olarak kullanılmaya başlanmıştır. Açık-kaynak kodlu olması ve resmi sitesinde iyi dokümanite edilmiş kurulum ve eğitim içeriğinden dolayı ilgisi olan herkesin kendi bilgisayarına kurabileceği ve deneyebileceği bir sistem haline gelmeye başlayan ROS gün geçtikçe daha fazla uygulama ile daha fazla robot için tercih edilir hale gelmektedir [13].

ROS’un bir diğer büyük avantajı ise robot sektöründe eksikliği çok büyük olan simülasyon sorununa güçlü bir çözüm getirmesidir. Yine Ubuntu üzerinde çalışan Gazebo adlı robot simülasyon programı ile entegre olabilen ROS sayesinde pahalı robotları satın almadan bilgisayarlarda uygulamalar yazıp oluşturulan robotları sanal ortamda test etme imkanı sunulmaktadır.

ROS’un rostopic adında komut satır araçları bulunmaktadır. ROS tüm yayınlar ve modüller için bir topic barındırır. Rostopic bu topiclerin bant genişliğini, topic ile ilgili yayınlanan mesajları görüntülemeye yarar. Rostopic diğer ROS araçları gibi mesajın içeriğini görüntülemek için YAML sözdizimini kullanır. Roslisp, ROS düğümleri yazmak için bir istemci kütüphanesidir. Bu kütüphane çalışan ROS sisteminin hata ayıklama işlemlerine yardımcı olur. Roslaunch, ROS paketlerini çalıştırabilmek için kullanılan XML formatındaki araçtır.

2. Önerilen Yöntem

Yapılan bu çalışmada kopter sensör ve kamera verilerini kullanarak Oto Pilot olarak adlandırılan kopterin yerden kontrol ve izleme amaçlı yazılımının gerçekleştirilmesi sağlanmıştır. Gerçekleştirilen yazılım sayesinde kopterin kalkış iniş ve hareket komutlarıyla yönlendirilmesi gerçekleştirilmektedir. Ayrıca kopterin hareketi sırasında gerçek zamanlı görüntüler Oto Pilot yazılım modülüne aktararak kullanıcıların bilgisine sunulmuştur. Çalışma sonuçlarını izleme ve değerlendirme amaçlı kopter olarak AR. Drone 2.0 kullanılmıştır. AR.Drone 2.0 seçilmesinin nedenlerinden biri çektiği görüntüyü online olarak aktarabilen

kameraya sahip olmasıdır. Ayrıca basit yapısı hem iç hem de dış mekân uçuşuna uygun olması bir diğer tercih sebebidir. Gerekli olan yazılımların geliştirilmesi ve testlerin yapılması amacıyla Ubuntu 12.04 üzerine ROS işletim sistemi ve kopter ile haberleşmeyi gerçekleştirecek sürücüler kurulup, bu sayede koptere bilgi aktarma ve kopterden bilgi alma işlemleri gerçekleştirilmiştir.

Özellik çıkarma ve nesne takibi işlemlerini gerçekleştirmek için OpenTLD adlı yazılım kullanılmıştır. OpenTLD görüntü üzerinde nesne takibini belirli argümanlara göre (yüz takibi, görüntü içindeki insan algılama, nesne algılama gibi...) takip algoritmasını işleyen açık kaynaklı bir yazılımdır [14]. OpenTLD’nin ana görevi zamanla değişen video imgeleri içerisinde nesnenin yörüngesini nesnenin pozisyonunu bularak elde etmektir. Nesnenin kesintisiz uzun süreli takibi nesne hareketlerindeki ani değişim, nesnenin gizlenmesi, ışık ve renk yoğunluklarındaki ani değişim ve görüntüdeki değişimler dolayısı ile zor bir görevdir. Nesne takibi $I1, I2, \dots, In$ sıralı imge dizisi içindeki Ik imgesi içinde Xk durum tahmininin elde edilmesi olarak adlandırılabilir. OpenTLD nesne yörüngesi etrafında imge yamalarını kullanarak nesne algılayıcının eğitilmesi esasına dayanmaktadır [14]. OpenTLD Takip, Öğrenme ve Algılama olmak üzere üç ana başlıktan oluşur. Bölüm 2.1, 2.2 ve 2.3’te bu başlıklara ait bilgiler verilecektir.

