



## Comparison of push-recovery control methods for Robotics-OP2 using ankle strategy

Emrah Aslan<sup>1\*</sup>, Muhammet Ali Arserim<sup>2</sup>, Ayşegül Uçar<sup>3</sup>

<sup>1</sup>Silvan Vocational School, Dicle University, Diyarbakır, Türkiye

<sup>2</sup>Faculty of Engineering, Dicle University, Diyarbakır, Türkiye

<sup>3</sup>Faculty of Engineering, Fırat University, Elazığ, Türkiye

### Highlights:

- Three different controllers have been designed for push-recovery.
- Controllers have been validated using Robotis-OP2 with twenty degrees of freedom.
- DQN gave the best results.

### Keywords:

- Push-Recovery
- Robotis-OP2
- Deep Q Network
- Model Predictive Control
- PID Control

### Article Info:

Research Article

Received: 13.09.2023

Accepted: 09.12.2023

### DOI:

10.17341/gazimmfd.1359434

### Correspondence:

Author: Emrah Aslan

e-mail:

emrah.aslan@dicle.edu.tr

phone: +90 553 493 4868

### Graphical/Tabular Abstract

The main objective of this research is to develop push-recovery control systems for bipedal humanoid robots. Humanoid robots may experience balance problems against external pushes. Robots, which do not have a flexible structure like humans, are unstable against external forces caused by applied pushes, ground and real world conditions. There are some difficulties in modelling bipedal humanoid robots. One of them is that they are systems with high degrees of freedom. As the degree of freedom increases, the number of actuators to be controlled increases. Another main difficulty is the variable robot kinematics. Depending on the movement of the humanoid robot, its kinematics constantly change. In this paper, we propose control strategies to solve the stability problems in humanoid robots. Our goal is to enable bipedal robots with human-like behaviour to return to the equilibrium position against external pushes. This study was performed in simulation and real environment. In practice, control systems have been developed that allow the humanoid robot, which has 20 degrees of freedom, to remain in balance against the external push. The humanoid robot Robotis-OP2 was used in the application phase of the study. Different control methods and algorithms have been proposed to bring the humanoid robot back to the balance position as a result of external pushes. Push-recovery controllers are designed by using PID controller as the classical control method, MPC as a predictive control method and DQN algorithm for deep reinforcement learning control. Applications were performed in the Webots robot simulator environment. In the study, applications were carried out with reference to the ankle strategy as a push-recovery controller. After the random pushing forces applied from the front and rear, the robot was brought to the balance point. As a result of the three methods applied in the study, the robot was able to reach the equilibrium position as a result of the external thrust applied to the humanoid robot. The application results are given in Table A. PD control was the method with the lowest reward average and success performance. MPC control method gave better results than PD control. However, MPC seems to be unsuccessful compared to the DQN algorithm. In this study, it was observed that DQN, which is a deep reinforcement learning algorithm, gave the best results.

Table A. Comparison of algorithm results

Control Method	Total Episode	Approximate of Success	Episode	Biggest Reward	Total Rewards Won	Average Reward
PD	100	100		7,5	324,265	3,24
MPC	100	100		7,5	347,068	3,47
DQN	7500	7200		7.5	37483,51	4,99

### Purpose:

The main purpose of this study is to develop push-recovery controllers for bipedal humanoid robots

### Theory and Methods:

Three different techniques of control were utilized in the applications. These methods are as follows; Classical control method is PID, Model Predictive Control (MPC) based on prediction and Deep Q Network (DQN) as deep reinforcement learning algorithm.

### Results:

The humanoid robot was tested with three methods and the results were compared. It has been determined that the Deep Q Network algorithm gives the best results among these methods.

### Conclusion:

It has been observed that the humanoid robot has obtained successful results in the tests in the three methods applied. Among these methods, the lowest success rate was PID. The PID method was followed by the MPC. It has been found that more successful results are obtained with DQN.



## Ayak bileği stratejisi kullanarak Robotis-OP2 için itme kurtarma kontrol yöntemlerinin karşılaştırılması

Emrah Aslan<sup>1\*</sup>, Muhammet Ali Arserim<sup>2</sup>, Ayşegül Uçar<sup>3</sup>

<sup>1</sup>Dicle Üniversitesi, Silvan Meslek Yüksekokulu, Diyarbakır, Türkiye

<sup>2</sup>Dicle Üniversitesi, Mühendislik Fakültesi, Diyarbakır, Türkiye

<sup>3</sup>Fırat Üniversitesi, Mühendislik Fakültesi, Elazığ, Türkiye

### Ö N E Ç İ K A N L A R

- İtme-kurtarma için üç farklı denetleyici tasarlanmıştır
- Kontrolörler, yirmi serbestlik derecesine sahip Robotis-OP2 kullanılarak doğrulanmıştır
- DQN en iyi sonucu vermiştir

### Makale Bilgileri

Araştırma Makalesi

Geliş: 13.09.2023

Kabul: 09.12.2023

DOI:

10.17341/gazimmfd.1359434

### Anahtar Kelimeler:

İtme-kurtarma,  
robotis-OP2,  
derin Q ağı,  
model öngörülü kontrol,  
PD

### ÖZ

Bu çalışmanın ana hedefi, iki ayaklı insansı robotlar için itme-kurtarma kontrolörleri geliştirmektir. İki ayaklı insansı robotlarda dışardan gelen itmelere karşı denge problemi oluşmaktadır. Bu makalede insansı robotlardaki denge problemlerine çözüm olacak kontrol yöntemleri önerilmiştir. Amacımız, insan gibi davranan iki ayaklı robotların dışardan gelen itmelere karşı denge pozisyonuna gelebilmesini sağlamaktır. İnsanlar dışarda gelen itmeler sonucunda denge problemleri ile karşılaştıklarında oldukça başarılı bir şekilde tepki vermektedirler. İki ayaklı insansı robotlarda ise bu yetenek sınırlıdır. Bunun başlıca sebebi insansı robotların karmaşık yapıları ve kapasitelerinin sınırlı olmasıdır. Gerçek dünyada insanların denge bozukluğu durumunda gösterdikleri tepkiler ele alınarak oluşturulan itme-kurtarma stratejileri bulunmaktadır. Bu stratejiler; ayak bileği, kalça ve adım stratejileridir. Bu çalışmada itme-kurtarma stratejilerinden ayak bileği stratejisini kullanılmıştır. Ayak bileği stratejisini kullanarak, farklı kontrol yöntemleri denenmiştir. Yapılan uygulamalarda üç farklı kontrol yöntemi kullanılmıştır. Bunlar; klasik kontrol yöntemi PD, tahmine dayalı olarak Model Öngörülü Kontrol (MÖK) ve derin pekiştirmeli öğrenme algoritmalarından Derin Q Ağı (DQA) yöntemleridir. Uygulamalar Robotis-OP2 insansı robotu üzerinde gerçekleştirilmiştir. Simülasyon testleri ise Webots ortamında 3 boyutlu olarak yapılmıştır. Her bir yöntem ile insansı robot test edilmiş ve sonuçları karşılaştırılmıştır. Bu yöntemlerden derin pekiştirmeli öğrenme algoritması olan Derin Q Ağı (DQA) en iyi sonuç verdiği gözlemlenmiştir.

## Comparison of push-recovery control methods for robotics-op2 using ankle strategy

### H I G H L I G H T S

- Three different controllers have been designed for push-recovery
- Controllers have been validated using Robotis-OP2 with twenty degrees of freedom
- DQN gave the best results

### Article Info

Research Article

Received: 13.09.2023

Accepted: 09.12.2023

DOI:

10.17341/gazimmfd.1359434

### Keywords:

Push-recovery,  
robotis-OP2,  
deep Q network,  
model predictive control,  
PD

### ABSTRACT

The main purpose of this study is to develop push-recovery controllers for bipedal humanoid robots. In bipedal humanoid robots, occur balance problems against external pushes. In this article, control methods that will be the solution to the balance problems in humanoid robots are proposed. We aim to ensure that bipedal robots that behave like humans can come to a position of balance against external pushes. When people encounter balance problems as a result of outside pushes, they respond quite successfully. This ability is limited in bipedal humanoid robots. The main reason for this is the complex structures and limited capacities of humanoid robots. In the real world, there are push-recovery strategies created by considering the reactions of people in case of balance disorder. These strategies; are ankle, hip, and step strategies. In this study, the ankle strategy, from the push-recovery strategies, was used. Different control methods have been tried with the ankle strategy. Three different techniques of control were utilized in the applications. These methods are as follows; Classical control method is PD, Model Predictive Control (MPC) based on prediction, and Deep Q Network (DQN) as deep reinforcement learning algorithm. The applications were carried out on the Robotis-OP2 robot. Simulation tests were done in 3D in the Webots simulator. The humanoid robot was tested with three methods and the results were compared. It has been determined that the Deep Q Network algorithm gives the best results among these methods.

## 1. Giriş (Introduction)

Robot fikri insanlık tarihinde çok eskiye dayanmaktadır. Günümüzdeki elektrik enerjisi ile çalışan robotlar gibi olmasa da M.Ö. yapılan mekanik otomat robotlara rastlamak mümkündür. Robot teknolojisinin temel amacı insan hayatını kolaylaştırma çabasıdır. Teknolojide yaşanan gelişmelere paralel olarak robotik alanda yapılan çalışmalarda hız kazanmıştır. Robotik çalışmalarında hedeflenen durum insan gücüne gerek kalmadan verilen görevi yerine getiren ve düşünebilen robotlar üretmektir. Fakat mevcut durumda robotlar bazı insan hareketlerini taklit etme ve insan gibi düşünme konusunda yetersiz kalmaktadır [1, 2].

Son yıllarda yapay zeka alanındaki gelişmeler ve teknolojideki üstün özelliklere sahip donanımlar sayesinde insan hareketlerini taklit etmede daha başarılı robotlar geliştirilmiştir. Bu robotlar, savunma sanayi, mühendislik uygulamaları, endüstri, tıp, psikoloji gibi birçok alanda kullanılmaktadır [3].

Araştırmacılar son yıllarda insansı robot çalışmalarına ilgi göstermektedir. İnsansı robotlar temel olarak iki grupta incelenmektedir. Bunlardan birincisi, iki ayaklı insansı robotlardır. İkincisi ise gezgin insansı robotlardır. İki ayaklı insansı robotlar, insan gibi bacaklar, kollar, gövde ve baş kısımlarından oluşmaktadır. Bu robotlarda genellikle yürüme, koşma gibi insan hareketlerini taklit etme üzerine araştırmalar yapılmaktadır. Gezgin robotlar ise üst gövde kısmı insana benzeyen alt kısmı ise tekerlekli ya da paletli olan robotlardır. Gezgin robotlar genellikle mobil işlemler ve servis yapma gibi işlerde kullanılmaktadır [4].

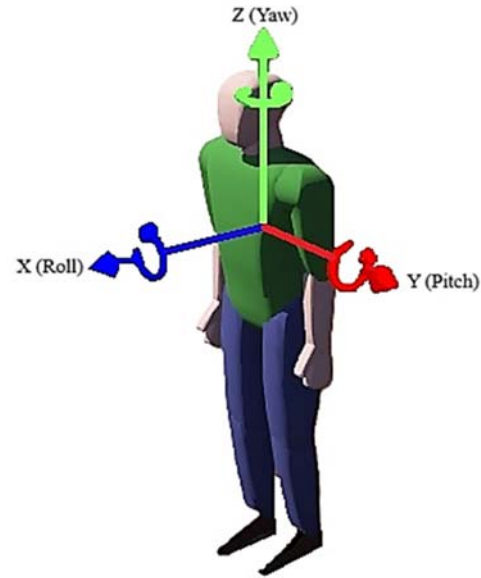
İki ayaklı robotlar gerçek insana benzer bir yapıda olduğundan engellerden geçme hareket etme gibi durumlarda tekerlekli ve paletli robotlara karşı daha avantajlıdır. Fakat iki ayaklı insansı robotların geleneksel tekerlekli ve paletli robotlara karşı dezavantajlı olduğu durumlarda bulunmaktadır. En önemli dezavantajı olarak denge problemi karşımıza çıkmaktadır. Denge problemi iki ayaklı insansı robotlarda ciddi bir problemdir [5]. İnsan gibi esnek bir yapıya sahip olmayan robotlar uygulanan itmelerde, zemin ve gerçek dünya şartlarından kaynaklı dış kuvvetlere karşı kararsızdırlar. Bu problemlere karşı geleneksel robotlar, iki ayaklı insansı robotlara göre daha kararlıdırlar [6, 7].

