

Kaldırma Aksiyonuyla Oluşan Yörüngenin Uzun Kısa Dönem Hafıza Modeliyle Tahmini

Ahmet Ercan Tekden
Bilgisayar Mühendisliği Bölümü
Boğaziçi Üniversitesi
İstanbul, Türkiye
Email: ercan.tekden@boun.edu.tr

Emre Uğur
Bilgisayar Mühendisliği Bölümü
Boğaziçi Üniversitesi
İstanbul, Türkiye
Email: emre.ugur@boun.edu.tr

Özetçe —İnsanlar, gerçekleştirecekleri aksiyonları belirli bir amaç doğrultusunda belirler. Bir aksiyonu yaparken sadece anlık etkiyi değil, bu aksiyonun daha sonraki etkilerini de tahmin ederek hareketlerini planlarlar. Yapılan aksiyonların çevreye farklı türde etkileri olabilir. Bu etkileri sürekli veya ayrık etki olarak ikiye ayırabiliriz. Bu çalışmada kaldırma aksiyonunun sürekli bir etkisi olan yörüngenin tahmini üzerine çalışılmıştır. Bu amaçla bir Uzun Kısa Dönem Hafıza (UKDH) modeli geliştirilmiş ve bu model silindirik kapak ve baskılı devre kartı içeren deneylerde incelenmiştir. Silindirik kapak deneyinden UKDH modelinin yörünge tahmininde genelleme yapabildiği gözlemlenmiştir. Ayrıca cismin izlediği yörüngeye göre kaldırma noktasını tahmin eden bir başka UKDH modeli oluşturulmuş ve oluşturulan bu modelin tahminlerde başarılı olduğu görülmüştür. Sonrasında yapılan baskılı devre kartı deneyinde de gözlenen tahmin hatalarından, modelin eksiklikleri tespit edilmiş ve buna yönelik olası iyileştirmeler tartışılmıştır.

Anahtar Kelimeler—Uzun Kısa Dönem Hafıza, Yörünge Tahmini, Robot Öğrenmesi

I. GİRİŞ

Yapılan bir aksiyonun sonuçlarını tahmin etmek insanlar için önemlidir. Eğer bu sonuçlar tahmin edilebilirse çok adımlı planlar hayal edilebilir, bu planların sonuçları değerlendirilip, o an için bir aksiyon seçimi yapılabilir. Seçilen aksiyonun etkilerinin oluşmasında çeşitli unsurlar bulunmaktadır. Etkileşilen cismin özellikleri (ör. şekil ve büyüklük) ve cisim ile etkileşen aktörün yapabileceği hareketlere göre aksiyonun etkileri değişir. Bunu bir bakıma cismin ajana sağladığı aksiyonlar ve onların olası etkileri olarak düşünebiliriz. Kısacası cisimler etkileşen ajana farklı sağlanlıklar (affordance) sunar. Gibson sağlanlığı (affordance) ortamın organizmaya sağladığı iyi veya kötü imkanlar ve etkiler diye tanımlamıştır [1]. Bu tanıma robotlara uyguladığımızda sağlanlık, bir robotun çevresindeki cisimlere uygulayabileceği aksiyonlar, bu aksiyonların yapılaş biçimleri, aksiyonların etkileri ve cisimlerin birbiri arasındaki etkileşimi arasındaki bir ilişki olarak tanımlanabilir. Ortamın o anki durumu, robotun (aktörün) o anki durumu ve cismin o anki durumu sağlanlığa etki eder. Örnek olarak robotun cisme uzaklığı, cismin ulaşılabilirlik sağlanlığını; bir tahtanın bir vida ile tutturulması o tahtanın kaldırılabilme sağlanlığını; bir top için, topun bir masa üzerinde mi yoksa bir kutu içinde mi olduğu ise itme sağlanlığının etkisini değiştirebilir.

Bizim çalışmamız, kaldırma sağlanlığı dolayısıyla yapılan hareketin karmaşık şekillere sahip cisimlere etkisini tahmin Türkiye Robotbilim Konferansı, 2018

etmeyi amaçlamaktadır. Genellikle bu etki sürekli (continuous) etki ve ayrık (discrete) etki olmak üzere iki yönden incelenmektedir.

Ayrık etki daha çok etkinin anlamsal ifadesi olarak karşımıza çıkar. Masanın üzerindeki bir topun itilince (yularlanma sağlanlığı) masadan düşmesi, küçük bir cismin üst yüzeyi düz olan bir başka cismin üzerine konulabilmesi (istiflenebilirlik sağlanlığı) veya kendinden büyük ve içi boş olan bir kabın içine konulabilmesi (girdirilebilirlik sağlanlığı) tarzında olabilir[3].