2.1 Takip

Rekursif nesne takibi nesnenin Xt şimdiki durum tahmininin, nesnenin önceki $Xt-1$ durum tahmini ve ölçüm bilgileri ($Z1, Z2, Z3, \dots, Zn$) göz önüne alınarak elde edilmesi işlemidir. Dolayısı ile nesnenin şu andaki durumu önceki durumu ile irtibatlıdır. En genel ifade ile takip işlemi nesne içerisine yayılmış noktaların Lukas Kanade optik akış yöntemi ile bir sonraki imgede tahmini üzerine kurulmuştur. Lukas Kanade optik akış yöntemi üç adet varsayımı temel alır.

Birincisi parlaklığın sürekliliği:

Parlaklığın sürekliliği

$$I(x) = J(X + d) \quad (2.1.1)$$

(2.1.1)’de gösterildiği hali ile formüle edilebilir.

I görüntüsündeki piksel konumu J görüntüsünde yerini değiştirmekle beraber parlaklığını korur. Buradaki d konumlandırma vektörü olarak değerlendirilir.

İkincisi varsayım ise geçici sabitlik:

Bu varsayım konumlandırma vektörünün küçük olduğunu ve (2.1.2)’de gösterildiği hali ile hesaplanabileceğini varsayar.

$$J(x) \approx I(X) + I'(X)d. \quad (2.1.2)$$

$I'(X)$ I ya ait X konumundaki türevi gösterir. Bu durumda d

$$d \approx \frac{J(x) - I(X)}{I'(X)}. \quad (2.1.3)$$

ifadesi yardımıyla elde edilir. 2.1.3 için çözüm uzayı nokta yerine bir çizgidir.

Üçüncü varsayım uzaysal tutarlılık olarak adlandırılan bu varsayım her bir çerçevedeki pikselin çevresindeki piksellerin bir tutarlılık ve uyum içinde hareket ettiğini varsayar. Bahsedilen tahminlerin ilişkilendirilmesi ile d ’nin bulunması en küçük kareler yöntemi olarak formüle edilip 2.1.4’te belirtilen şekilde ifade edilebilir.

$$\sum_{(x,y) \in W} (J(X) - I(X) - I'(X)d)^2. \quad (2.1.4)$$

[20]’de kapalı formda 2.1.4 aşağıdaki gibi gösterilebilir.

$$Gd = e. \quad (2.1.5)$$

Detaylı açıklamalar için [14] incelenebilir.

2.2 Algılama

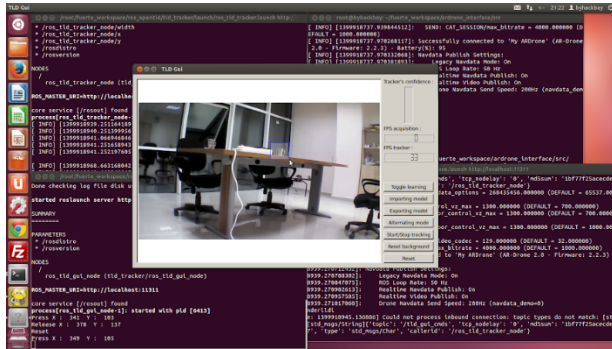
Nesne algılayıcısı ise nesnenin giriş görüntüsü içerisinde konumunun belirlenmesi işlemine kullanılır. Nesnenin algılanması olayında algoritmalar uygunluk seviyesine ulaşmıştır denebilir. Önemli asıl nokta görüntünün geometrik şeklinin ve özniteliklerinin belirlenmesi olayında karşımıza çıkmaktadır. Nesnenin tespiti konusunda kayan pencereler yöntemi kullanılmaktadır. İzlenen nesnenin kaybolması veya saklanması gibi durumlarda takip işlemi hata verir. Bu durumda nesnenin yeniden bulunup takip ediciye verilmesi gerekmektedir. Algılayıcı nesnenin o ana kadar olan görüntüleriyle eğitildiğinden kayan pencereler yöntemi ile nesnenin konumunu saptamada kullanılır. Sonrasında nesne hazırlanarak takip ediciye tekrardan verilir. Böylece nesne takip olayı kesintisiz uzun süreli olarak devam ettirilmiş olur.

