2022 30th Signal Processing and Communications Applications Conference (SIU) | 978-1-6654-5092-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/SIU55565.2022.9864769

PPO Kullanan Derin Pekiştirmeli Öğrenme ile Otonom İHA Navigasyonu
Autonomous UAV Navigation via Deep Reinforcement Learning Using PPO

Bilal Kabas
Elektrik-Elektronik Mühendisliği Bölümü, Abdullah Gül Üniversitesi
Kayseri, Türkiye
bilal.kabas@agu.edu.tr

***Özetçe* —Bu çalışmada, otonom hareket edebilen insansız hava araçları (İHA) için bilgisayar görüsü tabanlı bir navigasyon sistemi önerilmektedir. Önerilen navigasyon sistemi yapay sinir ağı tabanlı yüksek seviyeli bir kontrolcüye dayalıdır. Bu çalışmada bir derin pekiştirmeli öğrenme yöntemi olan PPO (yakınsal politika optimizasyonu) algoritması kullanılarak yapay sinir ağının sürekli bir ödül fonksiyonu ile uçtan uca eğitilmesi sağlanmaktadır. Önerilen sistem, *Unreal Engine* ve *Microsoft AirSim* kullanılarak oluşturulan simülasyon ortamlarında farklı kamera modlarından alınan imge türleri için test edilmiştir. Bu çalışmada ele alınan navigasyon problemi için RGB kamera kullanılarak %96 başarı oranına ulaşılmıştır. RGB kameraların derinlik kameralarına göre daha hafif olması ve eğitilen yapay sinir ağının 170.000'den daha az parametreye sahip olması, önerilen navigasyon sisteminin mikro hava araçlarında kullanılmasını mümkün kılmaktadır. Kaynak kodları erişime açık olarak paylaşılmaktadır*.**

Anahtar Kelimeler—derin pekiştirmeli öğrenme, otonom navigasyon

***Abstract*—In this paper, a computer vision-based navigation system is proposed for autonomous unmanned aerial vehicles (UAV). The proposed navigation system is based on a deep reinforcement learning-based high-level controller. In this paper, proximal policy optimization (PPO), which is a deep reinforcement learning method, is used to train the artificial neural network in an end-to-end way using a continuous reward function. The proposed method has been tested on images obtained from different modalities (RGB and depth) in simulation environments that are created using Unreal Engine and Microsoft AirSim. For the navigation problem that this work is concerned with, a success rate of 96% has been obtained by using RGB cameras. Since RGB cameras are lighter than depth cameras and the trained artificial neural network has a parameter number less than 170.000, the proposed method is suitable to be deployed in micro aerial vehicles. Code is publicly available*.**

Keywords—deep reinforcement learning, autonomous navigation

I. GİRİŞ

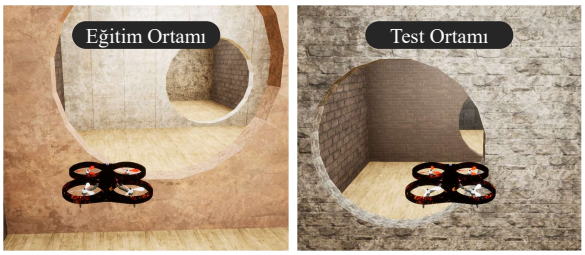
İnsansız hava araçları kargo, arama-kurtarma [1], gözetleme ve keşif [2], [3], gibi birçok farklı görevde kullanılabilmektedir. Bu görevler bir pilotun İHA'yı kumanda aracılığıyla kontrol etmesiyle gerçekleştirilebilir ya da otonom kabiliyetlerle donatılarak bir İHA'nın bu görevleri pilot müdahalesi olmaksızın yerine getirmesi sağlanabilir. Otonom bir İHA'nın üzerinde LIDAR, RADAR, RGB ya da derinlik kamerası gibi sensörler bulunabilir. Bu sensörlerden gelen veriler, İHA'nın üzerinde bulunan ya da İHA'nın iletişim içinde bulunduğu uzak bir bilgisayarda hareket ya da rota planlama algoritmaları ile işlenerek kontrol komutları üretilir. Ambar ve fabrika binaları gibi kapalı ve engellerle dolu bir ortamda İHA'nın otonom olarak bir noktadan diğer bir noktaya herhangi bir nesneye çarpmadan gitmesi zorlu bir görevdir. Dolayısıyla çarpışma önleme, bu tip bir navigasyon probleminin en önemli bileşeni olmaktadır. İHA'nın engellerle arasındaki mesafeyi algılaması LIDAR, RADAR ya da derinlik kamerası gibi sensörlerle doğrudan sağlanabilir. Ancak bu sensörler genellikle pahalı ve ağırlıkları sebebiyle mikro hava araçlarında kullanıma elverişli değildir. Bu sensörler yerine daha ucuz ve hafif olan RGB kameralar kullanılabilir. Ancak derinlik bilgisi RGB kameralar ile doğrudan elde edilememektedir. Bu tip bir navigasyon problemi SLAM (eş zamanlı konum belirleme ve haritalama) gibi klasik yöntemlerle rota planlama algoritmaları beraber kullanılarak çözülebilir. Ancak bu yaklaşım gerektirdiği işlem gücü sebebiyle mikro İHA'nın üzerinde bulunan bilgisayarda uygulanmaya yine elverişli olmayacaktır. Bu durumda İHA ile iletişim halinde olan uzak bir bilgisayar algoritmanın koşturulması için kullanılabilir [4]. Ancak bu durumda da iletişim hattında yaşanan gecikmeler, gürültü ve azami iletişim mesafesi gibi kısıtlamalar söz konusu olacaktır.