İki ayaklı insansı robotların modellenmesinde genel olarak temel bazı zorluklar bulunmaktadır. Bunlardan biri yüksek serbestlik derecelerine sahip sistemler olmalarıdır. Serbestlik derecesi arttıkça kontrol edilmesi gereken aktüatör sayısı artmaktadır. Bir diğer temel zorluk ise değişken yapıda olan robot kinematiğidir. İnsansı robotun hareketine bağlı olarak kinematiği sürekli değişir. Robot tek destek ve çift destek noktasında olduğu zaman denge noktası değişmektedir. Robotun destek noktası tespit edilip ona uygun bir denge pozisyonuna geçmesi sağlanmalıdır [8]. Bir diğer zorluk ise dış kuvvetlerdir. Bu zorluk robotun kendisinden kaynaklı olmadığı için ne zaman ve nereden geldiği belli olmadığından kontrol etmesi daha zordur. Araştırmacılar insansı robot çalışmalarında, insan gibi hareket eden ve kontrolünü gerçekleştirebilen robotların yapılmasını hedeflemektedirler. Bundan dolayı, insansı robotlar üzerinde dış itmelere karşı kurtarma üzerine çalışmalar yapılmaktadır. İnsansı robotlar dışardan aldıkları darbeler dışında ortamdaki kaynaklı denge bozuklukları da itme olarak tanımlanmaktadır [8]. İnsansı robotlara itmelere doğacak denge bozukluğunun önüne geçmek için itme geri kazanım kontrolü yeteneği kazandırılmalıdır. Literatürde insansı robotlardaki bu denge problemi iki ayaklı hareket olarak adlandırılmaktadır. İnsansı robotlardaki bu denge problemini çözmek için farklı kontrol yöntemleri önerilmiştir. Geleneksel kontrol,

tahmine dayalı kontrol, bulanık kontrol ve derin pekiştirmeli öğrenme kontrol yöntemleri bunlardan bazılarıdır [9].

İnsansı robotlarda denge alt ve üst vücut hareketleri ile gerçekleştirilmektedir. Alt vücut hareketleri yürüme sorunlarını, üst vücut hareketleri ise kol ve vücut hareketlerini kontrol ederek daha dengeli olmasını sağlamaktadır. Alt vücut hareketleri üç temel stratejiden oluşmaktadır. Bunlar; ayak bileği stratejisi, kalça stratejisi ve adım stratejisidir. Bu stratejiler gerçek hayatta insanların dengede durabilmek için verdiği tepkilerin matematiksel ifadesidir [10].

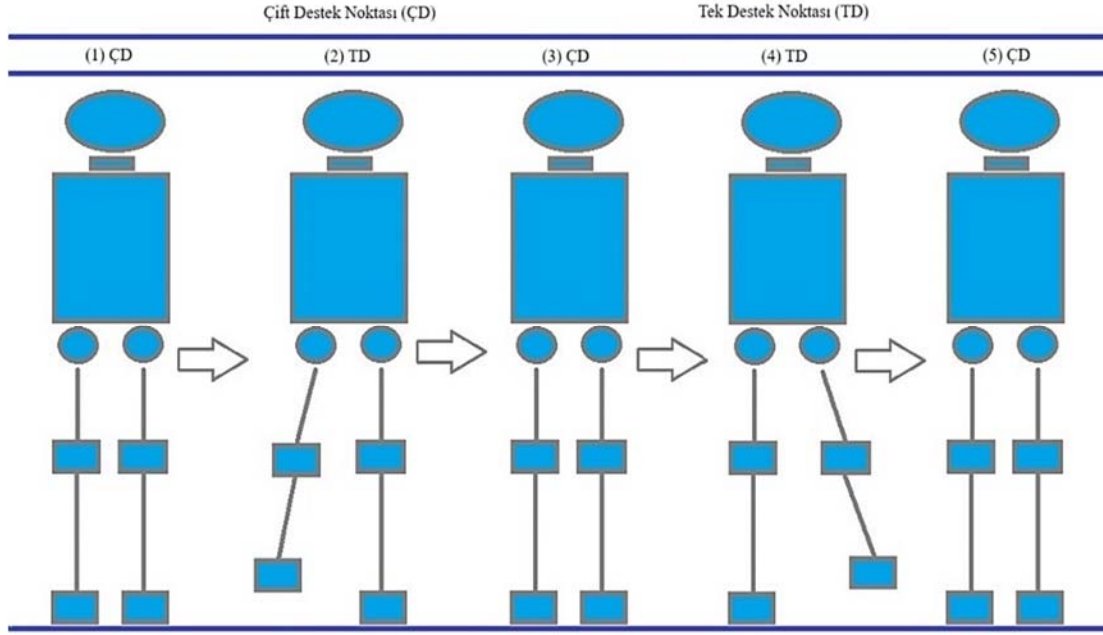
İnsansı yürüyüş için 3 boyutlu bir düzlemin tanımlanması gerekmektedir. Şekil 1'de insan hareketlerinin analizini kolaylaştıran üç temel referans düzlem ve bunlara karşılık gelen dönüş yönleri gösterilmiştir. X eksen robotun ileri/geri yönü, Y eksen robotun sağ/sol yönü ve Z eksen robotun yukarı/aşağı yönüdür. YZ düzlemi yatış (roll), XZ düzlemi yunuslama (pitch) ve XY düzlemi sapma (yaw) dönüş açıları olarak adlandırılır [11].



**Şekil 1.** İnsansı hareket referans düzlemi (Humanoid motion reference plane)

İki ayaklı bir robotun adım atarak ilerlemesi yürüyüş olarak tanımlanır. Yürüyüş esnasında iki farklı destek noktası oluşmaktadır. Bunlar tek destek ve çift destek noktası olarak adlandırılmaktadır. Tek bir bacak üzerinde tüm robot vücut ağırlığını dengede olduğu faz tek destek noktası olarak adlandırılır. İki bacak yerde iken tüm robot vücut ağırlığının bu iki bacak tarafından dengelendiği faz ise çift destek noktası olarak tanımlanmaktadır. Yürüyüş hareketinin belli bir periyod ile devam etmesi yürüme döngüsünü oluşturmaktadır. Yürüme döngüsü beş aşamadan oluşmaktadır. Yürüme döngüsünün her aşamasında insansı robotun önden görünümü Şekil 2'de gösterilmiştir.

İnsansı robotlarda itme kurtarma denetleyicileri üzerine yapılmış farklı araştırmalar mevcuttur. Bu araştırmalar gerçekleştirilirken farklı kontrol yöntemleri denenmiştir. Stephens tarafından yapılan araştırmalarda insansı robotların itme kurtarma stratejileri üzerine çalışmalar gerçekleştirilmiştir. Stephens üç farklı itme kurtarma stratejisi sunmuştur. Bunlar; ayak bileği, kalça ve adım stratejileridir [1]. Aftab vd. iki ayaklı insansı robotların tedirginlik durumunda



Şekil 2. İnsansı robotlarda yürüme döngüsünün önden görünümü (Front view of the walking cycle in humanoid robots)

denge problemi yaşadıklarını belirtmişlerdir. Denge problemini gidermek amacıyla Model Öngörülü Kontrol (MÖK) yöntemini çözüm olarak önermişlerdir. Tahmine dayalı olan bu yöntemde ayak bileği, kalça ve adım stratejilerinin farklı birleşimleri kullanılmıştır [5]. Ghassemi vd. kararsız yapıda bulunan insansı robotların denge problemlerine odaklanmıştır. Konum kontrollü olan bir insansı robota tork kontrolü yaklaşımı uygulayarak itme kurtarma denetleyicisi oluşturulmuştur. Çalışmalarında kontrol yöntemi olarak PD kontrolü kullanmışlardır [6]. Kim vd. çalışmalarında Lineer Ters Sarkaç Modeli kullanarak analizlerini gerçekleştirmişlerdir. Yüksek seviyeli itme-kurtarma stratejilerini derin pekiştirmeli öğrenme ile Derin Q Ağı(DQA) algoritmasında eğitimlerini uygulamışlardır [9].

Melo vd. yapmış oldukları çalışmada Sıfır Moment noktası kavramına dayanan bir itme kurtarma denetleyicisi tasarlanmıştır. Tasarladıkları kontrolör için Yakınsak Politika Optimizasyonu algoritmasını kullanmışlardır [12]. Li vd. insansı robotun doğrusal ters sarkaç modelini oluşturmak için Bulanık çift DQA algoritmasını önermişlerdir. Tasarlanmış oldukları denetleyici ile gerçek zamanlı bir yürüyüş modeli üzerinde çalışmışlardır. Bulanık çift DQA algoritması ile tasarladıkları denetleyici hem yürüyüş modelini hem de gelen itmelere karşı denge problemlerini çözmektedir. Yapılan denetleyici gerçek zamanlı çalıştığı için sensör bilgileri anında alınmaktadır. Böylece insansı robotun yürüyüşünü bağımsız olarak dengeli bir şekilde yaptığı gözlemlenmiştir [13].

İnsansı robotların dengesini sağlaması yapacağı üst düzey hareketlere, yürüyüş biçimine, yüzeyin eğimine, çevredeki engellere karşı göstereceği tepkilerden dolayı karmaşık ve zor bir görevdir. Bu araştırmada; insansı robotların ayakta dururken ve yürürken dengede durması ve dış kuvvetler sebebiyle oluşacak denge bozukluklarını önlemesini sağlayan aktif dengeleme ve itmeden kurtarma sorunlarına odaklanması amaçlanmıştır [14, 15].

Uygulamaların simülasyon testleri Webots ortamında 3 boyutlu model olarak gerçekleştirilmiştir. Gerçek dünya testleri ise Robotis-OP2 insansı robotu üzerinde yapılmıştır. Robotis-OP2 modern endüstriyel robotlara göre daha ucuz servo motorlardan üretilmiştir.

Bu yüzden gelen itmelere karşı dengede durmaları ve itme geri kazanım özellikleri azdır. Testlerde kullanılacak insansı robotlara farklı güçlerde itmeler uygulanarak dengesini bulması için kapalı döngü kontrol uygulanması yapılmıştır. Bu araştırmanın amacı, bir insanın eylemlerini taklit edebilecek tamamen özerk bir kapalı döngü itme kurtarma kontrol sistemi tasarlamak ve uygulamaktır. Harici bir itme uygulandıktan sonra da sistemin dengesini bulup yörüngesine girmesi istenmektedir. Bu işlemler sırasında insansı robotun tamamen otonom hale getirmek için de klasik kontrol yöntemi olan PID, tahmine dayalı MÖK kontrol ve derin pekiştirmeli öğrenme algoritmalarından DQA kullanılmıştır. Sonuçlar karşılaştırılarak en iyi sonucu veren algoritmanın bulunması amaçlanmıştır.