2.3 Öğrenme

Nesnenin yörüngesi çevresindeki pozitif ve negatif yamaları referans olarak sınıflandırıcının eğitilmesi ve güncel tutulması esasına dayanır. Zaman geçtikçe öğrenme algoritması sayesinde sınıflandırıcı hatalarından öğrenerek takip ve algılama işleminde daha başarılı sonuçlar elde edecek hale getirilir.

OpenTLD direkt olarak video cihazı ile çalıştığından bunun Ros'a uyarlanması gerekir. Bu amaçla Ros-OpenTLD kullanılmıştır. Ros-OpenTLD, OpenTLD kütüphanelerini kullanan fakat video cihazı yerine Ros üzerinden görüntü çerçevelerini alan bir yazılım olarak bilinir.

Tüm yapıları tek bir yerde toplamak amacıyla python kullanılarak bir ara yüz hazırlanarak ROS, Ardrone Autonomy ve ROS-OPENTLD'nin tek bir çatıda toplanması gerçekleştirilmiştir. Şekil 1'de ROS, Ardrone Autonomy ve ROS-OPENTLD'nin entegre edilmesi ile kopter kamerasından elde edilen görüntü gösterilmiştir.



Şekil 1. Ros – OpenTLD ile Ardrone kamerasından okunan görüntü çerçeveleri ile nesne takibi

3. Uygulamaya Ait Sonuçlar

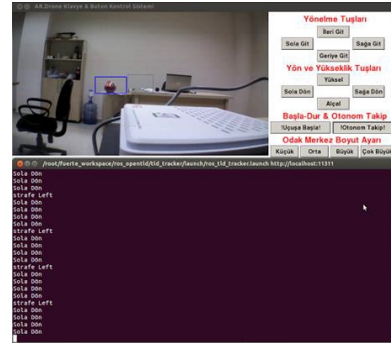
Proje AR.Drone 2.0 üzerinde tasarlanmış olup ancak istenilen tüm multi-copterler üzerine uygulanabilir niteliktedir.

Şekil 2'de manuel ve otonom uçuşların kontrol çıkışı gösterilmektedir. Manuel modda uçuşa başlama butonuna basıldıktan sonra kopter uçuşa başlayarak parametrik olarak



Şekil 2. Kullanıcı arayüzü

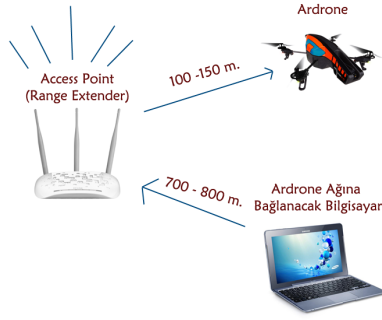
önceden ayarlanan belli bir yüksekliğe çıkarak stabil bir şekilde durmaktadır. Yön ve yükseklik ve yönelme tuşları ile kopterin istenilen konuma hareketi gerçekleştirilebilir. Görüntü çerçevesi altında bulunan % kontrolü anlık olarak kopterin pil seviyesi göstermede kullanılmaktadır. Otonom takip için takip edilmesini istediğimiz nesnenin seçilmesi gereklidir. Takip edilecek nesne seçildikten sonra odak merkez boyut ayarına ihtiyaç vardır. Odak merkez boyut ayarı nesnenin hareketine göre kopterin tepkisini ayarlayan boyut ayarı olarak düşünülebilir. Çok büyük odak merkez boyut ayarı durumunda kopter yavaş hareketlerle nesneyi takip etmeye çalışırken çok küçük odak merkez boyut ayarı durumunda ani tepkiler ile nesneyi takip etme eğiliminde davranır. Otonom takip butonuna basılıp takip edilecek nesne ve odak merkez boyut ayarı belirlendikten sonra kopter otonom nesne takibi işlemlerini gerçekleştirmeye başlayacaktır. Kullanıcının herhangi bir müdahalesine gerek kalmadan nesneyi sürekli görüntü çerçevesi içinde kalacak şekilde pozisyonunu güncellemeye çalışacaktır.



