Pekiştirmeli Öğrenme ile Gösterimlerden Ters Beceri Öğrenme
Inverse Skill Learning From Demonstrations via Reinforcement Learning

Bora Toprak Temir
Bilgisayar Mühendisliği Bölümü
Boğazici Üniversitesi
İstanbul, Türkiye
bora.temir@std.bogazici.edu.tr

Emre Uğur
Bilgisayar Mühendisliği Bölümü
Boğazici Üniversitesi
İstanbul, Türkiye
emre.ugur@bogazici.edu.tr

***Özetçe* —Robotikte semantik olarak anlamlı becerilerin çoğu, benzer becerilerin öğrenilmesinde kullanılabilir. İleri-ters beceri çiftleri, bir becerinin diğerini öğrenmek için kullanılabildiği böyle bir beceriler kümesidir. Bu çalışmada, pekiştirmeli öğrenme (PÖ) ve gösterimlerden öğrenme (GÖ) yöntemlerini birleştirerek, ileri beceriye ait gösterimlerden ters beceriyi öğrenebilen yenilikçi bir PÖ mimarisi öneriyoruz. Önerilen yaklaşımımız, ileri beceriye ait gösterimleri kullanarak ters beceri için uygun bir ödül fonksiyonu öğrenir, davranış klonlama ile gösterimlerin geriye sarılmış versiyonlarını taklit ederek eylem fonksiyonuna ön eğitim verir ve sonrasında öğrenilen ödül fonksiyonunu kullanarak PÖ ile eylem fonksiyonunu iyileştirmeye devam eder. Bu yöntem, standart PÖ'nün düşük örneklem verimliliği, zorlayıcı ödül şekillendirme süreçleri ve yüksek boyutlu keşif problemlerini hafifletir. Ayrıca, verilen gösterimlerden farklı, yeni beceriler yaratılmasını sağlar ki bu da saf GÖ yöntemleriyle mümkün değildir.**

Anahtar Kelimeler—Pekiştirmeli Öğrenme, Beceri Transferi, Robotik

***Abstract*—In robotics, most semantically meaningful skills can be utilized to learn similar skills. Forward - inverse skill pairs are such a family of semantically meaningful skills where one skill can be used to learn the other. In this paper, we propose an architecture that is able to learn an inverse skill given the demonstrations of the forward skill using Reinforcement Learning (RL) coupled with Learning from Demonstrations (LfD). Our architecture uses learns and appropriate reward function for the inverse skill using the demonstrations of the forward skill, warm-starts the policy by learning to imitate the rewinded version of the demonstrations using Behaviour Cloning, and continues training with RL using the learned reward function. This approach mitigates the poor sample efficiency, tedious reward shaping and difficult high-dimensional exploration problems of RL while being able to create novel skills different from the given demonstrations, something that can't be done by pure LfD.**

Keywords—Reinforcement Learning, Skill Transfer, Robotics

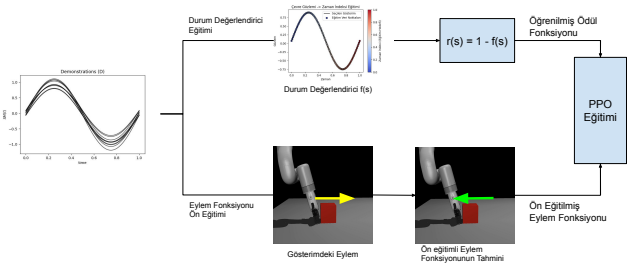
I. GİRİŞ

Çevresindeki değişimleri geri alabilme, insanlar tarafından akıcı bir şekilde gerçekleştirilen bir yetidir. Bunun için insanlar sadece değişimi yaratan yörüngenin tersini uygulamak ile yetinmez, tersini almak için gerekli değişiklikleri de otomatik olarak yapabilir. Dağılan eşyaların geri yerlerine konulması ya da elektromekanik teçhizatların demonte edilmesi gibi işlerin otomatik bir şekilde yapılması hedeflendiğinde robotlarda da önemli bir yeti olduğu tartışılabilir. Örnek olarak, insanlar bir nesnenin montajını yapabilirler, robotun gelecekte nesneyi geri dönüşüm için demonte etmesi isteniyor olabilir ancak bunun için demontajın nasıl yapıldığını doğrudan göstermektense robotun montajı izleyerek nasıl demonte edeceğini çıkarması otomasyonu kolaylaştıracaktır.