Sürekli etki ise bir andaki durumu bilinen cismin bir sonraki an veya anlardaki durumlarının yani yörüngesinin (trajectory) tahmin edilmesi olarak düşünülebilir. Çevre koşulları, cismin hareket yörüngesinde farklılaşmalara sebep olabilir. Normalde bir topu ittiğimizde düz hareket ettiği halde, başka bir cisme çarptığında yön değiştirebilir veya pürüzlü bir yüzeyin üzerinden geçerse yavaşlayabilir. Robotun aksiyonu yapma tarzı cismin yörüngesini değiştirir. Bu topa anlık bir vuruş yapıp daha sonra ona dokunmaması veya sürekli bir itiş hareketi yapması da yörüngeyi etkiler. Top gibi simetrik bir cisimde nispeten şeklin önemi daha az hissedilebilir. Fakat cismin şekli karmaşıktıkça, robot ve ortam ile etkileşim dinamiği de karmaşıktır, dolayısıyla bu dinamiği tahmin etmek de zorlaşır. Bu durumda şekilsel özelliklerin de hesaba katılması gerekir. Mesela ittiğimiz cisim bir kutuysa, ittiğimiz noktaya göre açılabilir bir momentum da kazanacak, bu da yörüngeyi etkileyecektir.

Yukarıda da ifade edildiği gibi yörünge tahmini yapmak istenildiğinde cismin şeklinin veya şekil özelliklerinin, cismin bulunduğu ortamın ve yapılacak aksiyonun da hesaba katılması gerekir. Bizim genel motivasyonumuz karmaşık bir ortamda, karmaşık şekilli bir cismin karmaşık yörüngesini tahmin etmektir. Bu bildiride ise, cisimlerin şekillerini sabit tutarak etkileşim sonucunda elde edeceğimiz yörüngeyi tahmin etmeyi öğrenmeyi planladık. Bu öğrenme ve tahmin, birleştirilmiş iki tahmin sistemi kullanılarak gerçekleştirildi. İlk sistem cisimle etkileşim noktasını girdi alıp cismin hareket yörüngesini tahmin ederken, ikinci sistem gözlemlenen hareket yörüngesini girdi olarak alıp etkileşim noktasını tahmin eder ve ilk sistemin girdisi olan etkileşim noktasının değerlerinde iyileştirme yapar. İki tahmin sistemi de Uzun Kısa Dönem Hafıza (Long Short Term Memory) metodunu kullanmaktadır. Metodumuzun performansını değerlendirmek için deneylerde fizik tabanlı VREP simülatörü [2] kullanıldı. Simülatörde, geometrisi basit olan silindirik bir disk kullanılıp UKDH modelinin genelleme

yapabildiği ve altındaki sabit diske bağlı ve karmaşık bir geometriye sahip olan baskılı devre kartı (printed circuit board) kullanılarak modelin eksiklikleri tespit edilmiştir. Bu cisimleri farklı noktalarından kaldırarak elde ettiğimiz hareket yörüngelerini kullanarak oluşturduğumuz veri kümesi ile eğitilen UKDH modelinin tahmin ve genelleme performansı, K-En Yakın Komşu (K-EYK) metodu ile karşılaştırılmış ve UKDH modelinin kaldırma pozisyonuna göre daha genellenebilir yörüngeler çıkardığı gösterilmiştir.

II. BENZER ÇALIŞMALAR

Sağlarlıklar objenin üzerinde aksiyon yapabilecek aktöre kazandırdığı özelliklerdir. Bu özellikler, robotun çevresini öğrenmesinde, tasvir etmesinde, çeşitli tahminler yapmasında ve planlar yapmasında işine yarar[4]. Bunların bir kısmı aktör tarafından hareketin öğrenilmesi için de kullanılabilir. Robotun çevresiyle nasıl etkileşime girdiği ve çevreyi nasıl öğrendiği de incelenen problemlerdendir [5].

İnsanlar çevre değişimlerini tahmin etmek için içsel dinamik modeller kullanırlar [6]. Bunun haricinde sinirbiliminde de insanın yaptığı hareketin sonucunu o anki motor durumundan ve motor sinyallerinden tahmin etmesi incelenmiştir [7] Cisimlerin hareketi de aynı şekilde dinamik modellerle işlenebilir. Bunun için de simülasyon kullanılabilir. Her ne kadar öğrenme metodlarının yaptığı hata oranı simülasyonlardan daha yüksek olsa da, ortamın ve cisimlerin tam fiziksel modellerini kullanmaları gerektiği için gerçek zamanlı gerçekçi sensörlerle algılanan dünyada çalışması şu anda imkan dahilinde görünmemektedir.

Genelde farklı cisimler ve farklı aksiyonlar için farklı modeller kullanılmıştır. İtme aksiyonunun etkisi, aksiyondaki değişim, cismin durumundaki değişim ve cismin şeklindeki değişim parametrelerle ifade edilip modellenmiştir[8]. Topun hareketi [9] ve istiflenmiş küplerden oluşan kuledeki küplerin düşerkenki hareketi [10] de oluşturulan bazı benzer tahmin modellerindendir.