Son zamanlarda otonom İHA navigasyonu için derin öğrenme tabanlı ve uçta gerçek zamanlı çalışabilen birçok bilgisayar görüsü tabanlı yaklaşım geliştirilmiştir. Örneğin, RGB imgelerin kullanıldığı yapay sinir ağı tabanlı bir navigasyon algoritması, gömülü platformda uygulanarak bir nano İHA'nın engellere çarpmadan hareket edebilmesini sağlayabilmektedir [5]. Bunun yanı sıra RGB kamera görüntülerinden CNN (evrişimli sinir ağı) kullanılarak derinlik haritaları tahmin edilebilmektedir [6], [7]. Derin pekiştirmeli öğrenmeye dayalı navigasyon yöntemleri son yıllarda dikkat çekmeye başlamıştır. $a_t^* = f(s_t)$ şeklinde İHA'nın t anındaki gözlemini (s_t) doğrudan optimal eyleme (a_t^*) dönüştürecek şekilde bir politika fonksiyonu ya da $Q^*(s_t, a_t)$ şeklinde optimal eylem (a_t^*) ile maksimum değere ulaşan Q^* fonksiyonu derin pekiştirmeli öğrenme kullanılarak yakınsanabilir [8].

Bu bildiride, İHA'nın koridor şeklinde kapalı bir ortamda engelleri aşarak ilerlemesi şeklinde tanımlanan bir navigas-

*Proje linki: https://github.com/bilalkabas/DRL-Nav

978-1-6654-5092-8/22/$31.00 ©2022 IEEE



Şekil 1: Derin pekiştirmeli öğrenmede kullanılan ve *Unreal Engine* ile oluşturulmuş ortamlardan alınan örnek resimler.

Şekil 2: Derin pekiştirmeli öğrenmede kullanılan üç farklı girdi çeşidi için örnek imgeler. A) Derinlik görüntüsü, B) RGB kamera görüntüsü, C) birleştirilmiş ardışık üç gri tonlu görüntü.

yon problemi için derin pekiştirmeli öğrenmeye dayalı bir yaklaşım sunulmaktadır. Sunulan yaklaşımda, yapay sinir ağı tabanlı yüksek seviyeli bir kontrolcü bir derin pekiştirmeli öğrenme tekniği olan PPO algoritması kullanılarak uçtan uca eğitilmektedir. Daha önceki bir çalışmada benzer navigasyon problemi RGB kamera görüntüsü [9] kullanılarak çözülmeye çalışılmıştır. Ancak girdilerden anlamlı niteliklerin çıkarılması için VAE (değişimsel otokodlayıcı) kullanılmıştır. Bu bildiride, ödül fonksiyonu doğru bir biçimde belirlendiğinde girdiden faydalı niteliklerin çıkarılması için fazladan (farklı) sensörlere gerek kalmayabileceği hem derinlik haritaları hem de RGB görüntüler ile elde edilen deneysel sonuçlarla gösterilmektedir.