Bu çalışmanın Bölüm I kısmında problem hakkında genel bilgilere yer verilmiştir. Çalışmanın ilerleyen kısımları şu şekildedir. Bölüm II de uygulamalar için gerekli olan sistem gereksinimleri, Bölüm III yapılan çalışmalar ve tartışmayı kapsamaktadır. Bölüm IV ise, sonuçlar ve gelecekteki çalışmaları içermektedir.

## 2. Sistem Gereksinimleri (System Requirements)

Bu çalışmada sistemi oluşturabilmek için Robotis-OP2 insansı robotuna gereksinim vardır. İnsansı robotu programlayıp simüle etmek için Webots simülatörü kullanılmıştır. İtmelerden kurtulma için itme kurtarma stratejilerinin teorik bilgisine ve kontrolörlerimiz için DQA algoritmasına, PID kontrolör ve MÖK kontrolör bilgilerine sahip olmamız gerekmektedir.

### 2.1. Robotis-OP2 (Robotis-OP2)

Bu çalışmada, Robotis firması tarafından eğitim ve araştırma amacı ile geliştirilen Robotis OP2 insansı robot kullanılmıştır. Bir insanın fiziksel yapısına uygun olarak kafa, gövde, kollar ve bacaklardan oluşmaktadır. Açık kaynak kodlu bir platform olup, gelişmiş işlem kabiliyetine sahiptir. Robot üzerinde 20 serbestlik derecesi vardır. İçerisinde kamera, hoparlör, usb girişi, mini HDMI, 3 eksenli gyroscope, 3 eksenli ivmeölçer, internet girişi ve ledler bulunmaktadır. Her bir ekleminde Dynamixel MX-28T servo

motorlar yer almaktadır. Intel Atom N2600@1.6 GHz çift çekirdekli işlemciye sahiptir. Robotis-OP2 Linux tabanlı işletim sistemlerinin yanı sıra Windows tabanlı işletim sistemlerini de desteklemektedir. Birçok simülâtörde Python, Java, C, C++ gibi programlama dilleri kullanılarak programlanabilmektedir. Donanımları sayesinde birçok robotik uygulamayı gerçekleştirmeye olanak sunan eğitim ve araştırma tabanlı bir insansı robot platformudur.



**Şekil 3.** Robotis firması tarafından üretilen Robotis-OP2 (Robotis-OP2 manufactured by Robotis)

### 2.2. Webots (Webots)

Robotlarda en önemli zorluklardan biri gerçek dünya testleridir. Gerçek dünyada robotların eğitilmesi uzun sürmekte ve maliyeti yüksek olmaktadır. Gerçek dünya testlerine alternatif yöntem simülasyon üzerinde işlemleri gerçekleştirmektir. Robotların yüksek maliyetli donanımlarının zarar görmemesi için önce sanal ortamda testleri yapılmaktadır [14].

Bu çalışmamızda Webots tercih edilmiştir. Webots, Cyberbotics firması tarafından geliştirilmiş ücretsiz bir robot geliştirme simülâtörüdür. Webots 3D tasarım yapmaya olanak sağlamaktadır. İçerisinde bulunan derleyicisi sayesinde C/C++, Java, Python, MATLAB gibi programlama dilleri ile kodlama olanağı sunmaktadır. Kolay kullanımı ve kullanıcı dostu arayüzü sayesinde birçok akademik çalışmada kullanılmaktadır. İçerdiği kütüphaneler ile birçok farklı robot programlanabilmektedir. Webots içerisinde yer alan robotlar gerçek dünyada çalışacak şekilde programlanmaktadır. Webots içerdiği fizik kütüphanelerindeki sensörler aracılığı ile araştırmacılara simülasyon ortamında istenilen gürültülerin eklenmesine olanak sağlamaktadır. Webots ile dört ayaklı, iki ayaklı, tekerlekli, uçan, yüzebilen ve sabit robotlar programlanıp simüle edilebilmektedir. İşletim sisteminden bağımsız olarak çalışması araştırmacılara kolaylık sağlamaktadır.

### 2.3. Kontrol Yöntemleri (Control Methods)

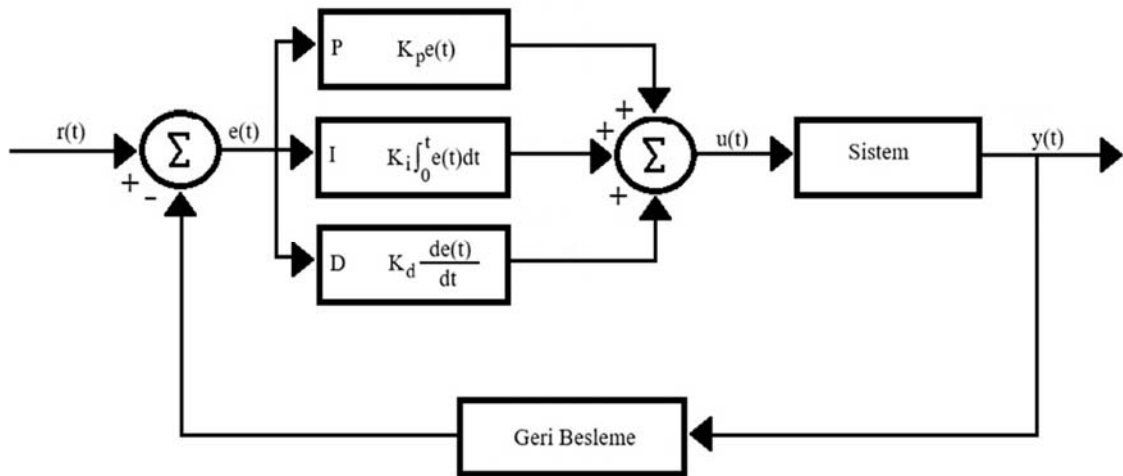
PID, oransal, integral ve türev elemanlarını toplamıyla oluşan bir kontrolördür. PID kontrolör, giriş sinyali ve geri besleme sinyalini karşılaştırarak hatayı hesaplayan bir kontrolördür. PID kontrolörde amaç zaman içerisinde oluşan hatayı en aza indirmektir. Hatayı minimize etmek için PID parametrelerinin iyi bir şekilde ayarlanması gerekmektedir. Sistemin ihtiyacına göre P, I, PI, PD, PID kontrolör olarak ayarlanabilmektedir. PID kontrolör uzun zamandan beri yaygın olarak kullanılan bir kontrolördür. Basit ve anlaşılır yapısı sayesinde endüstride birçok sistemin kontrolünde tercih edilmektedir [16]. PID kontrolörün blok diyagramı Şekil 4'te verilmiştir.

PID kontrolörler hatanın sıfır olabilmesi için üç parametre kullanılmaktadır. Bu parametreler  $K_p$ ,  $K_i$ ,  $K_d$  şeklindedir. Şekilde ifade edilen  $r(t)$  referans sinyali,  $e(t)$  hata fonksiyonu,  $u(t)$  sistemin girişi ve  $y(t)$  sistemin çıkışıdır. Blok diyagramında da görüldüğü gibi hata fonksiyonu Eş. 1'de verilmiştir.

$$e(t) = r(t) - y(t) . \quad (1)$$

Sistemin hata değeri, referans giriş değerinden sistemin çıkış değeri çıkarılarak bulunur. Sistemlerde  $e(t)$  fonksiyonunun sıfır olması istenmektedir. PID kontrolör matematiksel olarak oransal integral ve türevin toplamından oluşmaktadır. Eş. 2'de PID kontrolörün matematiksel ifadesi verilmiştir.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} . \quad (2)$$



**Şekil 4.** PID Kontrolör blok diyagramı (PID Controller block diagram)

Burada  $K_p$  oransal kazanç katsayısı,  $K_i$  integral kazanç sabiti,  $K_d$  türev kazanç sabiti ve  $t$  zamanı ifade etmektedir. PID kontrolördeki oransal terimi, tüm geçişlerin kazanç faktörü yoluyla hata sinyali ile orantılı olarak kontrol edilmesini sağlar. İntegral terimi düşük frekanslar için ayarlanarak kararlı durum hatalarını azaltır. Türev terimi ise yüksek frekanslı farklılaştırıcı telafisi yoluyla geçici yanıtı iyileştirir [17]. Bir PID kontrolörde girişe uygulanan sinyal bu üç terime bağlı olarak çıktı üretir. PID kontrolörün düzgün sonuçlar vermesi için katsayıların düzgün olarak seçilmesi gerekmektedir. Bu katsayıların hesaplanması için farklı yöntemler bulunmaktadır. Wang-Juan-Chan, Chien Hroners Reswik, Ziegler-Nichols gibi klasik yöntemler parametrelerin hesaplanmasında kullanılmaktadır [18].

MÖK sistemin gelecekteki hareketini tahmin etmek için bir süreç model kullanan gelişmiş bir kontrol yöntemidir. MÖK kontrolörün temel amacı, sistemin sonraki hareketini sınırlı zaman içerisinde tahmin etmek ve sistem kısıtlarını karşılayarak en az maliyetle en uygun kontrol girdisini hesaplamaktır. Oluşturulan sistemde tahmini hesaba katan MÖK yöntemi, kısıtlı bir optimizasyon problemini çözerek optimal bir çıkış oluşturur [19].

MÖK, gelecekteki olayları öngörme yeteneğine sahiptir ve buna göre kontrol önlemleri alabilir. PID kontrolörleri bu tahmin yeteneğine sahip değildir. MÖK, geleneksel geri besleme kontrol sistemleri tarafından kontrol edilemeyen sistemleri bile kontrol edebilir [20, 21]. MÖK kontrolör blok diyagramı Şekil 5'te verilmiştir.

Bir dizi kontrol sinyali uygulandığında;  $\bar{U}$  doğrusal bir model,  $X_{t+1} = AX_t + BU_t$  durum dizisi,  $\bar{X}$  bir sonraki N zaman adımı Eş. 3'de verilmiştir.

$$\bar{X} = \bar{A}X_t + \bar{B}\bar{U}, \quad (3)$$

Eş. 3 mevcut durum olan  $X_t$  ile eşleşen basit bir fonksiyondur.  $\bar{X}$  bir yörünge  $\bar{U}$  yörünge üzerindeki eylemlerdir ve sırasıyla Eş. 4 ve Eş. 5'de verilmiştir.

$$\bar{X} = (X_{t+1}^T, \dots, X_{t+N}^T)^T, \quad (4)$$

$$\bar{U} = (U_{t+1}^T, \dots, U_{t+N-1}^T)^T. \quad (5)$$

$\bar{A}$  ve  $\bar{B}$  matris formları sırasıyla Eş. 6 ve Eş. 7'de verilmiştir.

$$\bar{A} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N-1} \\ A^N \end{bmatrix}, \quad (6)$$

$$\bar{B} = \begin{bmatrix} A & 0 & \dots & \dots & 0 \\ AB & B & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A^{N-2}B & A^{N-3}B & \dots & B & 0 \\ A^{N-1}B & A^{N-2}B & \dots & AB & B \end{bmatrix}. \quad (7)$$

En iyi yörüngeyi belirlemek için bir maliyet fonksiyonu oluşturulur. Maliyet işlevi, yörüngeyi puanlayan skaler bir işlevdir. MÖK kontrolörler tipik olarak  $\bar{X}$  ve  $\bar{U}$  öğeleri kullanarak maliyet fonksiyonunun hesaplanması Eş. 8'de verilmiştir.

$$J = \frac{1}{2} \bar{X}^T \bar{Q} \bar{X} + \frac{1}{2} \bar{U}^T \bar{R} \bar{U}, \quad (8)$$

Eş. 8'de belirtilen denklemin içine Eş.3'deki denklem eklenerek maliyet Eş. 9'da verilmiştir.

$$J = \frac{1}{2} \bar{U}^T (\bar{R} + \bar{B}^T \bar{Q} \bar{B}) \bar{U} + X_t^T \bar{A}^T \bar{Q} \bar{B} \bar{U} + \frac{1}{2} X_t^T \bar{A}^T \bar{Q} \bar{A} X_t. \quad (9)$$

Eş. 9'u basit bir formda göstermek gerekirse Eş. 10'daki denklemi elde ederiz.