Son yıllarda derin öğrenme metodlarının performanslarının artmasıyla ve uçtan uca öğrenme (end-to-end learning) yaygınlaşmaktadır. Örnek olarak doğrudan ham resimden cisimlerin dinamiklerinin tahmini[11], ham nokta bulutundan (Point Cloud) aksiyon sonucunun tahmin edilmesi[12] farklı yapılan çalışmalardandır.

III. YÖNTEMLER

Cismin oluşacak yörüngesini hesaplamak istediğimizde alttaki özellikler dikkate alınmalıdır.

- 1) **Yapılan aksiyonun özellikleri:** Aksiyonun nereden yapıldığına ve nasıl yapıldığına göre cismin yörüngesi değişecektir. Burada, problemi basitleştirmek için hareketin yapılış tarzı sabitlenerek tek değişken hareketin nereden yapılacağı olarak bırakıldı.
- 2) **Şeklin özellikleri:** Etki edilecek cismin şekline ve hareketin uygulanacağı noktaya göre cismin kaldırılırken dayanak noktaları (zemine o an değmeye devam eden nokta), düşerken de ağırlık merkezi ve dayanak noktalarına göre düşeceği konum değişecektir.

- 3) **Ortam özellikleri:** Problemi basitleştirmek adına kaldırma hareketi sabit boş bir ortamda yapılarak ortamdaki bağımsız bir hale getirildi.

Bu koşullar dikkate alındığında, yörünge aşağıdaki gibi bir formülle ifade edilebilir.

$$X = f(P_{\text{kaldırma-noktası}}, x_0) \quad (1)$$

- $P_{\text{kaldırma-noktası}}$: Cismi kaldırdığı nokta. Amaç cismin üzerinde kaldırıp açma hareketi yapmak olduğu için, bu nokta, cismin kenar kısmına yakın olan noktalardan seçilmiştir.
- X : Cismin yörüngesindeki bütün pozisyonlar ve oryantasyonlar.
- x_i : Cismin i anındaki pozisyonu ve oryantasyonu, burası için x_0 0 anındaki pozisyona denk geliyor.

Amaç, kaldırma noktası, cismin başlangıç pozisyonu ve oryantasyonundan tüm yörüngeyi hesaplamaktır. Bunun için tahmin edilen pozisyon bilgisi bir sonraki pozisyonun tahmininde kullanılabilir, dolayısıyla öğrenme için tekrarlı bir model oluşturulabilir.

$$x_{i+1} = f(P_{\text{kaldırma-noktası}}, x_i) \quad (2)$$

Ancak bu model eksik kalır, çünkü cisim hareket uygulanmaya başlandığında, cismin bazı içsel parametreleri de değişmeye başlar. Mesela cisim hız kazanabilir. Bu hız değişimleri de tahmin edilecek pozisyonlarda değişime neden olur. Dolayısıyla bu içsel parametrelerin de modele dahil edilmesi gerekir.

$$(x_{i+1}, h_{i+1}) = f(P_{\text{kaldırma-noktası}}, x_i, h_i) \quad (3)$$

- h_i : İçsel parametreler.

Yukarıda bahsedilen sebepler dahilinde, biz özel bir Yineleyen Sinir Ağları (RNN) türü olan Uzun Kısa Dönem Hafıza (UKDH) ağlarını kullandık.

A. Uzun Kısa Dönem Hafıza Ağı

UKDH ağlarını anlamak için önce RNN'leri anlamak gerekir. RNNler kendi içinde döngü bulunduran ve kendi içsel hafızası olan bir Yapar Sinir Ağı (ANN) türüdür. Bu ağ türü, önceki basamaklardaki girdilerden hesaplanan gizli parametreleri içsel hafızasında tutarak sonraki basamaklarda bu bilgiyi kullanabilmektedir. UKDH ağları ise üç yeni birim kullanarak RNN'den farklıdır:

- 1) **Girdi Kapısı (Input Gate):** O anki girdilere göre bilgilerin hangilerinin hafızaya kaydedileceğine karar veren sinir kapısı.
- 2) **Çıktı Kapısı (Output Gate):** Hafızadaki hangi bilgilerin çıktı hesaplanacağı zaman kullanılacağına karar veren sinir kapısı.

- 3) **Unutma Kapısı** (Forget Gate): Hafızadaki hangi bilgilerin o anda unutulması gerektiğine karar veren kapı.

Denklem (3)'teki f fonksiyonunun öğrenilebilmesi için Şekil 1'deki UKDH ağı kullanılmıştır.

Bu model, cismin kaldırma pozisyonuna göre yörüngeyi tahmin etmekte ve dolayısıyla kaldırma pozisyonunun değerlerinin doğru olduğunu kabul etmektedir. Fakat görsel algıdaki hatalar nedeniyle kaldırma pozisyonu yanlış algılanabilir, yada hiç bilinmeyebilir. Robot, kaldırma pozisyonunu bilmesede kaldırma sırasında gözlemlediği yörüngeden pozisyonu tahmin edebilmesi ve böylece yanlışını düzeltebilmesi gerekir. Şekil 2'deki model de, yörüngeyi girdi olarak alıp kaldırma noktasını çıktı olarak veren farklı bir UKDH ağıdır.