II. Pekiştirmeli Öğrenme

Pekiştirmeli öğrenmede belirli bir ortamda bulunan bir özne, gerçekleştirdiği eylemlerle bu eylemlerin sonucunda alacağı toplam ödül miktarını artırmaya çalışır. Genellikle bu ortamların çalışma mekanizması MDP (Markov karar süreci) ile modellenir. Markov karar süreci $< \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \rho_0 >$ şeklinde beşli bir veri grubu ile ifade edilir [10]. $\mathcal{S}$ öznenin ve ortamın olası tüm durumlarını, $\mathcal{A}$ öznenin gerçekleştirebileceği tüm eylemleri, $\mathcal{R}$, $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$'de tanımlı ödül fonksiyounu, $\mathcal{P}$, $P(s'|s,a)$ şeklindeki sonraki duruma geçiş olasılık yoğunluk fonsiyonunu ve ρ_0 başlangıç anındaki durumların dağılımını ifade etmektedir. Pekiştirmeli öğrenmede özneye ilişkin herhangi bir model varsıyımı yapılmaz dolayısıyla $P(s'|s,a)$ sonraki duruma geçiş olasılık yoğunluk fonksiyonunun bilinmesine gerek yoktur. Ortam ve öznenin etkileşiminden ortaya çıkan *durum, eylem, ödül* üçlüsü öznenin izlediği yol boyunca örneklenir. T adım boyunca öznenin izlediği yol $(s_{1:T+1}, a_{1:T}, r_{1:T})$ şeklinde ifade edilir. Her bir adımda elde edilen ödül kullanılarak $R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t$ ile toplam ödül miktarı hesaplanır ve öğrenme sürecinde bu terim azami ölçüde arttırılmaya çalışılır.

III. Yakınsal Politika Optimizasyonu

Bu çalışmada bir derin pekiştirmeli öğrenme tekniği olan PPO [11] (yakınsal politika optimizasyonu) algoritması eylem uzayının sürekli olmasına imkan sağladığı ve politikada monotonik iyileşme sağlayabilmesi sebebiyle kullanılmaktadır. Politika optimizasyonu tabanlı algoritmalarda durumları eylemlere dönüştüren politika $\pi_\theta(a_t|s_t)$, yapay sinir ağı ile tanımlanır ve θ bu ağın parametrelerini ifade eder. Bu durumda robotun izlediği tüm yollar ($\tau \sim \pi_\theta$) hesaba katılarak optimizasyon hedefi (1) ile ifade edilir.

$$\max_\theta J(\pi_\theta) = \mathop{\mathbb{E}}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right] \tag{1}$$

PPO'dan daha önceki bir çalışma olan TRPO [12] (güvenli alan politika optimizasyonu) algoritmasında (1) ile belirtilen optimizasyon hedefi önem örneklemesi kullanılarak (2)'de ifade edildiği şekilde değiştirilmektedir. Bu algoritmada yapay sinir ağının parametre güncellemesi eski ($\pi_{\theta_{\text{old}}}$) ve yeni ($\pi_\theta$) politikalar arasındaki KL-ıraksaması belirli bir eşik değerinin (δ) altında kalacak şekilde gerçekleştirilir. Bu kısıtlama (3) ile ifade edilmektedir.

$$\max_\theta \mathop{\mathbb{E}}_{s,a \sim \pi_{\theta_{\text{old}}}} \left[\frac{\pi_\theta(a \mid s)}{\pi_{\theta_{\text{old}}}(a \mid s)} A^{\pi_{\theta_{\text{old}}}}(s,a) \right] \tag{2}$$

$$\mathbb{E}_{s \sim \pi_{\theta_{\text{old}}}} \left[D_{KL}\left(\pi_\theta(\cdot \mid s) \| \pi_{\theta_{\text{old}}}(\cdot \mid s) \right) \right] \leq \delta \tag{3}$$