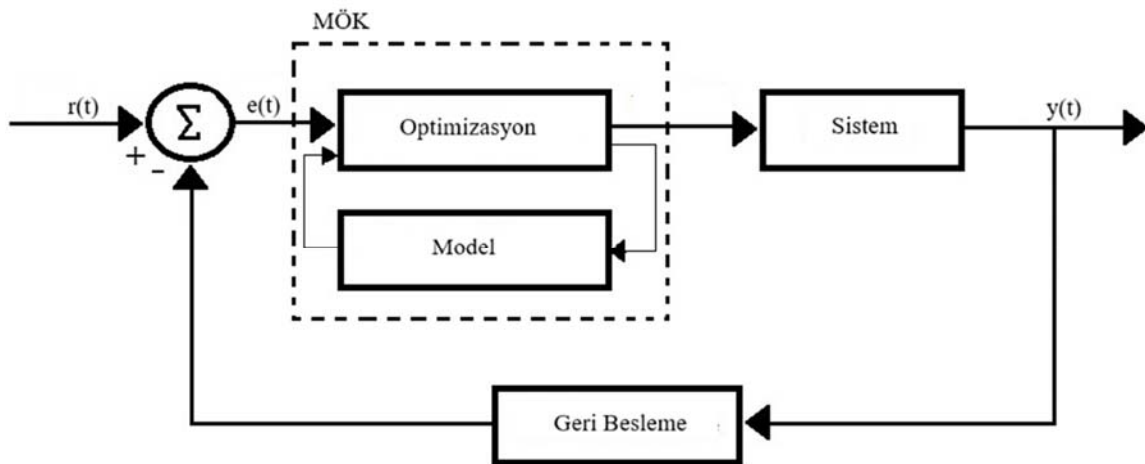
$$J = \frac{1}{2} \bar{U}^T H \bar{U} + f^T \bar{U} + J_0. \quad (10)$$

Burada yer alan  $H$ ,  $f^T$ ,  $J_0$  sırasıyla Eş.11, Eş. 12 ve Eş. 13'de verilmiştir.

$$H = \bar{R} + \bar{B}^T \bar{Q} \bar{B}, \quad (11)$$

$$f^T = X_t^T \bar{A}^T \bar{Q} \bar{B}, \quad (12)$$

$$J_0 = \frac{1}{2} X_t^T \bar{A}^T \bar{Q} \bar{A} X_t. \quad (13)$$



Şekil 5. MÖK kontrolör blok diyagramı (MPC controller block diagram)

$J_0$  teriminin  $\bar{U}$ 'ya göre sabit olduğu ve minimum  $J$ 'nin konumu üzerinde hiçbir etkisinin olmadığına dikkat edilmelidir. Kısıtlamalar olmadan en uygun yörünge bulunabilir. Eş. 10'da yer alan  $\bar{U}$  yerine, sonucu sifıra ayarlamak ve en uygun yörüngeyi çözmek için  $\bar{U}^*$ ,  $\bar{U}$ 'nun bir fonksiyonu olarak Eş. 14'de gösterilmiştir.

$$\bar{U}^* = -H^{-1}f. \quad (14)$$

Bununla birlikte, kısıtlamalar genellikle sisteme eşitlik kısıtlamaları Eş. 15'de olduğu şekilde uygulanır,

$$C_{eq}\bar{U} = b_{eq}. \quad (15)$$

Veya eşitsizlik kısıtlamaları Eş. 16'daki gibi ifade edilir:

$$C_{in}\bar{U} \leq b_{in}. \quad (16)$$

Kısıtlamaların varlığında, bir programlama çözücü gereklidir. Bu çözücüler, kısıtlamaları eşzamanlı olarak karşılayan en uygun sonucu bulmak için daha basit alt problemleri yinelemeli olarak çözmektedirler.

DQA, Q öğrenmede yapay sinir ağları ile kullanılan bir derin öğrenme algoritmasıdır. DQA algoritması sürekli zamanlı politika dışı bir öğrenme yöntemidir [22-24]. DQA sinir ağının girdisini durumlar, çıktısını ajanın gerçekleştirdiği eylemler oluşturmaktadır. DQA algoritmasında her bir keşifte elde edilen örnekler (s,a,r,s') hafızada tutulmaktadır. Buradaki s durum, a eylem, r ödül, s' ise sonraki durumdur. DQA algoritmasında öncelikle s durumunda a eylemi gerçekleştirilir. Gerçekleştirilen bu eylem sonucunda bir r ödülü kazanılır ve s' sonraki duruma geçilir. DQA algoritması deneyim tekrarı özelliğini kullanarak ajanın deneyimlerini hafızada tutmaktadır. Hafızada tutulan bu örneklerle alınarak ajanın deneyimleri üzerinden öğrenmesi sağlanmaktadır [25]. DQA algoritmasının akış diyagramı Şekil 6'da gösterilmiştir.

Bu algoritmanın en önemli özelliği hatırlatma ve yeniden oynatma fonksiyonlarıdır. Ajan öğrenmiş olduğu deneyimleri belli bir süreden sonra unutma eğilimindedir. Ajanın yeniden eğitimi için öğrenmiş olduğu deneyimlere ihtiyaç duyulmaktadır. DQA algoritması ile ajanın deneyimleri hafızada tutulur [26]. Öğrenme esnasında bu veriler rastgele örnekler ile ajanın eğitimi için kullanılır. DQA algoritması aşağıda verilmiştir.

DQA algoritması (DQN algorithm)

```

1: Yeniden oynatma belleğini başlat(M)
2: Ağ ağırlıklarını başlat
3: for bölüm=1 to M do
4:   s0 başlangıç durumunu elde etmek için çevre ile etkileşim kurun
5:   for t=0 to T do
6:     Rastgele bir at eylemlerini elde etmek için ε stratejisini kullan
7:     at eylemi gerçekleştir ve st+1 durumuna ilerle ve mevcut ödül rt'yi hesapla.
8:     (st,at,st+1,rt) değerlerini M'ye kaydet
9:     L(θ) = [Q(s, a) - (r + γmaxQ(s', a'))]2 kayıp değerini hesapla.
10:   end for
11: end for

```

Algoritmada ajanın tahmininin gerçek hedeften ne kadar uzakta olduğunu veren değer kayıp olarak ifade edilmektedir. Burada tahmini, hedeften çıkararak kaybı bulabiliriz [27]. DQA algoritmasında kayıp fonksiyonu Eş. 17'de tanımlanmıştır.

$$L(\theta) = [Q(s, a) - (r + \gamma \max_{a'} Q(s', a'))]^2. \quad (17)$$

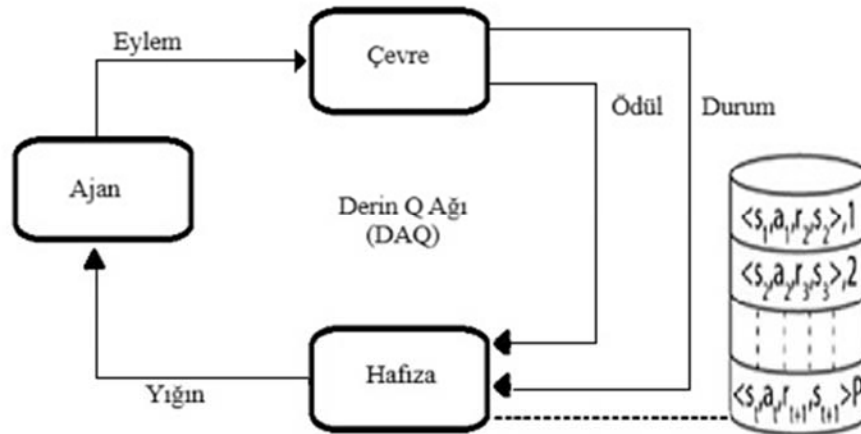
Burada a ajanın yapmış olduğu eylemi, r eylem sonunda elde edilen ödülün değerini, s durumu, γ indirim oranını göstermektedir.

Yıkıcı bir girdi uygulandığında, sistemin zaman içinde yaklaşık denge noktasına yaklaşması durumunda asyptotik olarak kararlı olduğu söylenir [28]. Çalışmalarında, Fan vd. Q-yineleme algoritmasının basitleştirilmiş bir sürümünü inceleyerek DQA algoritmasının yakınsamasını oluşturmayı önermişlerdir. Bu algoritma ile DQA arasındaki fark, durum-eylem-ödül-durum kümelerini depolamak için yinelenen arabellek M olmamasıdır [29, 30].

#### 2.4. İtme-Kurtarma Stratejileri (Push-Recovery Strategies)

İnsansı robotların denge problemlerine farklı çözümler önerileri sunulmuştur. Bu çözüm önerilerinin çoğu önceden tanımlanmış engeller ve itmelere karşı başarılı olmaktadır [30, 31]. Fakat insansı robotların gerçek dünyada beklenmeyen itme ve engellerle karşılaşma ihtimalleri yüksektir. Bu tarz problemler de başarılı olmak için farklı stratejiler geliştirilmiştir. Bu stratejiler arasında en temel üç yöntem bulunmaktadır. Bunlar: ayak bileği, kalça ve adım stratejileridir.

Ayak bileği stratejisi; kütle merkezini değiştirerek robotun ayak bileklerini dengelemek için kullanan basınç merkezi stratejisidir [32-35]. Mekanik olarak bu strateji, robot vücudunu ayak bileği eklemleri



Şekil 6. DQA blok diyagramı (DQN block diagram)

etrafında döndürerek dengeyi sağlamasından ibarettir. Ayak bileği stratejisi, robotun kütle merkezini kontrol etmektedir. Kütle merkezini dengede tutmak için ayak bileği eklemlerine kontrol torku uygulanır. Ayak bileği stratejisinin matematiksel ifadesi Eş. 18'de verilmiştir.

$$\ddot{x} = \frac{g}{z_0} \left( x - \frac{\tau_{ankle}}{mg} \right). \quad (18)$$

Burada  $\tau_{ankle}$  tork değerini,  $m$  doğrusal ters sarkaç modelinin kütleliğini,  $g$  yerçekimini ve  $z_0$  kütle merkezinin yüksekliğini ifade etmektedir.  $\ddot{x}$ , kütle merkezinin mevcut destek noktasından yatay mesafesidir. Şekil 7a'da ayak bileği stratejisine dayalı model verilmiştir.

Kalça stratejisi; robotu dengede tutabilmek için hem ayak bileklerini hem de kalçalarını kullanan, vücudun pozisyonunu değiştirmesine ve gövdeyi sallayarak açılmal momentum oluşturmaya izin veren merkezi moment eksenli stratejisidir [1]. Bu metot ile robotun kalça kısmında bulunan servo motorlar gelen itmeye uygun bir tork üreterek dengede kalmaya çalışır. İnsanların gelen itmelere karşı kalça eklemlerini kullanarak dengede kalmaya çalışmalarına benzer bir yol izlerler. Bu strateji ayak bileği stratejisine göre daha etkili bir yöntemdir. Kalça stratejisini matematiksel ifadesi Eş. 19 verilmiştir. Eş. 19'da yer alan  $\tau_{hip}$  ifadesinin eşiti Eş. 20'de verilmiştir.

$$\ddot{x} = \frac{g}{z_0} \left( x - \frac{\tau_{hip}}{mg} \right). \quad (19)$$

$$\tau_{hip} = I\theta. \quad (20)$$

Burada  $\tau_{hip}$  gövdeye uygulanan tork değerini,  $m$  modelinin kütleliğini,  $g$  yerçekimini,  $z_0$  kütle merkezinin yüksekliğini ve  $I$  dönme ataletini ifade etmektedir.  $\ddot{x}$ , kütle merkezinin mevcut destek noktasından yatay mesafesidir. Şekil 7b'de kalça stratejisine dayalı model verilmiştir.