Bu çalışmada, bir becerinin sadece nasıl yapıldığına dair gösterimlerini kullanarak, bu becerinin etkisini geri alacak bir "ters beceri" öğrenmek için yenilikçi bir pekiştirmeli öğrenme (PÖ) yöntemi sunuyoruz. Bu çalışma boyunca, "beceri" ifadesi robotun etrafını belirli bir girdi durumunu farklı bir çıktı durumuna getirme kabiliyeti olarak, "ters beceri" ifadesi de karşılık gelen becerinin oluşturacağı çıktı durumunu girdi durumuna geri getirme kabiliyeti olarak tanımlanmıştır.

Pekiştirmeli Öğrenme (PÖ), ortamdan verilen bir ödül fonksiyonu için uzun vadeli alınan ödülü maksimize edecek ideal eylem fonksiyonunu öğrenir. İdeal eylem fonksiyonuna dair sıfır bilgiden başlamak yerine takviye olarak ideale yakın gösterimlerden yararlanmak, pekiştirmeli öğrenmenin hesaplama maliyetini önemli ölçüde düşürdüğü [1], ödül şekillendirme sürecine ihtiyacı ortadan kaldırarak hem öğrenmeyi kolaylaştırdığı hem de elle yapılan ödül şekillendirmelerde karşılaşılan amacı yanlış tarif etme sorununu çözdüğü için son yıllarda popülerleşen bir yaklaşımdır [2]–[4]. Bu yaklaşımlarda çoğunlukla eylem fonksiyonundan bağımsız öğrenme algoritmaları (off-policy algorithm) kullanılır ve eylem fonksiyonu güncellemelerinde deneyim belleği deneyim belleği ile birlikte bir "gösterim belleği" de kullanılır [5]. Bu sayede öğrenimin erken aşamalarında içi gösterimlerle doldurulan gösterim belleğine ağırlık verilerek gösterimlerdeki ustalık taklit edilirken ilerleyen aşamalarda deneyim belleğine ağırlık verilerek gösterimleri aşan bir model eğitilebilir. Ancak biz gösterimlerdeki görevi değil, bu görevin sonuçlarını geri almayı öğrenmeye çalıştığımız için bu yaklaşımlar kullanılamaz, çünkü bu yaklaşımlarda hep gösterimlerin öğrenilecek ideal eylem fonksiyonundan geldiği ya da en azından yakın ustalık seviyesine sahip

979-8-3315-6655-5/25/$31.00 ©2025 IEEE



Şekil 1: Mimari Tasarımı. Becerinin gösterimleri sisteme girdi olarak verilir, sistem ters beceriyi teşvik eden ödül fonksiyonu ve gösterimlerin tersini uygulayan ön eğitimli eylem fonksiyonu öğrenir

olduğu varsayılmıştır. Bu sorunu aşıp gösterimlerden farklı olan rotalar öğrenmeye olanak sağlamak için yöntemimizde eylem fonksiyonuna bağlı bir öğrenme algoritması (on-policy algorithm) önerip gösterimleri tamamlanan görevi anlayarak bir ödül öğrenmek ve eylem fonksiyonuna ön eğitim vermek için kullandık.

Bu çalışmanın katkıları aşağıda sıralanmıştır:

- Bir becerinin tersini PÖ ile öğrenmek için, verilmiş düz gösterimleri kullanan bir ödül fonksiyonu öğrenme metodu önerilmiştir.
- PÖ sürecini kısaltmak için gösterimlerin geriye sarımlarını taklit eden eylem fonksiyonu ön eğitimi önerilmiştir.
- Ön eğitimli model ve öğrenilmiş ödül ile eğitimini tamamlamış model elle şekillendirilmiş ödül ile eğitilen PÖ ile karşılaştırılmıştır.