Öğrenme sırasında parametrelerin güncellenme miktarının belirli bir kısıtlamaya tabi olması, politikanın her güncellemeden sonra iyileşmesini sağlamaya yöneliktir. Ödül sinyali gözetimli öğrenmedeki etiketler kadar güçlü bir gösterge değildir. Bu durumda büyük güncellemeler politikanın geri döndürülemez bir şekilde kötüleşmesine sebep olabilir. PPO'da bu kısıtlama $\pi_\theta / \pi_{\theta_{old}}$ ifadesinin $(1-\epsilon, 1+\epsilon)$ aralığında kalacak şekilde kırpılmasıyla sağlanmaktadır. PPO'da optimizasyon hedefi (4)'teki ifadenin azami arttırılması olarak tanımlanır.

$$\begin{aligned} &\mathop{\mathbb{E}}_{s,a \sim \pi_{\theta_{\text{old}}}} \left[\min\left(r_t(\theta)\hat{A}, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A} \right) \right], \\ &\text{where,} \quad r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}. \end{aligned} \tag{4}$$

PPO'da, TRPO'daki gibi KL-ıraksama hesaplanmadan güncelleme miktarları daha sade bir teknik ile kısıtlanmış olur.

IV. Sunulan Yöntem

A. Simülasyon Ortamlarının Oluşturulması

Bu çalışmada kullanılan simülasyon ortamları *Unreal Engine* oyun motoru kullanılarak oluşturulmuştur. İHA'nın bu ortamlarda kullanılabilmesi ise *Microsoft AirSim* [13] eklentisi ile mümkün olmaktadır. AirSim'in sunduğu API (uygulama programlama arayüzü) ile İHA'dan sensör verileri alınabilmekte ve İHA'ya çeşitli kontrol komutları gönderilebilmektedir. Şekil 1'de gösterildiği üzere eğitim ve test için iki farklı ortam oluşturulmuştur. Eğitim ortamında 15 odacık bulunmaktadır ve her odacık birbirinden üzerinde İHA'nın geçebilmesi için dairesel geçitler bulunan duvarlarla ayrılmıştır. Test ortamında ise 16 odacık ve 15 geçit bulunmaktadır. Geçitlerin çapları sabit ancak merkezlerinin pozisyonları farklıdır. Aynı ortamda farklı doku tipleri kullanılmıştır. Eğitim ve test ortamlarında geçitlerin bulunduğu duvarların dokuları birbirinden farklıdır.

B. Pekiştirmeli Öğrenme Formülasyonu

Sunulan derin pekiştirmeli öğrenmede robotun ortam ile olan etkileşimleri *durum, eylem, ödül* (s_t, a_t, r_t) üçlüsü ile ifade edilir. Durum bilgisi doğrudan yapay sinir ağının girdisidir. Çıktı ise robotun gerçekleştireceği eylemdir. Robot bu durum ve eyleme göre ödüllendirilir.

1) Durum: İHA'ya ve ortama ilişkin t anındaki durum bilgisi $s_t \in \mathbb{R}^{H\times W\times C}$ şeklindeki kamera görüntülerinden oluşmaktadır. H, yükseklik piksel sayısını; W, genişlik piksel sayısını ve C, kanal sayısını ifade eder. Durum bilgileri yapay sinir ağının girdileridir. Bu çalışmada kullanılan üç farklı girdi çeşidi Şekil 2'de gösterildiği gibi derinlik görüntüsü, RGB kamera görüntüsü ve birleştirilmiş üçlü gri tonlu görüntüdür. İHA'nın algıladığı derinlik görüntüsü ve RGB görüntü AirSim kullanılarak elde edilir. Derinlik görüntüsünde İHA'ya yakın nesneler daha koyu renkte görünür. Çoklu gri görüntüyü elde etmek için öncelikle t, $t-1$ ve $t-2$ anlarındaki RGB kamera görüntüleri gri tonlu görüntülere dönüştürülür. Bu gri tonlu görüntüler birleştirilerek boyutu $\hat{H}\times(\hat{W}\times 3)\times\hat{C}$ olan çoklu gri görüntü elde edilmiş olur.

2) Eylem: Şekil 3'te gösterildiği gibi İHA'nın t anındaki eylemi $a_t = [v_{y,t}, v_{z,t}]$ şeklinde, İHA'nın y ve z eksenlerindeki hızını belirler. İHA x yönünde ise her zaman $v_x = 0.4$ m/s sabit hızla ilerlemektedir. İHA'nın eylemleri, v_y ve v_z değerleri $[-0.6, 0.6]$ m/s aralığında kalacak şekilde sınırlandırılmıştır ve xy ile xz düzlemlerinde yaklaşık 113 dereceye kadar manevralar yapabilmektedir. İHA, z ekseninde hiçbir zaman dönme hareketi yapmamaktadır.