Bu noktada biz şu ana kadar gördüğümüz noktalardan ve tahmin ettiğimiz kaldırma noktasından yörüngeyi hesaplayabiliriz. Bu iki modeli birleştirilerek oluşturulan Şekil 3'teki genel entegre model hem yörüngeyi tahmin etmek hem de kaldırma noktası tahminini güncellemek için kullanılabilir.

Bu durumda genel iş akışını şöyle verebiliriz.

- 1) Elimizdeki kaldırma noktalarından olası yörüngeleri hesapla.
- 2) Kaldırma hareketine başla.
- 3) Cismin gözlemlediğimiz pozisyonlarından kaldırma yapmış olabileceğimiz noktaları bul.
- 4) Yörüngeyi geri hesaplamayla düzelt.

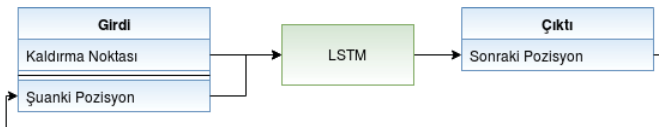
IV. DENEY DÜZENİĞİ

Bu bildiride, deneyler fizik tabanlı VREP simülöründe yapıldı ve robotun kaldırma hareketi cismin ilgili noktasına, dikey yönlü yapay bir kuvvet uygulanarak modellendi.

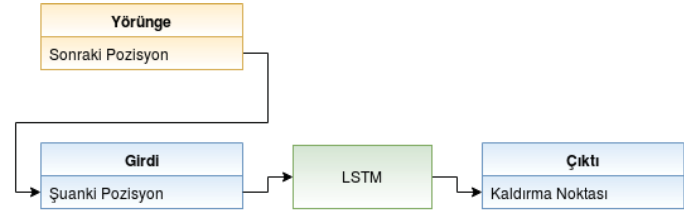
İlk deney VREP içinde bir silindir kapak olan boş bir ortamda yapıldı. Silindir, 10cm yarı çapa ve 2 cm yüksekliğe sahip olacak şekilde oluşturuldu. Zaman aralıkları 10 milisaniye ve fizik motoru da ODE (Open Dynamics Engine) olarak belirlendi.

Uygulanacak kuvvet miktarı deneysel olarak deneme yanılma ile bulundu. Cismin kenar noktalarından dikey kuvvet uygulayarak deneylere başlandı. Her bir deney için cismin 40 ms zaman aralıklarıyla pozisyon ve oryantasyonu, yani pozu (pose) kaydedildi. Cismin nereden kaldırıldığı ve cismin kaydedilen pozlarından veri kümesi oluştu. Bu veri kümesinde silindir 100 farklı noktasından kaldırıldı. Kaldırma noktaları cismin kenarından tekdüze dağılımlı seçildi.

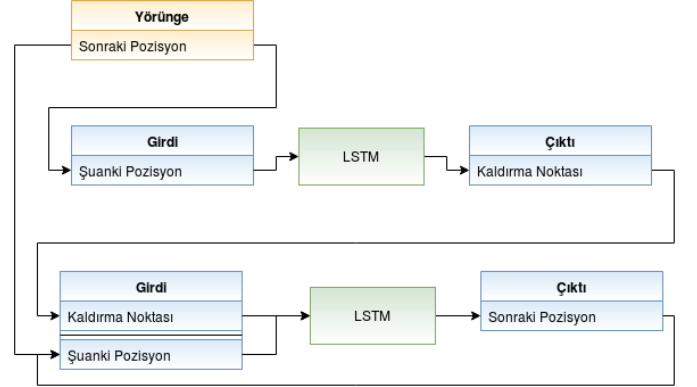
Öğrenme için içerisinde hazır UKDH dizaynı olan KERAS yazılım programlama arayüzü (<https://keras.io>) ve arkaplan motoru olarak Tensorflow Kütüphanesi kullanıldı.



Şekil 1. Yörünge Hesaplamak için kullanılan UKDH ağı



Şekil 2. Kaldırma noktasını hesaplamak için kullanılan UKDH ağı. t anında ilk t adımı gözlemlenen pozisyonlardan kaldırma noktası bulunuyor.



Şekil 3. 1. ve 2. model birleştirilerek elde edilen yeni model. Burada t anı için, ilk t adımı bulunan kaldırma noktası ve t anına kadar gözlemlenen pozisyonlardan yörüngeyi çıkarabiliyor.