II. YÖNTEM

Geliştirilen mimarinin genel tasarımı Şekil 1'de gösterilmektedir. Öncelikle ters beceriyi eğitmek için kullanılmak üzere orijinal becerinin gösterimleri toplanır. Bir gösterim, robotun hareketi boyunca gerçekleşme zamanıyla etiketlenmiş gözlemler listesidir. Bir ana ait gözlem, o anda robot tarafından algılanan her şeyi (robotun kendi eklem açıları, etrafındaki objelerin konumları) içerir. Eğitim için kullanılmadan önce, gösterimlerin zaman etiketleri 0 ilk an, 1 son an olacak şekilde normalize edilir. Eğitim üç aşamadan oluşur: (1) Ödül fonksiyonu olarak kullanılacak olan "durum değerlendirici ağ"ın eğitimi, (2) gösterimleri ters taklit etmek üzere eylem fonksiyonu ön eğitimi, (3) ön eğitimli eylem fonksiyonunun öğrenilen ödül fonksiyonu ile PÖ kullanılarak eğitimi. İlk iki aşama sadece gösterimlere dayandığından için paralel şekilde gerçekleştirilebilir.

A. Durum Değerlendirici Ağ

Bir beceri, etrafındaki objeleri uygun bir başlangıç durumundan alıp hedef durumuna getirme kabiliyeti olarak tanımlandığı için, ödül fonksiyonu $r(s)$, s durumunun başlangıç durumuna uzaklığını ölçen bir metrik olarak tanımlanmalıdır. Bunun için durum değerlendirici ağ, girdi olarak aldığı duruma gösterimlerin hangi anında karşılaşıldığını tahmin edecek şekilde eğitilir. Zaman etiketleri 0-1 arasında normalize edilmiş olduğu için, bir durumun gösterimdeki zamanını belirlemek, o durumun ne kadar "ilerlemiş" olduğunu tahmin etmek anlamına gelir. Durum değerlendirici ağ $f(s)$, girdisi olan s durumunun gösterimdeki normalize edilmiş zamanını tahmin edecek şekilde eğitilir:

$$f(s) \approx t, \quad t \in [0, 1]$$

Bu yaklaşım sayesinde, ters beceriyi öğrenen PÖ ajanı, bir durumun başlangıç durumuna olan mesafesini zamansal bağlamda öğrenmiş olur. Ajan, belirli bir durum için düşük bir $f(s)$ tahmini aldığında, bu durumun başlangıç durumuna benzediğini, yüksek tahmin aldığında ise başlangıca benzediğini anlayacaktır. Bunu kullanarak ajana $r(s) = 1 - f(s)$ şeklinde bir ödül fonksiyonu verilebilir. Gösterimde daha geç karşılaşılan durumlarda $f(s)$ artacaktır, dolayısıyla ajan ödülü maksimize etmek için gösterimde erken karşılaşılan durumlara ulaşmayı öğrenir. Ayrıca, ödül fonksiyonunun bu şekilde öğrenilmesi gösterimlerdeki her bir anın durum değerlendirici ağ eğitiminde geriyayılım için ayrı bir girdi olarak kullanılabilmesini sağladığı için, deneyler bölümünde gösterdiğimiz gibi, düşük gösterim sayısıyla bile oldukça stabil ve doğru şekilde öğrenilebilmesini sağlar. Ağ, gösterimlerden örneklenmiş durum-zaman çiftleri kullanılarak ortalama karesel hata (MSE) kaybı ile eğitilir. Bunun yanında, tahminlerin ani küçük sıçramalar yapmasını engellemek için toplam varyasyon kaybı da kullanılır. Bunlarla birlikte kayıp fonksiyonunun son hali:

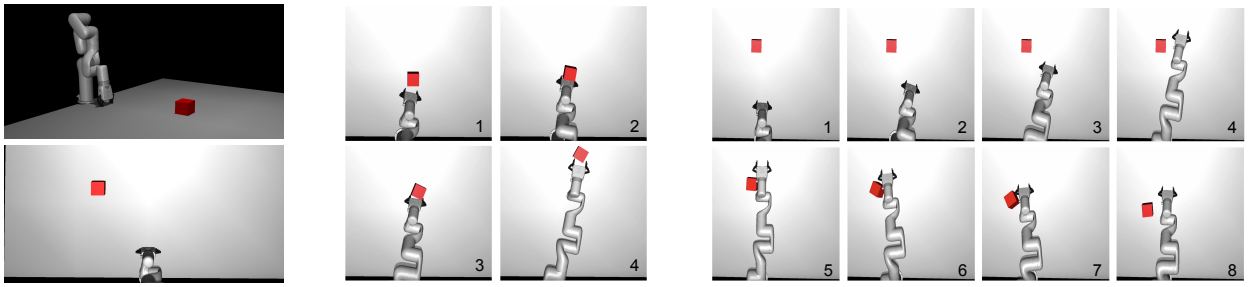
$$L = \frac{1}{N}\sum_{i=1}^{N}(f(s_i) - t_i)^2 + c_{\text{TVK}}\frac{1}{N-1}\sum_{i=2}^{N}|f(s_i) - f(s_{i-1})|$$

şeklindedir. Burada (s_i, t_i), gösterimlerden rastgele N örnekleme yapılarak seçilip rastgele numaralandırılmış durum-zaman çiftleri, c_{TVK} toplam varyasyon kaybı terimi için ağırlık katsayısı, ve $f(s_i)$ durum değerlendirici ağın s_i durumu için tahmin ettiği zamandır.

B. Eylem Fonksiyonu Ön Eğitimi

Başlangıçta rastgele hareketler yapan PÖ ajanı eğer anlamlı ödüller bulamazsa öğrenmekte tamamen başarısız olabilir, ya da hata uzayında sığ bir minimumda takılabilir. Bu durum özellikle yüksek boyutlu sürekli eylem uzaylarında daha büyük bir sorun teşkil eder, çünkü rastgele keşif ile doğru durumtu keşfetmek boyut sayısına üstel olarak bağlıdır. Bu nedenle, PÖ sürecini hızlandırmak ve anlamlı keşfi teşvik etmek için bir ön eğitim stratejisi uygulanır. Bu ön eğitim, gösterimlerin tersine çevrilmiş versiyonlarını taklit edecek bir eylem fonksiyonunu göstererek öğretme yoluyla eğiterek PÖ sırasında ajanın başlangıçtan itibaren anlamlı hareketler yapmasını sağlar. Gösterimler, ileri becerinin nasıl uygulanacağını öğrenmek için kaydedilmiş olup, doğrudan ters beceriyi öğrenmek için kullanılamazlar. Ancak, gösterimler zamanda geriye doğru uygulanırsa, ters beceriye rastgele hareket etmekten daha yakın bir hareket örneği oluştururlar. İleri beceri ve geri becerinin birbirlerine benzemediği durumlarda bile, ileri rotanın tersini uygulamak yolundaki objelerle etkileşime geçmeye sebep olacağı için PÖ ajanının öğrenme sürecini erkenden başlatır. Ön eğitim için eylem fonksiyonu gösterimleri tersten uygulayacak şekilde davranış klonlama metoduyla eğitilmiştir. Eylem fonksiyonu, her durum s için çıktı eylemleri bir olasılık dağılımı olarak verir:

$$\pi(a \mid s) = s \text{ durumunda } a \text{ eylemininin olasılığı}$$



(a) Deney Ortamı (b) İtme Gösterimleri (c) Öğrenilen Geri Çekme Hareketi

Şekil 2: Deney ortamı ve hareket gösterimleri.

Model, bu olasılık dağılımından rastgele örnekleme yaparak eylemleri seçer. Eğitimde eylem fonksiyonu verilen durum girdisi s_n'e karşı gösterimde o durumdan önce karşılaşılan s_{n-1}'e ulaşan eylemi tahmin edecek şekilde optimize edilir. Eylem fonksiyonunun bu eylemi doğru tahmin etmesinin beklenen olasılık değeri maksimize edilmek istendiği için, olasılığın negatif logaritması kayıp fonksiyonu olarak kullanılır:

$$L_{\text{ön eğitim}} = - \sum_{(s_{i-1}, a_{i-1}, s_i) \sim \mathcal{B}} \log \pi_{\text{ön}}(a_{i-1} \mid s_i)$$

Burada s_i gösterimlerden rastgele seçilen bir durum, a_{i-1} gösterimdeki bu duruma sebep olan eylem ve $\mathcal{B}$ gösterim kümesinden rastgele seçilen bir mini-kümedir (batch). Bu kayıp fonksiyonu, modelin ters gösterimlere mümkün olduğunca yakın eylemler üretmesini sağlar.