3) Ödül: Ödül fonksiyonu r_t, (5)'te gösterildiği gibi iki terim içerir. İlk terim olan r_t^d fonksiyonu (6)'da ifade edildiği gibi duvarda bulunan geçitin İHA'ya olan Öklid uzaklığı bulunarak hesaplanır.

$$r_t = r_t^d + r_t^c \tag{5}$$

$$r_t^d = 30 \times e^{-\|p_{a,t} - p_{h,t}\|_2} \tag{6}$$

r_t^c fonksiyonunun değeri ise (7)'de ifade edildiği gibi belirli şartlara göre belirlenmektedir. $\mathbb{1}_{\text{coll},t}$ fonksiyonu, t anında çarpışma gerçekleşmiş ise 1, aksi halde 0 değerini alır. Benzer şekilde, $\mathbb{1}_{\text{miss},t}$ fonksiyonu, geçidin görüntüsünün İHA'nın kamerasından kaybolup kaybolmadığını gösterir ve değerin 1 olması durumunda $r_t^c = -100$ olur.

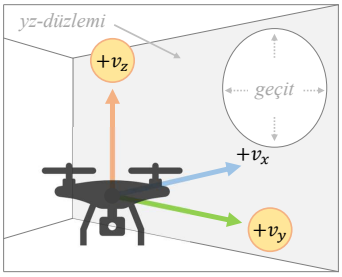
$$r_t^c = \begin{cases} -100, & \mathbb{1}_{\text{coll},t} = 1 \\ -100, & \mathbb{1}_{\text{miss},t} = 1 \\ 0, & \text{diğer durumlarda} \end{cases} \tag{7}$$

$\mathbb{1}_{\text{miss},t}$ gösterge fonksiyonunun değeri ise (8)'deki kurala göre belirlenir. r_h, geçidin yarıçapını; x_h, geçidin x pozisyonunu; x_0, İHA'nın başlangıç anındaki x pozisyonunu; ve θ, İHA'nın kamera görüntüsünün görüş alanını ifade eder.

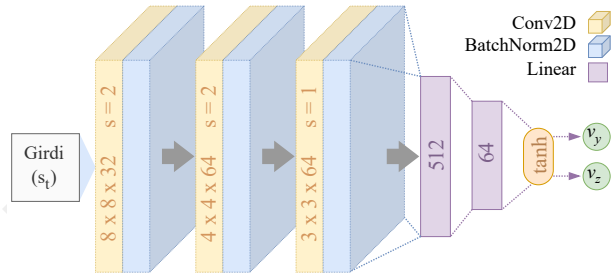
$$\mathbb{1}_{\text{miss},t} = \begin{cases} 0, & d - r_h > (x_h - x_0 - x) \times sin(\frac{\theta}{2}) \\ 1, & \text{diğer durumlarda} \end{cases} \tag{8}$$

C. Evrişimli Sinir Ağı Mimarisi

PPO'da eğitilen yapay sinir ağının mimarisi Şekil 4'te gösterilmektedir. Üç evrişimli ve iki tam bağlantılı katmandan oluşan sinir ağında, [14]'ten farklı olarak evrişimli katmanlar



Şekil 3: İHA x yönünde v_x sabit hızıyla hareket eder ve İHA'nın yz-düzlemindeki hareketi yapay sinir ağından çıkan v_y ve v_z hızları ile sağlanır.