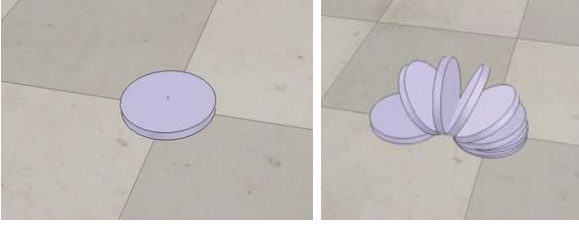
UKDH bazlı öğrenmenin ve yörünge tahmininin ne kadar genellenebilir olduğunu ölçmek için öğrenme ve test kümeleri mantıksal olarak ayrıldı: öğrenme ve test kaldırma noktalarının silindirin farklı bölgelerinden olmasına dikkat edildi. Öğrenme ve test kümesi ayrımı %75-%25 şeklinde yapıldı ve Şekil 5'te gösterildi.

Öğrenme yaparken yörüngeler modele birer birer verilerek öğrenme gerçekleşti. Modele öğrenme setindeki her yörünge 80 defa verildi. Bir yörüngenin üzerinde daha fazla önyargı (bias) oluşmaması için her seferinde tekrar rastgele karışmış öğrenme noktaları verildi. UKDH ağı öğrenirken fark kümeleriyle daha iyi sonuç verdiği için her bir adımda, veriyi modele verirken, pozların önceki adımdaki poza ilişkin farkı (relative difference) verildi. Bu her adımda hem pozisyon hem oryantasyon için yapıldı.

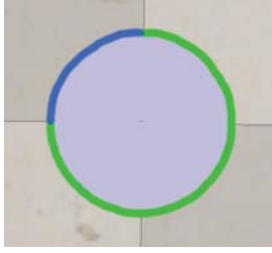
$$X_i = (\Delta x_i, \Delta y_i, \Delta z_i, \Delta \gamma_i, \Delta \beta_i, \Delta \alpha_i, P_x, P_y) \quad (4)$$

$$Y_i = (\Delta x_{i+1}, \Delta y_{i+1}, \Delta z_{i+1}, \Delta \gamma_{i+1}, \Delta \beta_{i+1}, \Delta \alpha_{i+1}) \quad (5)$$

- X_i : i. anda modelin girdisi.
- Y_i : i. anda modelin çıktısı.
- $\Delta \gamma$: x eksenindeki açı farkı(roll)
- $\Delta \beta$: y eksenindeki açı farkı(pitch)
- $\Delta \alpha$: z eksenindeki açı farkı(yaw)
- P : Kaldırma noktası



Şekil 4. V-Rep Ortamı: 10 cm yarıçaplı, 2 cm yükseklikli silindir, 10 ms zaman aralıkları ve ODE fizik motoru.



Şekil 5. Öğrenme ve test kümeleri. Öğrenme kümesindeki noktalar yeşil bölgeden, test kümesindeki noktalar mavi bölgeden alınmıştır.

Buradan UKDH ağı ile yörüngedeki noktaların farkını hesaplanabilir. UKDH ağında 50 nöron kullanıldı aynı zamanda aşırı uymayı (overfitting) engellemek için düşme (dropout) tabakası kullanıldı. Buradan hesaplanan poz farklarından rotasyon matrisleri oluşturuldu ve bu rotasyon matrisleri kullanılarak her adım için yörüngede x, y, z ekseninde değerler bulundu.

Kaldırma noktasını hesaplayan modele de girdi olarak da poz bilgisi verildi çıktı olarak da kaldırma noktasını alındı. Bu modelde de aynı şekilde 50 nöron ve düşme tabakası kullanıldı.

$$X_i = (\Delta x_i, \Delta y_i, \Delta z_i, \Delta \gamma_i, \Delta \beta_i, \Delta \alpha_i) \quad (6)$$

$$Y_i = (P_x, P_y) \quad (7)$$

Bu iki modeli aynı anda çalıştırarak da birinin çıktısı diğerinin girdisi olacak şekilde kullanılıp yörünge geri tahmini yapıldı.

V. DENEY SONUÇLAR

A. Yörünge Tahmin Modeli

Silindirde öğrenme yaparken kullanılmayan kaldırma noktalarından oluşturulan yörüngeleri tahmin eden UKDH modelinin (ve K-EYK modelinin) deney sonucu Şekil 6'daki gibidir. İlk iki satırda K-EYK'dan aldığımız sonuçlar, diğer iki satırda UKDH ağından aldığımız sonuçları gösterilmiştir. Tahmin de edilebileceği gibi K-EYK test ettiğimiz noktalara yakın öğrenme verisi olmadığı için yörüngeyi doğru tahmin etmekte zorlanmakta, ama UKDH ağı daha genel bir model öğrendiği için daha iyi tahminler yapabilmektedir.