C. Pekiştirmeli Öğrenme

Ön eğitim tamamlandıktan sonra, RL ajanı durum değerlendirici ağ tarafından sağlanan ödülü kullanarak eylem fonksiyonunu son haline getirir. Bu aşamada, ajan önceden eğitilmiş politikayı başlama noktası olarak kullanır ve hareketlerini daha iyi hale getirmek için öğrenmeyi sürdürür.

Bu çalışmada, PÖ eğitimi sürekli uzayda stabil bir şekilde çalışabilmesi, yüksek performansı ve kolay kullanımı nedeniyle PPO (Proximal Policy Optimization) [6] algoritmasıyla gerçekleştilmiştir. Bu yöntem, eylem fonksiyonu güncellemelerinde ani değişiklikleri önlemek için kısıtlayarak hata uzayında yanlış bir yöne büyük bir adım atılmamasını sağlar. Önceden eğitilmiş eylem fonksiyonu üzerine inşa edilen bu süreç, her güncellemeden sonra çevreyle tekrar etkileşime girerek öğrenilmiş ödül fonksiyonundan optimal şekilde ödül alan eylem fonksiyonunu öğrenir.

III. Deneyler

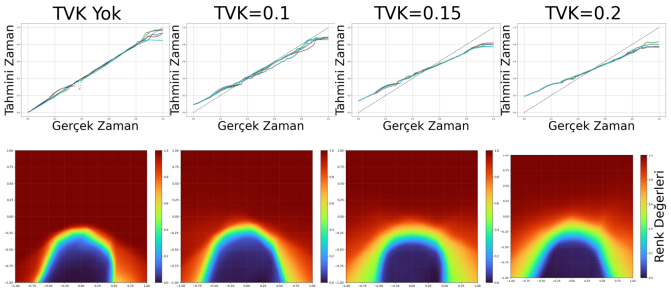
Deneylerde ele alınan problem tam gözlenebilir bir Markov Karar Süreci (MDP) olarak modellenmiştir. Deney ortamı olarak MuJoCo fizik simülasyonunda 7 serbestlik dereceli kolu ile xArm7 robotu kullanılmıştır. Temel senaryoda, robota etrafında ulaşabileceği herhangi bir rastgele pozisyonda oluşturulan küp şeklindeki objeleri itme gösterimleri verilmiş olup, modelin bu gösterimlerden kübü geri çekme becerisini öğrenmesi beklenmiştir. Robotun aksiyon uzayı $\mathbb{R}^3$ (x, y, z), gözlem uzayı ise robotun 7 eklem açısı ile objenin 3 boyutlu uzay koordinatı ve 4 boyutlu uzaydaki açısına karşılık gelecek şekilde $\mathbb{R}^{14}$ olarak belirlenmiştir. Robotun objeyi itmesini sağlamak kolaydır fakat çekme görevi birden çok adım gerektiren daha zorlu bir görevdir. Başarılı olmak için robotun ya önce objenin yanından ya da üstünden geçerek arkasından kendine doğru itmeyi öğrenmesi; ya da objenin üzerine gelip doğru pozisyonda hareket etmeden kalarak kavrayıcısını kapaması ve ardından çekmeyi öğrenmesi gerekmektedir.

Her simülasyon güncellemesinde eylem fonksiyonuna gözlem olarak robotun kendi eklem açıları ve nesne pozisyonu verilmiştir. Aksi belirtilmediği sürece eğitimde 20, validasyonda ise test çeşitliliğini artırmak amacıyla 1000 boyutlu gösterim kümeleri kullanılmıştır. Ön eğitimde 32 büyüklüğünde yığınlar ve 10^{-3} öğrenme oranı ile davranış klonlama, PÖ eğitiminde ise Stable Baselines 3'ün [7] standart hiperparametreleri kullanılmıştır. Şekil 2'de deney ortamı ile itme gösterimlerinden ve eğitilmiş modelin geri çekme hareketinden örnekler gösterilmiştir.