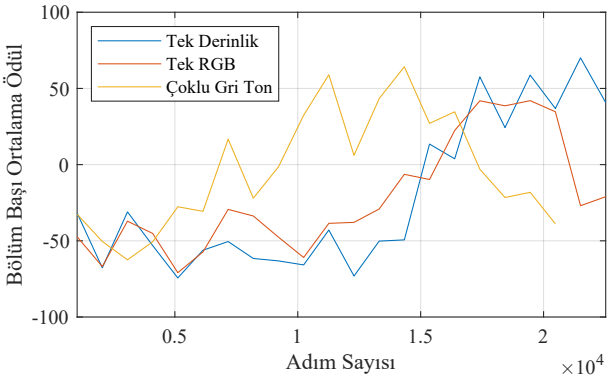
Şekil 4: Derin pekiştirmeli öğrenmede kullanılan beş katmanlı evrişimli sinir ağı.

arasında ReLU (doğrultulmuş doğrusal ünite) aktivasyon fonksiyonunun öğrenmeyi olumsuz etkilememesi için normalleştirme katmanları bulunmaktadır. Ayrıca ilk evrişimli katmanda atlama miktarı (s) dört yerine iki olarak belirlenmiştir. Böylece geçitlerin kamera görüntülerindeki pozisyonuna ilişkin bilgi sonraki katmanlara daha doğru bir şekilde aktarılmış olur.

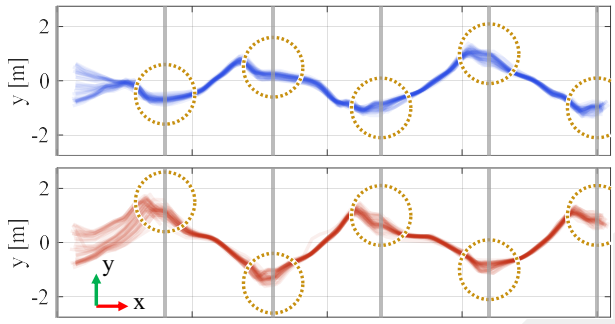
V. Deneysel Çalışmalar ve Sonuçlar

1) Eğitim: Yapay sinir ağı PPO algoritması ile üç farklı girdi çeşidi kullanılarak uçtan uca eğitilmektedir. Girdilerden derinlik görüntüsü, $50\times50\times1$; tek RGB görüntü, $50\times50\times3$ ve çoklu gri görüntü, $150\times50\times1$ boyutlarındadır. Her 1024 simülasyon adımında bir, güncel politika ile 20 deneme yapılarak ortalama ödül miktarı hesaplanır. Şekil 5'te bu değerlendirmeye ilişkin sonuçlar gösterilmektedir. Genellikle çoklu gri görüntü, tek RGB görüntünün kullanıldığı duruma göre daha erken yüksek ödül değerlerine ulaşmaktadır. Nihai model ortalama ödül miktarının maksimum olduğu noktada elde edilmektedir.

2) Test: Üç farklı girdi ile eğitilen modellerle iki farklı test gerçekleştirilmiştir. İlk testte, Tablo I'de gösterildiği üzere, eğitim ve test ortamlarında 200 denemede elde edilen ortalama geçidi kazasız aşma oranı ve uçuş mesafesi, üç model ve rastgele eylemler için ayrı ayrı hesaplanmıştır. Çoklu gri ton görüntü ile eğitilen model tek RGB imge kullanılarak eğitilen modele göre %4 daha iyi performans göstermektedir. İkinci testte ise farklı zorluk seviyelerine sahip iki test ortamında derinlik görüntüleri ve çoklu gri ton görüntüler kullanılarak eğitilmiş olan modellerin performansları karşılaştırılmıştır. Şekil 6'da derinlik görüntüleri ile, Şekil 7'de ise çoklu gri ton görüntüler ile eğitilen modellerin orta ve zor seviyeli ortamlarda yapılan 40 denemede izledikleri yollar gösterilmektedir. Orta ve zor seviyelerde derinlik görüntüleri ile sırasıyla 19.7



Şekil 5: Farklı girdi tipleri için (yalnız derinlik, tek RGB, çoklu gri) yapılan eğitimlerde bölüm başı ödül miktarının her 1024 simülasyon adımındaki değişimi.



Şekil 6: Orta (mavi) ve zor (kırmızı) test ortamlarında derinlik görüntüsü ile eğitilmiş modeli kullanan İHA'nın izlediği yollar. Çemberler duvarlardaki geçitleri ifade etmektedir.