Adım stratejisi; Dış kuvvetlere karşı uygun bir yönde bir veya daha fazla adım atarak denge konumuna geçiren yakalama noktası stratejisidir [36, 37]. Dışardan gelen itme robotun dayanacağı maksimum tork değerini aştığı zaman ayak bileği ve kalça stratejileri yetersiz kalır. Bu durumda robotun dengede kalması için adım atması gerekir. Robota gelen itmeye karşı destek noktasına adım atarak

yapılan bu hamle adım stratejisi olarak adlandırılmaktadır. Robotun düşmeden dengede kalabilmesi için atacağı en iyi adım uzunluğu, denge noktası içinde olmalıdır. Adım stratejisinde, itmenin geldiği yöne doğru destek adımı atılarak denge sağlanır. Şekil 7c'de adım stratejisine dayalı model verilmiştir. Dengede olmayan bir robotun yakalama noktası gövde hızıyla orantılıdır ve Eş. 21'de verilen matematiksel formül ile hesaplanır.

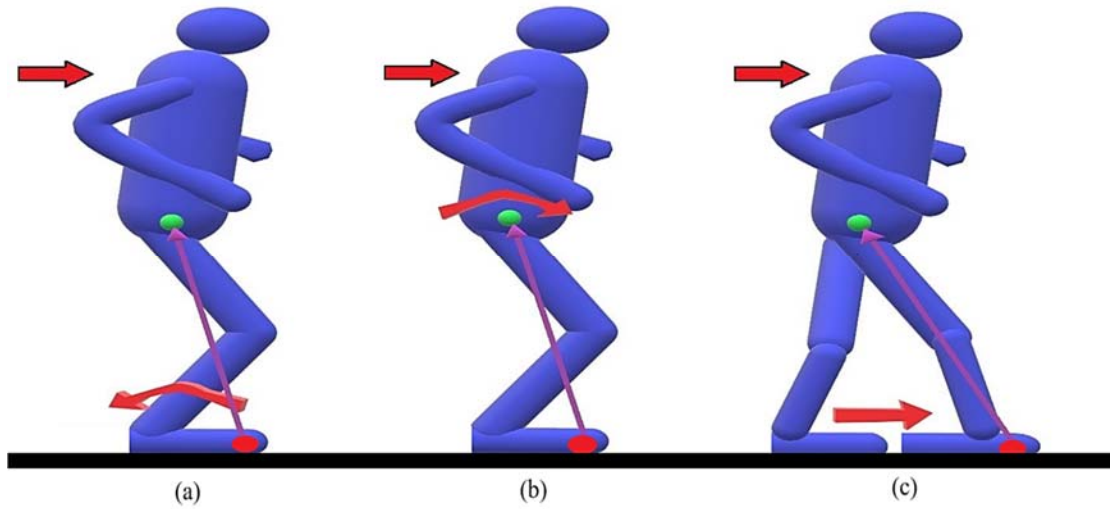
$$X_{yakalama} = \ddot{x} \sqrt{\frac{z_0}{g}}. \quad (21)$$

Burada  $X_{capture}$  yakalama noktası,  $\ddot{x}$  gövdenin doğrusal hızıdır. Adım stratejisi, yakalama noktasının robotun gövdesinin doğrusal hızıyla orantılı olduğunu ve robotu itmeden-kurtarmak için atılması gereken adımın uzaklığını göstermektedir.

### 3. Deneysel Metot (Experimental Method)

Bu çalışmada iki ayaklı insansı robotlara gelen itmelere karşı dengede durabilmelerini sağlayan kontrol yöntemleri uygulanmıştır. Yapılan uygulamada hem simülasyon hem de gerçek dünya testleri gerçekleştirilmiştir. Simülasyon ortamı olarak Webots robot geliştirme simülâtör aracı kullanılmıştır. Testlerin gerçekleştirdiği insansı robot Robotis-OP2'dir. Robotu eğitmek için PID kontrolör, MÖK kontrol yöntemi ve derin pekiştirmeli öğrenme algoritmalarından DQA yöntemi kullanılmıştır. Üç yöntemin sonuçları karşılaştırılmıştır. Uygulamaları gerçekleştirmek için Python ve C++ programlama dilleri kullanılmıştır.

Robotun sensörlerinden alınan ivme, gyro, tork ve konum gibi bilgileri anlık olarak kaydedilmiştir. Kaydedilen bu veriler ile robotun gövdesinin denge pozisyon aralığında olup olmadığı kontrol edilmiştir. Simülasyon içerisinde robota kuvvet uygulamak için bir fizik kütüphanesi oluşturularak robot ile bağlantısı gerçekleştirilmiştir. Robota önden ve arkadan 10 N ve -10 N aralığında rastgele kuvvetler uygulanmıştır. Uygulanan kuvvetler sürekli kontrol edilerek, denge konumu dışına çıktığı zaman ayak bileğine müdahale ederek robot denge pozisyonuna getirilmiştir. Projede itme-kurtarma stratejilerinden ayak bileği stratejisi referans alınarak gerçekleştirilmiştir. Robotun eğitim sürecinde klasik kontrol yöntemi olarak PID, tahmine dayalı kontrol yöntemi olarak MÖK ve



Şekil 7. Üç temel dengeleme stratejisi. Yeşil nokta kütle merkezini, kırmızı nokta basınç merkezini ve mor ok yer tepki kuvvetini temsil eder. a) Ayak Bileği Stratejisi b) Kalça Stratejisi c) Adım Stratejisi. (Three basic balancing strategies. The green dot represents the center of mass, the red dot the center of pressure and the purple arrow the ground reaction force. a) Ankle Strategy b) Hip Strategy c) Step Strategy.)



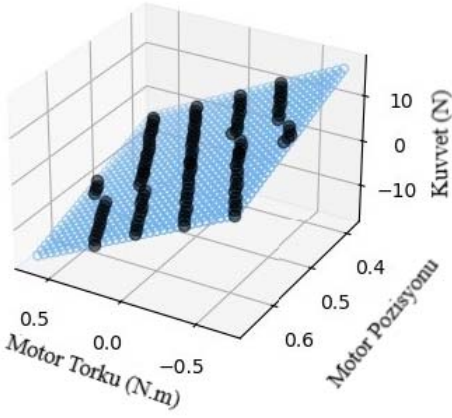
derin pekiştirmeli öğrenme algoritmasından DQA uygulanmıştır. Algoritmelerde kullanılan durumlar Eş.22'de gösterilmiştir.

$$S = (tSign, abs(Joint_{tork}), anklePos, fSign, abs(Force)) . (22)$$

Burada; tSign torkun işareti, abs(Joint\_Tork) torkun mutlak değerini, anklePos ayak bileğinin pozisyonunu, fSign uygulanan itmenin işaretini ve abs(Force) uygulanan itmenin büyüklüğünün mutlak değerini verir. Uygulan itmenin değeri ayak bileğinin tork ve pozisyon bilgileri kaydedilip multi-lineer regresyon ile Eş. 23'deki gibi hesaplanmıştır.

$$F = 20.9255 - (16.0364 * T) - (40.5437 * P) \quad (23)$$

Burada; F uygulanan itme kuvvetinin büyüklüğü, T ayak bileğinin torku, P ise ayak bileğinin konum bilgisidir. Rastgele alınmış 120 pozisyon ve tork bilgilerinin multi-lineer regresyon grafiği Şekil 8'de verilmiştir. R-squared, istatistiksel ölçme metriği 1 olarak elde edilmiştir.



Şekil 8. Robota uygulanan kuvvet grafiği (Graph of force applied to the robot)

Robotun gelen itmeye karşı uygulayacağı eylemler üç tanedir. Bunlar sırasıyla;

- Dur(0),
- ileri hareket(1),
- geri hareket(2).

Robotun yaptığı harekete göre kazanacağı ödül fonksiyonu R olarak ifade edilir. R formülü Eş. 24'te verilmiştir. Robotun almış olduğu ödül puanına göre kullanılan yöntemlerin karşılaştırılması yapılmıştır. Ödül hesaplanırken itme uygulandıktan sonraki robotun ayak baz alınmıştır.

$$R = 0.25 - T^2 \quad (24)$$

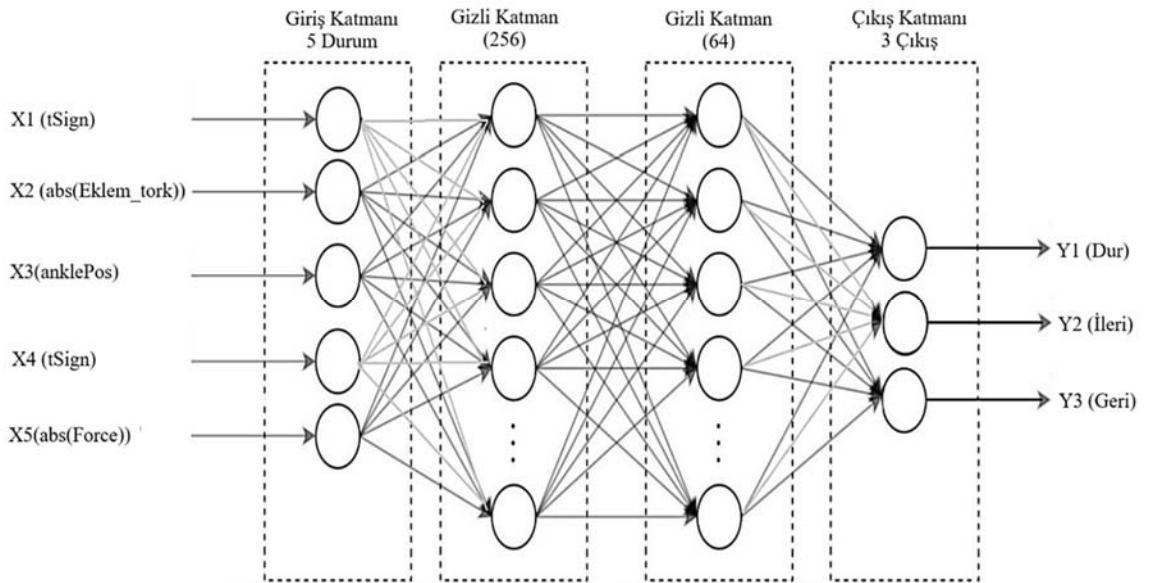
Oluşturulan yapay sinir ağı modelin mimarisi Şekil 9'da gösterilmiştir.

Klasik kontrolör için kullanılan parametre değerleri Tablo 1'de verilmiştir. İtme esnasındaki ani değişimlere sönmüldürmek için PD kontrolör tercih edilmiştir. Bundan dolayı I değeri 0 olarak alınmıştır. P ve D değerleri seçilirken deneme yanılma yöntemi kullanılmıştır. Bu yöntemin tercih edilme sebebi robotun transfer fonksiyonunun açık erişimli olarak verilmemesinden kaynaklanmıştır. 0-10 değeri arasında değerler denenerek en iyi parametrelere Tablo 1'de olduğu gibi erişilmiştir.