Şekil 3. Manuel ve otonom uçuş kontrolü

Şekil 3'te seçilen nesnenin otonom takibi için ROS'a gönderilen yön ve hareket parametreleri ile kopter yeniden konumlandırılır. Bu işlem döngü içerisinde defalarca tekrarlandığında otonom takip yapılmış olur.

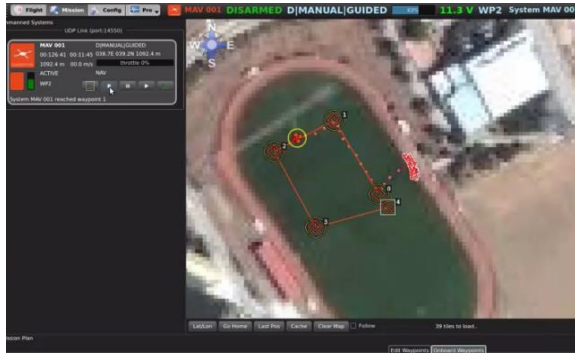
Çalışmalarımızda menzil problemine de yer verdik. Normalde 100–150 metre arasında bir mesafede görüntü ve kontrol işleminin yapıldığı Ardrone'da, Access Point yardımıyla 750–800 metre menzile kadar ulaşılabilir. Bunu Access Point'in Range Extender moduyla sağlamaktayız. Range Extender, kendisini wifi yayını gibi gösterebilen bir donanımdır. İstenilen yayını alıp kendi üzerinden sanki tek bir yayın yapıyormuş gibi gösterebilme özelliği mevcuttur.



Şekil 4. Access Point ile menzil genişletme (Range Extender)

Mesafenin daha çok arttırılması için birden fazla Access Point konulabilmektedir. Ardrone en fazla 100-150 metre bir mesafede yayın yaparken Access Point'in menzili 700-800 metredir. Bu durumda ikinci bir Access Point, ilk Access Point'in menziline ilave olarak 00-800 metrelik ilave menzil kazandıracaktır.

Bu makalede yapılan bir diğer çalışma ise GPS ile belirtilen konumlar üzerinden otonom uçuştur. Şekil 2'de GPS programına geç butonuna basıldığında Şekil 5'te gösterilen GPS koordinatları seçilerek uçuş gerçekleştirilebilen QGroundControl programına geçiş yapılmaktadır. GPS koordinatları manuel olarak harita üzerinde kopterin uçuşmasını istediğimiz noktaların işaretlenmesi ile belirlenir. QGroundControl yazılımı açık kaynak kodlu bir yazılım olup küçük hava aygıtları ile MAVLink açık kaynak kodlu haberleşme protokolünü kullanarak bağlantı kurar. Üç boyutlu harita üzerinde uçuş rotası ve güzergahı belirlenerek uçuş planlanır ve istendiğinde gerçekleştirilir. Uçuş sırasında anlık GPS verileri ile kopterin harita üzerindeki güncel pozisyonu gösterilmektedir. Hedef noktaya vardikten sonra kopterimiz yere inerek uçuşu tamamlamaktadır.



Şekil 5. GPS koordinatları seçilerek uçuş planının gerçekleştirilmesi.