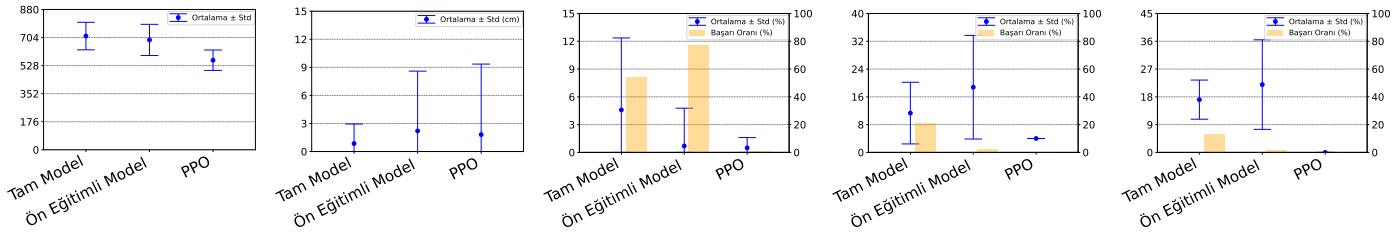
A. Öğrenilen Ödül Performansı

Bu deneyin amacı durum değerlendirici ağıyla öğrenilen ödülün başarısını, oluşan ödül uzayının PÖ ağı için görevin inceliklerini kavrayan bir ödül verip veremediğini ve ödül uzayının toplam varyasyon kaybı (TVK) katsayısıyla nasıl değiştiğini ölçmektir.

Şekil 3'te sistemin beklendiği gibi itme gösterimleri üzerinden çekmeyi teşvik edecek sürekli ve yumuşak bir yüzey



Şekil 3: Farklı TVK katsayılarıyla Durum Değerlendirici Ağın ödül uzayı görselleri ve tahmin edilen zaman - gerçek zaman grafikleri. Ödül uzaylarında renklerin hangi durum değerlendirici ağ çıktılarına karşılık geldiği sağda bulunan renk çubuğunda rapor edilmiştir, mavi düşük, kırmızı yüksek değerleri göstermektedir.



(a) Uzaklık Ödülü (b) Son Mesafe (c) Çekim miktarı (≥ 0) (d) Çekim miktarı (≥ 0.01) (e) Çekim miktarı (≥ 0.05)

Şekil 4: Çekme deneyinde 5 farklı metriğe ait sonuçlar. Çekim miktarı deneylerindeki miktarlar sadece başarı filtresinden geçen episodlardaki sonuçlar dikkate alınarak oluşturulmuştur.

oluşturabildiğini görsel olarak göstermektedir. Şekilde üst tarafta farklı toplam varyasyon kaybı katsayılarına göre eğitilmiş durum değerlendirici ağlar tarafından oluşturulan ödül uzayları, alt tarafta ise rastgele seçilen gösterim rotalarının her anı için tahmini zaman - gerçek zaman grafikleri gösterilmiştir. TVK arttıkça bir yandan ödül fonksiyonunun daha yumuşak geçişlerle daha yönlendirici bir ödül uzayı oluşturduğu, bir yandan da modelin ekstrem zaman değerlerinin tahmininden kaçındığı, ancak hala ilk ve son değerler arasında doğrusal olan ve 0.5 noktasından geçen bir grafik oluşturduğu görülmektedir. Grafikteki bu kayma, sadece doğrusal öteleme ve ölçeklenmeye karşılık geldiği için ödül fonksiyonunun doğasını etkilemeyecek niteliktedir. Deneylerin devamında 0.15 TVK ile eğitilen durum değerlendirici ağ kullanılmıştır.