Modelimizden gelen genel hata oranı Şekil 7'de verilmiştir. Bu hatanın değerlendirilebilmesi için taban model olarak seçtiğimiz 1-EYK'yı ve genel hareket miktarının da

çıkarılabilmesi için elimizdeki veri setinden tamamen rastgele gelen bir yörünge seçildi. Görüldüğü gibi, daha önce görmediği kaldırma noktalarından 1-EYK'nın yörünge tahmin performansı UKDH ağına göre daha düşük kalmaktadır. Bu şekilde çizgiler hata ortalamasını, gölgeler ise hata varyansını göstermektedir.

B. Kaldırma Noktası Tahmin Modeli

Kaldırma noktası tahmin etmekle sorumlu modelin bulduğu noktalar örnek verilen yörünge için (genel olarak bulunan noktalar diğer yörüngeler için buna benzer çıkmaktadır) Şekil 8'deki 1. sütunda olduğu gibidir. Görüldüğü üzere bu noktalar kaldırma noktasına yön olarak yakın olmakla birlikte pozisyon olarak farklı yerdedir. Burada kısıtlayıcı koşul olarak tüm tahmin edilen kaldırma noktalarını silindirin kenarına getirme koşulu koyulabilir. Cismin, kenarından kaldırdığı kesin olarak biliniyor varsayıp, UKDH ağına model 2'de tahmin ettiği sonuç noktaları, cismin kenarına taşınabilir. Bunun için bulunan kaldırma pozisyonun x, y değerlerinden açı bulup, açı değerini kullanarak kaldırma pozisyonunu kenarda bir yer olacak şekilde güncelledik. Bu güncelleme yapıldığında, yeni pozisyon tahminleri Şekil 9'daki ikinci sütundaki gibi oldu. Ayrıca sayısal olarak bu yaptığımız işlemin ikinci modelin performansını ne kadar etkilediği de Şekil 9'da görülebilir.

C. Birleştirilmiş Modeli

Test kümesindeki kaldırma noktalarından kaldırma hareketinin başlandığı varsayılarak o yörünge üzerindeki hareket başlatılıyor. Bu yörünge üzerinde hareketi izledikçe modelin tahmin ettiği kaldırma noktalarından ve o ana kadar gözlemlenen yörüngedeki pozlardan yeni yörüngeler tahmin edilir. Gözlenen poz sayısına göre zamanla hata düzeltilir, buradaki kaldırma noktası hatasının düzeltilmesiyle elde edilen kazanç da Şekil 10'da verilmiştir.

D. Karmaşık şekil: baskılı devre kartı

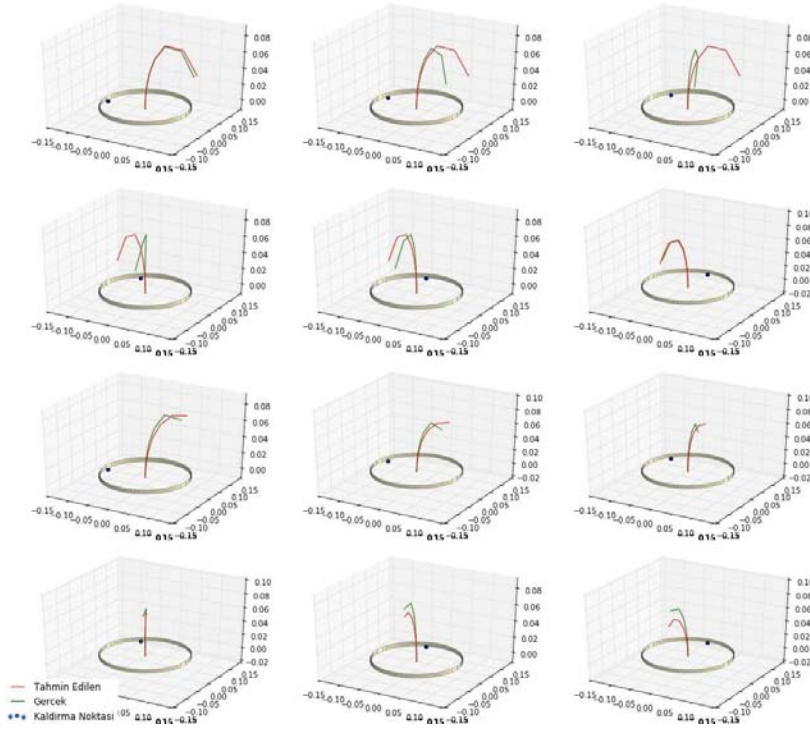
Bu deneyde sabit bir diskin üzerindeki baskılı devre kartının diskten kaldırılırken gözlemlenmesi beklenen hareket yörüngesi öğrenildi ve test edildi. Bu cisim önceki deneydeki silindir kapakla kıyasladığımızda daha karmaşık olduğu için farklı noktalardan kaldırıldığında öğrenilmesi ve tahmin etmesi daha zor hareket yörüngeleri üretecektir. Yörünge tahmin eden modelimizin tahminlerini Şekil 12'de bulabilirsiniz.