4. Sonuçlar

Gerçekleştirilen Oto Pilot yazılım modülü kopterlerin yerden kontrolü ve uçuş bilgilerinin analizi açısından önemlidir. Yapılan çalışmalar ROS destekli kopterlerde kullanılabilir nitelikte olup güvenlik, izleme amaçlı kullanılacak kopterlerde ait yer kontrol birimi olarak hizmet verecektir. Otonom nesne takip modülünün eklenmesi kopterin güvenlik amaçlı kullanımlarında öne çıkan bir özellik olarak güvenlik

hizmetlerinin sunulmasında yardımcı olacaktır. İleriki çalışmalarda kopterin yön bulma ve engeller ile karşılaştığında rotasını güncelleme ile ilgili çalışmaların var olan yapıya eklenmesi planlanmaktadır. Ayrıca otonom nesne takibini geliştirme amaçlı [15]'de yapılan çalışmaların var olan yapıya aktarılması adına çalışmalar yapılacaktır.

Teşekkür ve Bilgilendirme:

Bu makalede yapılan çalışmalar Tekno Girişim Sermaye Desteği ile desteklenen 0936.TGSD.2014 nolu Kopter Güvenlik Robotu projesi kapsamında desteklenmektedir.

5. Kaynaklar

- [1] http://en.wikipedia.org/wiki/Unmanned_aerial_vehicle. 2014.
- [2] Soumelidis, A., P. G'asp'ar, G. Regula, B.Lantos, "Control of an experimental mini quad-rotor UAV", 16th Mediterranean wConference on Control and Automation Congress Centre, Ajaccio, France, June 25-27, 2008.
- [3] Mikrocopter in Europe www.mikrokopter.de. 2011.
- [4] Draganflyer in US www.draganfly.com. 2011.
- [5] Jorge Artieda, José M. Sebastian, Pascual Campoy, Juan F. Correa, Iván F. Mondragón, Carol Martínez and Miguel Olivares, "Visual 3-D SLAM from UAVs" Journal of Intelligent & Robotic Systems Volume 55, Numbers 4-5, 299-321, DOI: 10.1007/s1084-008-304-8.
- [6] S. Ahrens, D. Levine, G. Andrews, and J. P. How, "Vision-based guidance and control of a hovering vehicle in unknown, gps-denied environments," IEEE International Conference on Robotics and Automation, May 2009.
- [7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", Int. J. Comput. Vision, vol. 60, no. 2, pp. 91-110, 2004.
- [8] Bouabdallah, S., M. Becker, R. Siegwart, "Autonomous miniature flying robots: coming soon! - Research, Development and Results", Robotics & Automation Magazine, IEEE, Vol. 14, No. 3, Sept. 2007, pp. 8 - 98.
- [9]Ax, Markus; Kuhnert, Lars; Langer, Matthias; Schlemper, Jens; Kuhnert, Klaus-Dieter; "Architecture of an autonomous mini unmanned aerial vehicle based on a commercial platform," Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK) , vol., no., pp.1-6, 7-9 June 2010.
- [10] Homan, G., H. Huang, C. J. Tomlin, S. L. Waslander, "Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment" In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Hilton Head, SC, August 2007.
- [11] Kim J. and Sukkarieh S.: Flight Test Results of GPS/INS Navigation Loop for an Autonomous Unmanned Aerial Vehicle (UAV), In International Technical Meeting of the Satellite Division of the Institute of Navigation (ION'02), pp. 510-517, 2002.
- [12] Jong-Hyuk Kim, Salah Sukkarieh Flight Test Results of GPS/INS Navigation Loop for an Autonomous Unmanned Aerial Vehicle (UAV).
- [13] <http://wiki.ros.org/ROS/Tutorials>, 2014
- [14] Nebehay G., Robust Object Tracking Based on Tracking-Learning-Detection, Master Tezi, 2012
- [15] Dilmeh H.; Talu M. F. , "Automatic Detection and Tracking Pedestrian In The Scene Using System With Learning Capabilities", 7th. International Advanced Technologies Symposium, 20