Tablo 1. PD kontrolör parametreleri (PD controller parameters)

Parametreler	Değerler
P	0.5
I	0
D	0.01
Örnekleme Zamanı	0.00001
Eylem Ölçüsü	3
Durum Ölçüsü	5
Maksimum Adım	30

Oluşturulan DQA algoritmasında kullanılan parametreler ayrıntılı olarak Tablo 2'de, yapay sinir ağı özellikleri Tablo 3'de verilmiştir.



Şekil 9. Oluşturulan yapay sinir ağı mimarisinin yapısı (Structure of the artificial neural network architecture)

DQA algoritmasında aktivasyon fonksiyonu olarak ReLU kullanılmıştır.

**Tablo 2.** DQA algoritması parametreleri (DQN algorithm parameters)

Parametreler	Değerler
Bölüm Sayısı	7500
Hafıza	2000
Gamma	0.95
Epsilon	1.0
Minimum Epsilon	0.005
Epsilon Azalma	0.995
Grup Boyutu	64
Eğitim Başlangıç	100
Eylem Ölçüsü	3
Durum Ölçüsü	5
Maksimum Adım	30

**Tablo 3.** DQA algoritması yapay sinir ağı özellikleri (DQA algorithm artificial neural network properties)

Parametreler	Değerler	Aktivasyon Fonksiyonu
Giriş Katmanı	5	ReLU
Gizli Katman 1	256	ReLU
Gizli Katman 2	64	ReLU
Çıkış Katmanı	3	ReLU
Optimize Edici	RMSprop Algoritması	
Kayıp	Ortalama Kare Hata	
Öğrenme Oranı	0.00025	
Rho	0.95	
Epsilon	0.01	

İnsansı robotun denge hareket akış şeması Şekil 10'da verilmiştir.

Her yöntem için her bir bölüm 30 adımdan oluşmaktadır. PD ve MÖK kontrolörler 100 bölümden oluşmaktadır. Çünkü belli bir döngü içerisinde sonuçları tekrar etmektedir. DQA algoritması ise 7500 bölümden oluşmaktadır. Tablo 4'te Kontrol yöntemlerinin kaç bölüm ve adımdan oluştuğu gösterilmiştir.

**Tablo 4.** Algoritma bölüm sayıları (Algorithm episodes numbers)

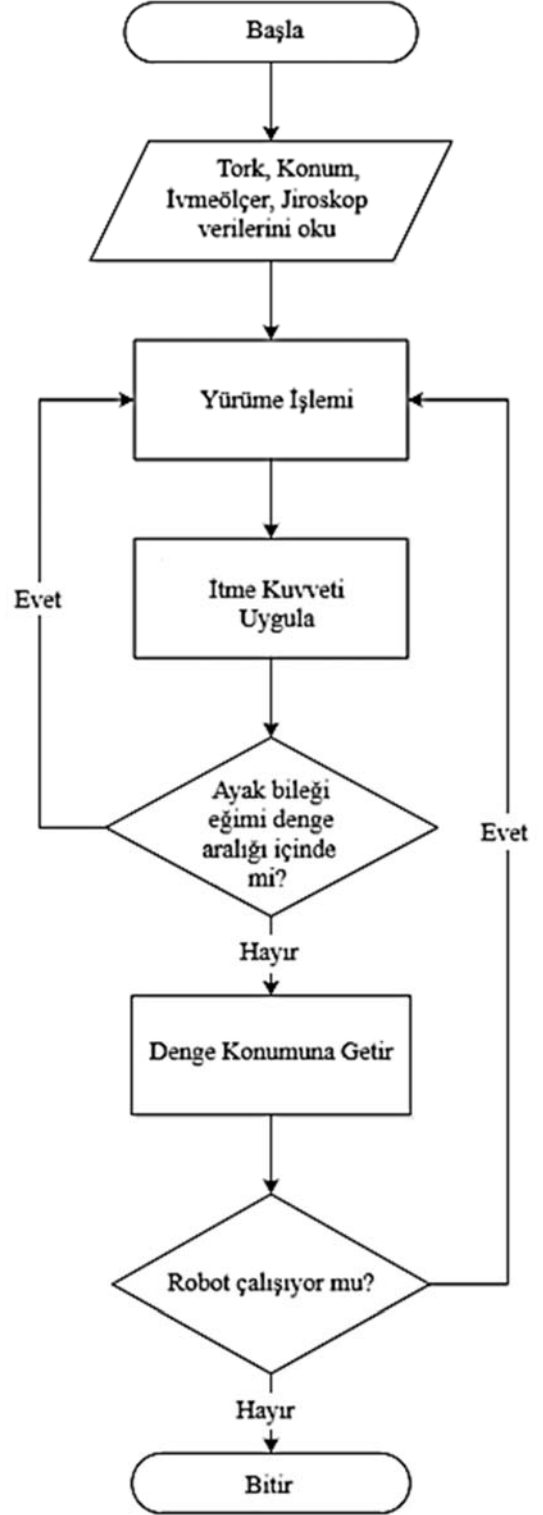
Kontrol Yöntemi	Bölüm	Adım Sayısı
PD	100	30
MÖK	100	30
DQA	7500	30

Şekil 11'de simülasyon ortamında arkada uygulanan itme sonucunda robotun denge konumuna gelmesi gösterilmiştir. Şekil 12'de ise simülasyon ortamında önden uygulanan itme sonucunda robotun denge konumuna gelmesi gösterilmiştir.

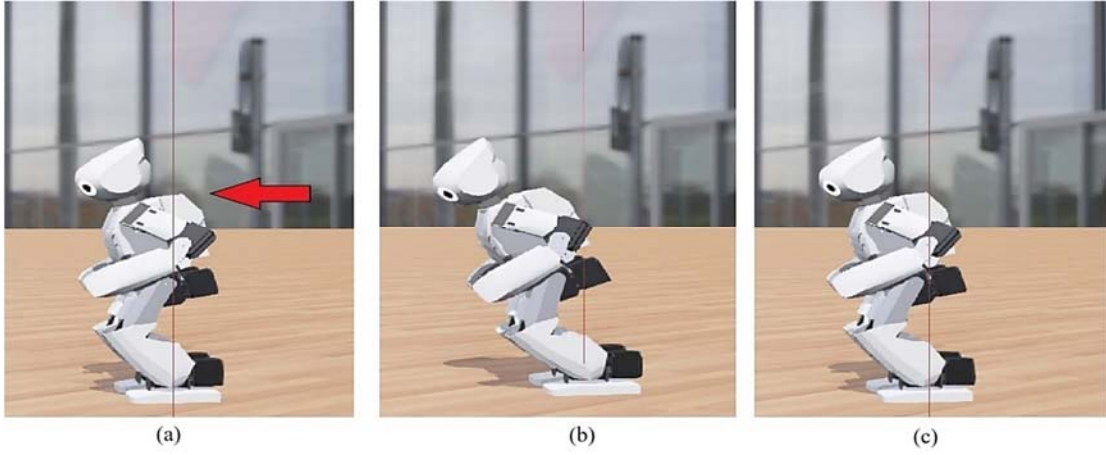
PD kontrolör ile 100 bölüm gerçekleştirilmiştir. Her bir bölüm 30 adımdan oluşmaktadır. Uygulamada belirlenen ödül fonksiyonuna göre en yüksek ödül puanı 7,5'tur. PD klasik bir kontrol yöntemi olduğu için verilen değer girdilere uygun bir çıktı hesaplamaktadır. Sonuç değerleri bir döngü olarak tekrar etmektedir. PD kontrolör için ödül puanı ortalama 3,24 olarak hesaplanmıştır. 100 bölüm sonunda toplam 324,265 puan elde etmiştir. Şekil 13'te PID kontrol yöntemi ile alınan sonuçlar verilmiştir. Eğitim sonunda %43,23 başarı elde edildiği gözlemlenmiştir.

MÖK kontrolör ile 100 bölüm gerçekleştirilmiştir. Her bir bölüm 30 adımdan oluşmaktadır. Uygulamadaki belirtilen ödül fonksiyonuna göre en yüksek ödül puanı 7,5'tur. MÖK kontrolör uygulandığında sonuç değerleri PD de olduğu gibi bir döngü olarak tekrar etmektedir. Ödül puanı ortalama 3,47 olarak hesaplanmıştır. 100 bölüm sonunda

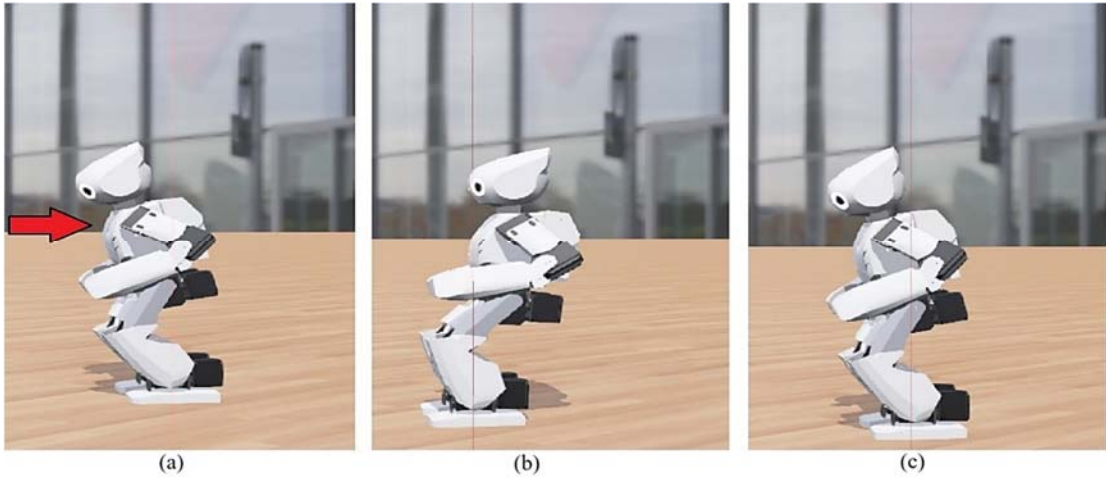
347,068 puan elde etmiştir. Şekil 14'te MÖK kontrol yöntemi ile alınan sonuçlar gösterilmiştir. Eğitim sonunda %46,27 başarı elde edildiği gözlemlenmiştir.