Burada bizim tahminimiz kaldırma hareketi yapılırken, cismin şekline yönelik özellikleri modele sunmadığımız için yeterince iyi öğrenme yapamıyor. Burada şekil bilgisine yönelik özellikler de verince daha iyi sonuç alabileceğimizi düşünüyoruz.

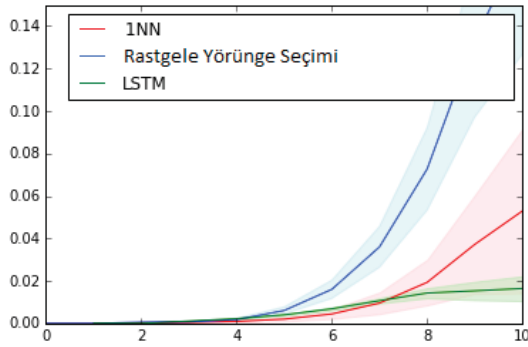
VI. SONUÇ

Bu çalışmada, UKDH modeliyle yörünge tahmini yapılabileceğini gösterdik. Her ne kadar bu çalışma kaldırma hareketi ile sınırlansada, üç boyutlu bir hareketi tahmin edebilmesinden ötürü farklı aksiyonlarda da bu modelin iyi sonuç verebileceğini düşünüyoruz.

İleri çalışma olaraksa amacımız modele cismin şeklini ve robotun hareketini de parametre olarak verebilecek şekilde



Şekil 6. Birinci modelden gelen sonuçlar. 1 ve 2.Satır 1-EYK, 3 ve 4.Satır da UKDH açısından bulunan yörüngeler. Bu 2 farklı modelde sırasıyla aynı test yörüngesi kullanıldı ve aynı sırayla şekile eklendi.

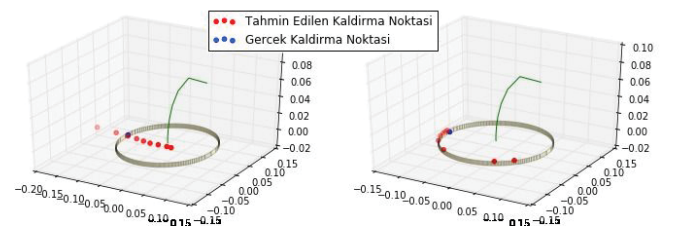


Şekil 7. UKDH, K-EYK ve rastgele yörünge seçiminin karşılaştırılması. Pozisyon değerlerinin hatalarının ortalama kare kökü(RMSE) kullanıldı.

modeli genişletmek. Bunun dışında farklı cisimlerle yapabileceği etkileşim ve birbiri ardına yapılabilecek aksiyonlar çalışılabilecek alanlar arasındadır.

VII. TEŞEKKÜR

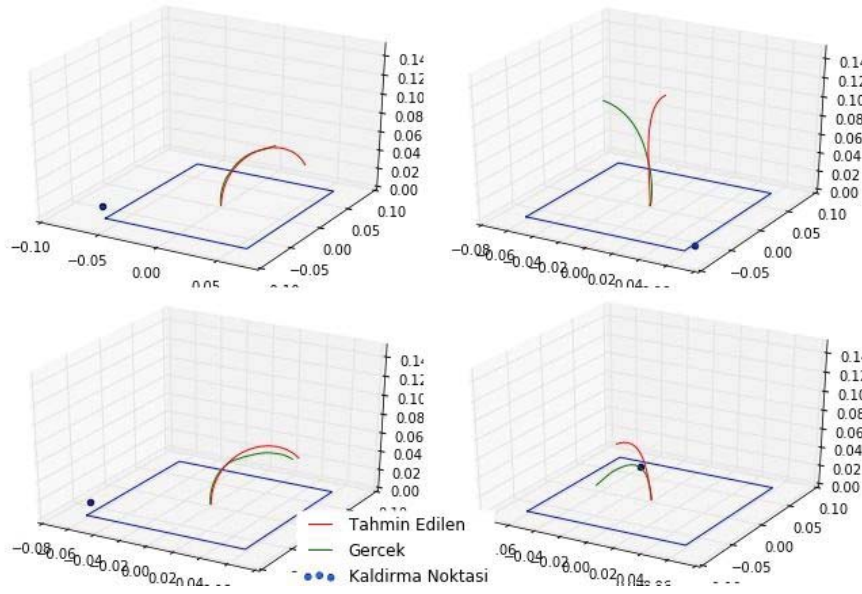
Bu çalışma TÜBİTAK BİDEB 2232 Yurda Dönüş Araştırma Burs Programı'nın 117C016 numaralı projesi, Avrupa Komisyonu Ufuk 2020 Programı'nın 731761 numaralı IMAGINE projesi ve Bogazici Araştırma Fonu Startup projesi 12022 tarafından kısmi olarak desteklenmiştir.