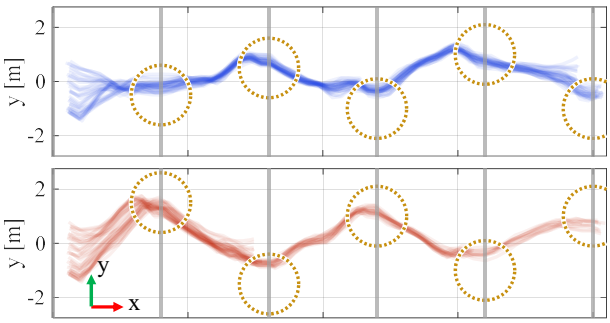
m ve 19.4 m; çoklu gri ton görüntülerle 18.2 m ve 12.7 m ortalama uçuş mesafesi değerleri elde edilmiştir.

VI. Sonuç

Bu bildiride, PPO kullanan derin pekiştirmeli öğrenme tabanlı bir navigasyon algoritması sunulmuştur. Sunulan algoritmanın performansı, *Unreal Engine* ve *AirSim* üzerinde geliştirilen simülasyon ortamında test edilmiştir. Rastgele ortam testlerinde çoklu gri ton görüntü ile derinlik görüntüsünün performansına yakın sonuçlar elde edilmiştir. Ardışık testlerde orta zorlukta ortalama uçuş mesafeleri arasında 2 m'den daha az fark varken zorluk seviyesi arttığında fark 6.7 m'ye çıkmaktadır. Bu bildiride önerilen navigasyon sistemi, eğitilmiş olan yapay sinir ağının boyutu ve RGB kamera ile elde edilen sonuçlar göz önünde bulundurulduğunda mikro hava araçları için uygulanabilir durumdadır. İleride farklı derin pekiştirmeli öğrenme algoritmaları ile daha karmaşık simülasyon ortamlarında, sistemdeki aksiyon seti ve ödül fonksiyonu geliştirilerek çalışmalar yapılacaktır.

TABLO I: Farklı Girdi Tiplerine Göre Eğitim ve Test Ortamlarındaki Başarı Oranı ve Uçuş Mesafesi

Girdi türü	Ortalama başarı oranı (%)		Ortalama uçuş mesafesi (m)	
	Eğitim	Test	Eğitim	Test
Tek derinlik	%100	%99.5	3.877	3.888
Tek RGB	%98.5	%92	3.918	3.83
Çoklu gri görüntü	%98.5	%96	3.892	3.921
Rastgele eylemler	%11	%22.5	3.232	3.334



Şekil 7: Orta (mavi) ve zor (kırmızı) test ortamlarında çoklu gri ton görüntüler ile eğitilmiş modeli kullanan İHA'nın izlediği yollar.

VII. Bilgilendirme

Bu bildiri Dr. Sedat Özer'in rehberliğinde hazırlanmıştır, kendisine katkılarından dolayı teşekkürlerimi sunarım.

Kaynaklar

[1] M. Atif, R. Ahmad, W. Ahmad, L. Zhao, and J. J. P. C. Rodrigues, "Uav-assisted wireless localization for search and rescue," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3261–3272, 2021.

[2] D. Gozen and S. Ozer, "Visual object tracking in drone images with deep reinforcement learning," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 10 082–10 089.

[3] B. M. Albaba and S. Ozer, "Synet: An ensemble network for object detection in uav images," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 10 227–10 234.

[4] O. Dunkley, J. Engel, and D. Cremers, "Visual-inertial navigation for a camera-equipped 25 g nano-quadrotor," 2014.

[5] D. Palossi, A. Loquercio, F. Conti, E. Flamand, D. Scaramuzza, and L. Benini, "A 64-mw dnn-based visual navigation engine for autonomous nano-drones," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8357–8371, 2019.

[6] M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia, "J-mod2: Joint monocular obstacle detection and depth estimation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1490–1497, 2018.

[7] P. Chakravarty, K. Kelchtermans, T. Roussel, S. Wellens, T. Tuytelaars, and L. Van Eycken, "Cnn-based single image obstacle avoidance on a quadrotor," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 6369–6374.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[9] Z. Xue and T. Gonsalves, "Monocular vision obstacle avoidance uav: A deep reinforcement learning method," in *2021 2nd International Conference on Innovative and Creative Information Technology (ICITech)*, 2021, pp. 1–6.

[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, 2018.

[11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.

[12] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, "Trust region policy optimization," in *32nd International Conference on Machine Learning, ICML 2015*, 2015.

[13] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," in *Springer Proceedings in Advanced Robotics*, 2018.

[14] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, 2015.