B. Ödül Şekillendirme ile Karşılaştırma

Bu deneyde sistemimizin başarısı, referans olarak elle şekillendirilmiş ödül ile eğitilen PPO modeli ile karşılaştırılmaktadır. Bütün deneylerde sadece ön eğitimden geçirilmiş model, ön eğitimden sonra eğitimini tamamlamış model ve PPO üçlü olarak karşılaştırılmıştır. Bütün modeller 10^7 epok boyunca eğitilmiştir, ve hepsi bu süre içinde doygunluğa ulaşmıştır. Referans PPO algoritması için Stable-Baselines3 [7]'ün standart hiperparametreli gerçekleştirimi kullanılmıştır. PPO'ya ödül fonksiyonu olarak

$$r(s) = \frac{0.5}{d_{ko}} + \frac{1}{d_{ro}}$$

verilmiştir, burada d_{ko} kavrayıcı ile obje arasındaki mesafe, d_{ro} ise robot tabanı arasındaki mesafedir. Bu yaygın ödül fonksiyonu yapısındaki düşünce PÖ ajanının önce anlaması daha kolay olan ilk terimi öğrenerek kavrayıcısını objeye yaklaştırmayı öğrenmesi, sonrasında da şans eseri objeye çarparak doğru yöne itmeyi öğrenmesidir.

Şekil 4'te 5 farklı deneyin sonuçları gösterilmiştir. Bütün deneyler 1000 episod boyunca tekrarlanıp tablolara ortalama değer ve standart sapma değeri yansıtılmıştır. Şekil 4a'da, her durum geçişinde $1/d_{ro}$ ödülü alan bir ortamda modellerin toplayabildikleri ödül miktarları verilmiştir. Sonuçlar PPO'nun kendi ödül terimini maksimize edemediğini, gösterimler üzerinden eğitilen bizim modelimizinse problemi kavramakta daha başarılı olduğunu göstermektedir. Şekil 4b'de objenin episod sonunda robota uzaklıkları verilmiştir. Bizim modelimizin hem daha yakına çekebildiği hem de çok daha düşük standart sapmaya sahip olduğu görülmektedir. Şekil 4c, 4d ve 4e'de objenin deney sonucunda objenin orijinal konumundan robota yaklaşma miktarları ve modellerin farklı başarı filtrelerini geçme yüzdeleri verilmiştir. Episodlar, obje şekil başlığında metre cinsinde belirtilen miktarlardan fazla robota yaklaştırılabiliyorsa başarılı sayılmıştır. Deneylerde başarı kriteri sıkılaştırıldıkça tüm modellerin başarı oranının beklendiği şekilde düştüğü, ancak tam modelin başarı oranının çok daha az azaldığı ve başarı değerlerinin daha yüksek kaldığı gözlemlenmektedir.

IV. Sonuç ve Gelecek Çalışmalar

Bu çalışmada gösterimlerden robotikte ters beceri öğrenme problemine uygun yeni bir ödül öğrenme ağı ve pekiştirmeli öğrenme sistemi önerilmiştir. Önerdiğimiz ödül öğrenme ağının uygunluğu ve sistemin performansı simülasyon ortamında test edilmiş, PPO yönteminin elle şekillendirilmiş ödülle gösterdiği performansla karşılaştırılmıştır. Sistemimizin saf PÖ ile öğrenilmekte zorlanılan becerileri, sadece ters beceri gösterimleri kullanarak öğrenebildiği gösterilmiştir. Sistemin zayıflıklarının biri ön eğitim performansına yüksek oranda bağımlı olmasıdır, gelecek çalışmalarda ön eğitim için geriye sarılmış davranış klonlamadan daha genelleştirilebilir ve gösterim performansına daha az bağımlı yöntemler denenenebilir, ayrıca tüm parçaların daha çeşitli deney ortamlarında daha zorlu düz-ters beceri çiftleri için incelenmeleri yapılabilir.

V. Bilgilendirme

Bu çalışma, AB tarafından finanse edilen INVERSE projesi (no. 101136067) tarafından desteklenmiştir.

Kaynaklar

[1] G. Dulac-Arnold, D. Mankowitz, and T. Hester, "Challenges of real-world reinforcement learning," 2019.

[2] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," 2018.

[3] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE ICRA*, 2018, pp. 6292–6299.

[4] K. Pertsch, Y. Lee, Y. Wu, and J. J. Lim, "Demonstration-guided reinforcement learning with learned skills," 2021.

[5] M. Akbulut, E. Oztop, M. Y. Seker, X. Hh, A. Tekden, and E. Ugur, "Acnmp: Skill transfer and task extrapolation through learning from demonstration and reinforcement learning via representation sharing," in *Conference on robot learning*. PMLR, 2021, pp. 1896–1907.

[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[7] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, pp. 1–8, 2021.