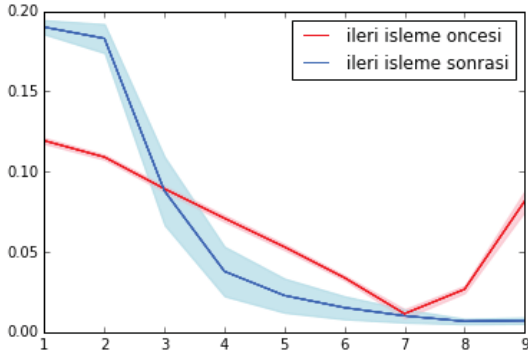
Şekil 8. İkinci modelde ileri işleme yapmadan önceki ve sonraki bulunan noktalar.

KAYNAKÇA

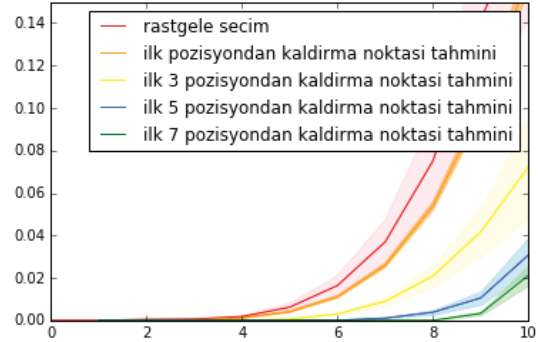
- [1] Gibson, James J. "The theory of affordances." The people, place, and space reader (1979): 56-60.
- [2] M. F. E. Rohmer, S. P. N. Singh, "V-rep: a versatile and scalable robot simulation framework," in Proc. of The International Conference on Intelligent Robots and Systems (IROS), 2013.
- [3] E. Ugur and J. Piater, "Bottom-up learning of object categories, action effects and logical rules: From continuous manipulative exploration to symbolic planning," 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, 2015, pp. 2627-2633. doi: 10.1109/ICRA.2015.7139553
- [4] L. Jamone et al., "Affordances in psychology, neuroscience and robotics: a survey," in IEEE Transactions on Cognitive and Developmental Systems, vol. PP, no. 99, pp. 1-1.
- [5] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. Sukhatme, "Interactive perception: Leveraging action in perception and perception in action," arXiv preprint arXiv:1604.03670, 2016. doi: 10.1109/TCDS.2016.2594134
- [6] Hamrick, Jessica, Battaglia, Peter, and Tenenbaum, Joshua B. Internal



Şekil 9. Baskılı devre kartı kaldırma tahminlerinden gelen bazı sonuçlar. Burada doğru tahmin ettiği, değişim noktasını kaçırdığı, hedefi aştığı(overshoot), z yönündeki yükseklik değişimini tutturamadığı durumlar var.



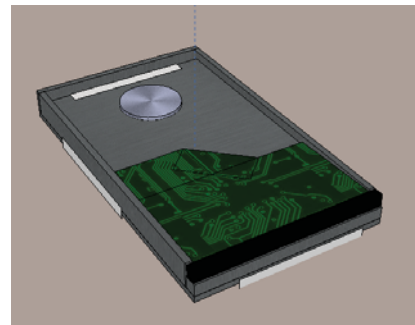
Şekil 10. İkinci modelde ileri işleme yapmadan önceki ve sonraki bulunan noktaların hatalarının ortalama kare kökü.



Şekil 11. Modellerin beraber işlendiğinde verdiği hatalarının ortalama kare kökü. Burada modelin ilk 1,3,5,7 pozü gördükten sonra tahmin ettiği kaldırma noktasından geri tahmin ettiği yörüngelerdeki hata miktarı görülebilir.

physics models guide probabilistic judgments about object dynamics. In Proceedings of the 33rd annual conference of the cognitive science society, pp. 1545–1550. Cognitive Science Society Austin, TX, 2011.

- [7] Miall, R., & Wolpert, D. (1996). Forward models for physiological motor control. *Neural Networks*, 9(8), 1265–1279.
- [8] Kopicki, M., Zurek, S., Stolkin, R. et al. *Auton Robot* (2017) 41: 1061. <https://doi.org/10.1007/s10514-016-9571-3>
- [9] K. Fragkiadaki, P. Agrawal, S. Levine, and J. Malik, “Learning visual predictive models of physics for playing billiards,” *arXiv preprint arXiv:1511.07404*, 2015.
- [10] . Lerer, S. Gross, and R. Fergus, “Learning physical intuition of block towers by example,” *arXiv preprint arXiv:1603.01312*, 2016.
- [11] . Mottaghi, H. Bagherinezhad, M. Rastegari, and A. Farhadi, “Newtonian scene understanding: Unfolding the dynamics of objects in static images,” in *CVPR*, 2016, pp. 3521–3529.
- [12] A. Byravan and D. Fox, “SE3-nets: Learning rigid body motion using deep neural networks,” *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, pp. 173–180. doi: 10.1109/ICRA.2017.7989023



Şekil 12. VREP’teki sabit disk modeli