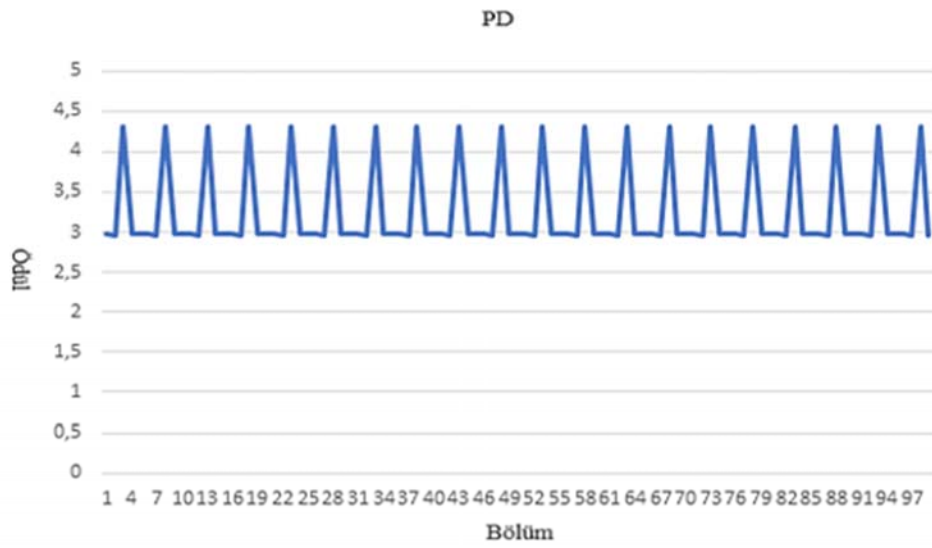
**Şekil 10.** İtme-Kurtarma kontrollü insansı robotun hareket akış diyagramı (Motion flow diagram of a humanoid robot with push-recovery control)



Şekil 11. a) Arkadan itme b) İtme sonucu aldığı pozisyon c) Denge pozisyonu  
(a) Push from behind b) Position as a result of pushing c) Balance position)



Şekil 12. a) Önden itme b) İtme sonucu aldığı pozisyon c) Denge pozisyonu (a) Front push b) Position as a result of push c) Balance position)



Şekil 13. PD kontrol sonuçları (PD control results)

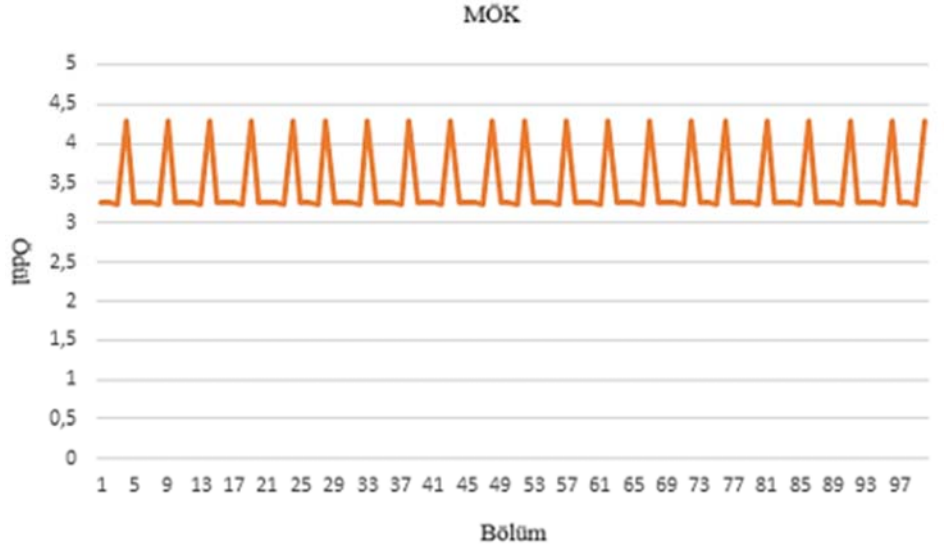
PID ve MÖK kontrol yöntemleri karşılaştırıldığında MÖK kontrol yöntemi PD kontrolöre göre daha iyi sonuç vermektedir. Şekil 15'te PD ve MÖK yöntemlerinin karşılaştırma sonuçları verilmiştir.

DQA algoritması kullanılarak 7500 bölüm gerçekleştirilmiştir. Robot yaklaşık 7200 bölüm sonra dengede kalmayı öğrenmiştir. Her bir bölüm 30 adımdan oluşmaktadır. Ödül fonksiyonuna göre tam başarılı olan bir bölümden alınacak maksimum puan 7,5'tur. DQA algoritmasında yaklaşık 7200 bölüm sonra robotun kazanmış olduğu puan 7,411 olduğu tespit edilmiştir. DQA algoritması kullanılarak eğitilen robot eğitim sonunda toplam 37483,51 puan almıştır. DQA algoritması ile alınan sonuçlar Şekil 16'da gösterilmiştir. Ödül puanı ortalama 4,99 olarak hesaplanmıştır. 7500 bölüm sonunda almış olduğu en yüksek ödül ise 7,437 olduğu görülmüştür.

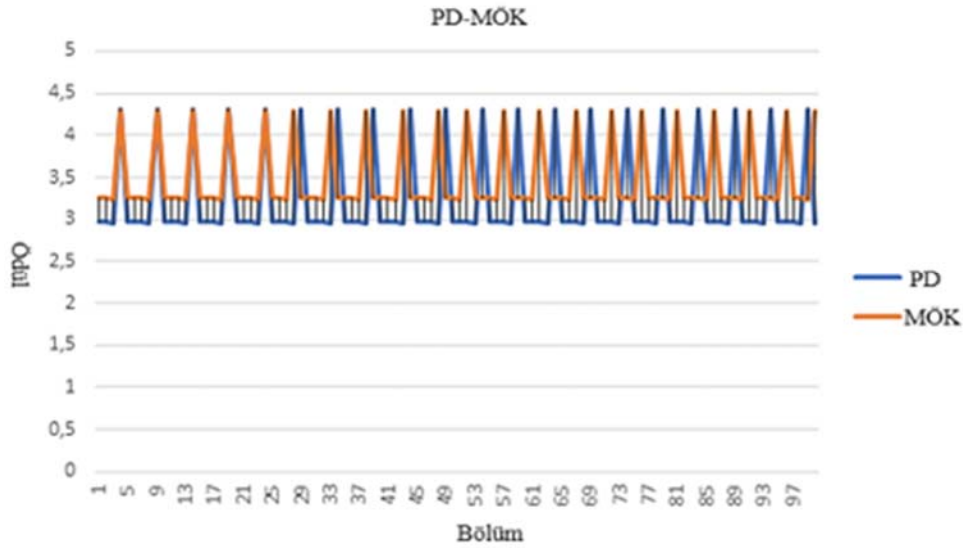
Çalışmada uygulanan üç yöntem sonucunda da insansı robota uygulanan dış itme sonucunda robot denge konumuna gelebilmiştir. Robotun denge konumuna gelmesinde uygulanan kuvvetin büyüklüğü

ve yönü önemli rol almaktadır. Uygulama sonuçları Tablo 5'te verilmiştir. PD kontrol en düşük ödül ortalamasına ve başarı performansı gösteren yöntem olmuştur. MÖK kontrol yöntemi, PD kontrolden daha iyi sonuçlar vermiştir. Fakat derin pekiştirmeli öğrenme algoritmasına göre başarısız olarak görünmektedir. Yapılan bu çalışmada derin pekiştirmeli öğrenme algoritması olan DQA en iyi sonuçları verdiği gözlemlenmiştir. Uygulama sonucunda en başarılı yöntem Tablo 5'te de görüldüğü gibi DQA algoritması olmuştur.

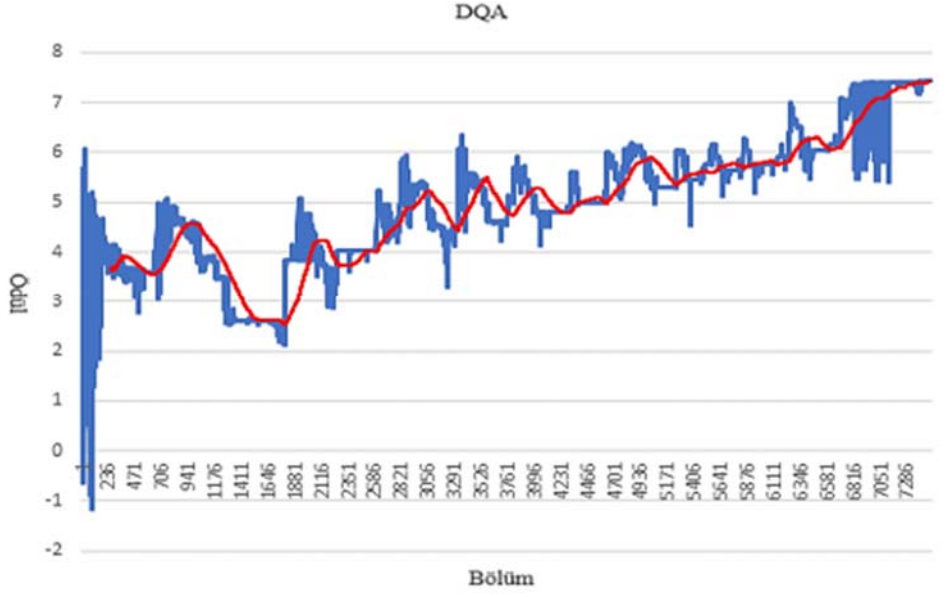
Tablo 5'de görüldüğü gibi kazanılan ortalama ödül 3,24 ile en düşük PD, 3,47 ile MÖK ve 4,99 ortalama ile DQA yöntemi olmuştur. DQA algoritması diğer iki yöntemle göre daha başarılı olmuştur. Fakat bu algoritmanın öğrenme süresi PD ve MÖK'e göre çok uzundur. DQA toplam 7500 bölüm eğitimin sonlarına doğru istenilen oranda öğrenme gerçekleşmiştir. DQA algoritmasının öğrenme süresi 59,41 saat sürmüştür. MÖK ile test süresi 100 bölüm için 54 dakika, PD ile test süresi 100 bölüm için 55 dakika sürmüştür. Eğitim süresinin uzun olması DQA algoritmasının dezavantajı olarak karşımıza çıkmaktadır.



Şekil 14. MÖK kontrol sonuçları (MPC control results)



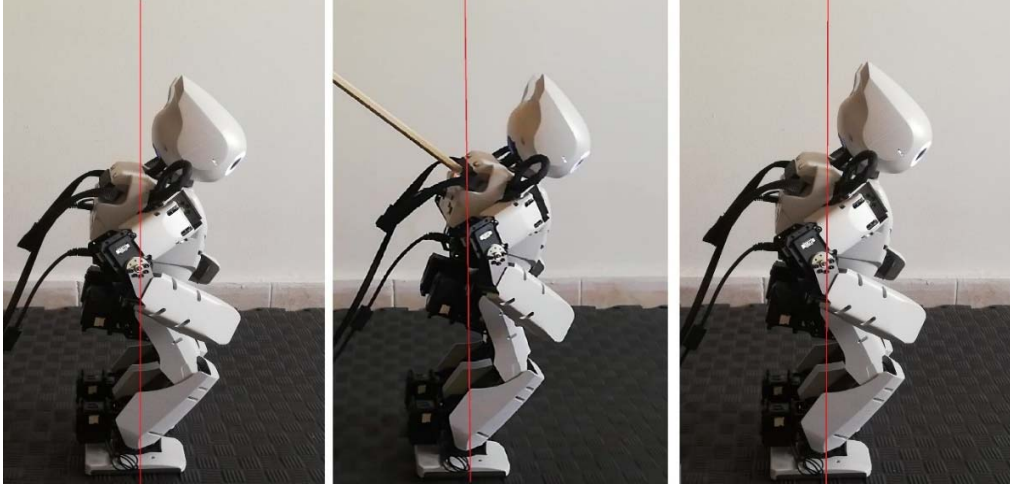
Şekil 15. PD ve MÖK kontrol sonuçlarının karşılaştırılması (Comparison of PD and MPC control results)



Şekil 16. DQA algoritması sonuçları (DQN algorithm results)

Tablo 5. Algoritma sonuçlarının karşılaştırılması (Comparison of algorithm results)

Kontrol Algoritması	Toplam Bölüm	Başarıya Ulaştığı Yaklaşık Bölüm	En Büyük Ödül	Toplam Aldığı Ödül	Ortalama Ödül
PD	100	100	7,5	324,265	3,24
MÖK	100	100	7,5	347,068	3,47
DQA	7500	7200	7,5	37483,51	4,99



Şekil 17. Gerçek dünya test görüntüsü (Real world test image)

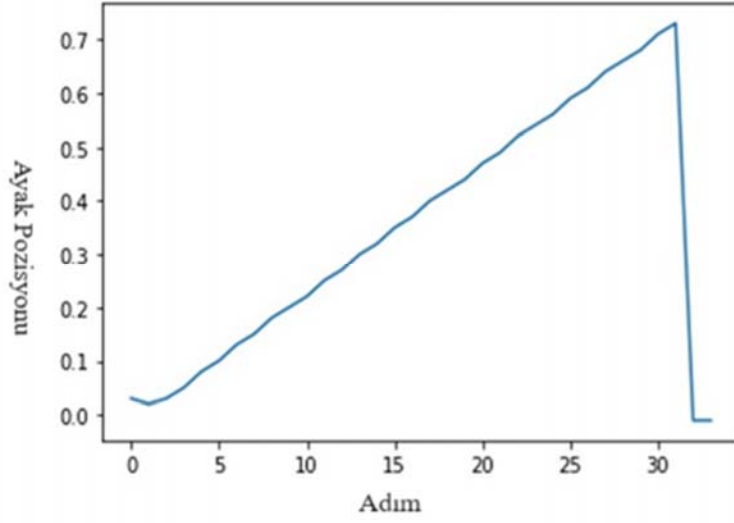
DQA algoritması ile 7500 eğitim bölümü gerçekleştirilmiştir. MÖK ve PD kontrolörleri ile 100'er bölüm gerçekleştirilmiştir. DQA algoritmasının son 100 eğitim ödülleri ile MÖK ve PD kontrolörlerinin 100 bölümlerinin karşılaştırılması Tablo 6'da verilmiştir. Ödül sonuçlarında PD %43,23 MÖK %46,27 ve DQA %99,46 başarı elde ettiği görülmektedir.

Uygulamanın gerçek ortam testi Şekil 17'de verilmiştir. Tasarlanan itme-kurtarma kontrolörü, ayak bileği stratejisi kullanılarak geliştirilmiştir. Robotun dış kuvvetlere karşı ürettiği ayak bileği tork değerleri kaydedilmiştir. Dış kuvvetin büyüklüğüne bağlı olarak tepki vererek kararlı olduğu denge noktasına yaklaşmıştır.

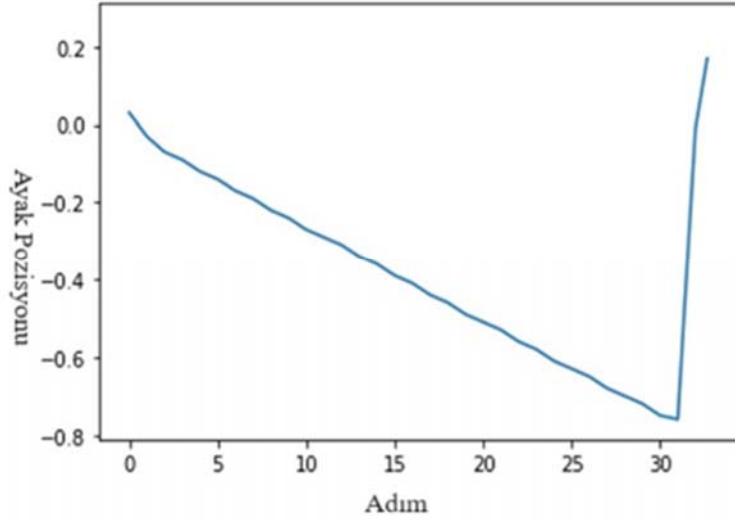
Tablo 6. Algoritmaların son 100 bölümlerinin karşılaştırılması sonuçlarının karşılaştırılması (Comparison of the results of the last 100 sections of the algorithms)

Kontrol Algoritması	Toplam Aldığı Ödül	Ortalama Ödül
PD	324,265	3,24
MÖK	347,068	3,47
DQA	746,743	7,46

Webots simülasyon ortamında uygulamanın gerçek dünyada robota uygulanabilmesi için robot ile bağlantı kurulması gerekmektedir. Robot kontrol ekranı "Robot Penceresini Göster" ile açılır. Bu ekranda



Şekil 18. Arkadan uygulanan kuvvetten sonra ayak bileği pozisyonu (Ankle position after force applied from behind)



Şekil 19. Önden uygulanan kuvvetten sonra ayak bileği pozisyonu (Ankle position after anteriorly applied force)

IP adresi, kullanıcı adı, şifre bilgileri girilerek ve uzaktan kumandayı başlat butonuna basılarak bağlantı yapılır. Veya simülasyon ortamında geliştirilen uygulamalar bir internet kablosu üzerinden gerçek robota aktarılır. Gerçek ortamda, bir çubuk yardımıyla robota itmeler uygulanmıştır. Çalışmanın hem simülasyon hem de gerçek ortam testlerinde 10 N ile -10 N arasındaki rastgele büyüklüklerdeki kuvvetler uygulanmıştır. Robotun simülasyon ortamında ve gerçek testlerde aldığı darbelerle karşı ayakta kalabilecek şekilde dengede kaldığı görülmektedir. Gerçek dünyada yapılan test videosuna <https://youtu.be/nWG53hVrAE8> linkinden ulaşabilirsiniz. Çubuk yardımıyla itme kuvveti uygulandıktan sonra robotun ayak pozisyon ve tork bilgileri kaydedilmiştir.

Robota uygulanan kuvvetten sonra ayak bileğinin aldığı pozisyonlar Şekil 18 ve Şekil 19'da sunulmuştur. Dikey eksen ayak bileği ve yatay eksen basamağı ifade eder. Şekil 18 arkadan uygulanan kuvveti, Şekil 19 ise önden uygulanan kuvveti göstermektedir. Şekillerde görüldüğü gibi robot, yıkıcı kuvvetten sonra denge noktasına yaklaşır ve stabilize gösterir. Gerçek dünyada ve simülasyon ortamında alınan sonuçların benzerdir.

#### 4. Sonuçlar (Conclusions)

Bu çalışmada, hem simülasyon hem de gerçek ortamda gerçekleştirilmiştir. Uygulamada 20 serbestlik derecesine sahip olan bir insansı robota uygulanan itmeye karşı dengede kalabilmesini sağlayan kontrol sistemleri gerçekleştirilmiştir. Çalışmanın uygulama aşamasında Robotis-OP2 insansı robotu kullanılmıştır. İnsansı robotun dış tepkiler sonucunda tekrar denge konumuna getirmek için farklı kontrol yöntemleri ve algoritmalar önerilmiştir. Klasik kontrol yöntemi olarak PD kontrolör, tahmine dayalı kontrol yöntemi olan MÖK ve derin pekiştirmeli öğrenme algoritmalarından DQA kullanılarak itme-kurtarma kontrolörleri tasarlanmıştır. Webots ortamında uygulamalar gerçekleştirilmiştir. Çalışmada itme-kurtarma kontrolörlerinden ayak bileği stratejisi temel alınarak uygulamalar yapılmıştır. Robota önden ve arkadan rastgele itme kuvvetleri uygulanıp dengede kalması sağlanmıştır.

İnsansı robotun uygulanan üç yöntemindeki testlerde başarılı sonuçlar aldığı gözlemlenmiştir. Bu yöntemlerden başarı oranı en düşük yöntem PD olmuştur. PD yöntemini MÖK takip etmiştir. DQA ile

daha başarılı sonuçlar alındığı tespit edilmiştir. Önerilen kontrolör yöntemlerinde, en iyi test sonucu DQA algoritması ile olduğu gözlemlenmiştir. Eğitim süresi uzun olması DQA algoritmasının bir olumsuz tarafı olarak karşımıza çıkmaktadır. Dış itme, simülasyonda bir kütüphane oluşturularak, gerçek ortamda bir çubuk yardımıyla robota uygulanmıştır. Robotun önden ve arkadan itmelere karşı dengede kalmayı öğrendiği gözlemlenmiştir. Simülasyon ve gerçek ortam sonuçları paralellik göstermektedir.

Gelecekteki çalışmalarda, itme kurtarma kontrolörlerinden kalça stratejisi, adım stratejileri ve bunların birleşimleri birlikte kullanılarak daha güçlü bir kontrolör tasarlanması planlanmaktadır.

#### Kaynakça (References)

- Stephens B., Humanoid push recovery In Humanoid Robots, 2007 7th IEEE-RAS International Conference, 589–595, 2007.
- Stephens B., Push recovery control for force-controlled humanoid robots. PhD thesis, Carnegie Mellon University Pittsburgh, Pennsylvania USA, 2011.
- Huang Q., Yokoi K., Kajita S., Kaneko K., Aral H., Koyachi N., Tanie K., Planning walking patterns for a biped robot, IEEE Transactions on Robotics and Automation, 17 (3), 280–289, 2001.
- Takanishi A., Lim H., Biped walking robots created at Waseda University: WL and Wabian family, Philosophical Transactions of the Royal Society Series A, 365 (1850), 49–64, 2007.
- Aftab Z., Robert T., Pierre-Brice W., Ankle, hip and stepping strategies for humanoid balance recovery with a single Model Predictive Control scheme. IEEE-RAS International Conference on Humanoid Robots, Osaka, Japan. 159–164, Nov 2012.
- Ghassemi P., Masouleh M.T., Kalhor A., Push Recovery for NAO Humanoid Robot, 2014 Second RSI/ISM International Conference on Robotic and Mechatronics(ICRoM), IEEE, 175–181, 2014.
- Shafiee-Ashtiani M., Yousefi-Koma M., Shariat-Panahi M., Khadiv M., Push Recovery of a Humanoid Robot Based on Model Predictive Control and Capture Point, Proceedings of the 4th International Conference on Robotics and Mechatronics October 26–28, 2016.
- Shafiee-Ashtiani M., Yousefi-Koma M., Mirjalili A., Maleki R., Karimi H., Push Recovery of a Position-Controlled Humanoid Robot Based on Capture Point Feedback Control, ICRoM, 2017.
- Kim H., Seo D., Kim D., Push Recovery Control for Humanoid Robot Using Reinforcement Learning, 2019 Third IEEE International Conference on Robotic Computing (IRC), Naples, Italy, 488–492, 2019.
- Missura M., Behnke S., Omnidirectional capture steps for bipedal walking, In Proceedings of Humanoids, 401–408, Atlanta, GA, 2013.
- Missura M., Behnke S., Online learning of foot placement for balanced bipedal walking, IEEE-RAS International Conference on Humanoid Robots, 322–328, 2015.
- Melo D.C., Maximo M., Da Cunha A., Push Recovery Strategies through Deep Reinforcement Learning, 2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE), Natal, Brazil, 147–154, 2020.
- Li S., Ping-Huan K., Lin-Han C., Chia-Ching H., Po-Chien L., Hao-Ping H., Chien-Hsin C., Yi-Ting H., Wen-Hsun L., Fuzzy Double Deep Q-Network-Based Gait Pattern Controller for Humanoid Robots, IEEE Transactions on Fuzzy Systems, 2022.
- Robert C., Sotiropoulos T., Waeselynck H., Guiochet J., Vernhes S., The virtual lands of Oz: testing an agrirobot in simulation, Empirical Software Engineering, Springer Verlag, 2020.
- Prakash V., Saran R., An Enhanced Coding Algorithm for Efficient Video Coding, Journal of the Institute of Electronics and Computer, 1, 28–38, 2019.
- Mnih V., Kavukcuoglu K., Silver D., Human-level control through deep reinforcement learning, Nature, 518:529–533 2015.
- Behnke S., Online trajectory generation for omnidirectional biped walking, Proceedings - IEEE International Conference on Robotics and Automation, 214–221, 2006.
- Seung-Joon Y., Online learning of a full body push recovery controller for omnidirectional walking. Humanoid Robots 2011 11th IEEE-RAS International Conference on. IEEE, 2011.
- Seung-Joon Y., Whole-body balancing walk controller for position controlled humanoid robots, International Journal of Humanoid Robotics 13.01, 2016.
- Yang C., Komura T., Li Z., Emergence of humancomparable balancing behaviours by deep reinforcement learning, 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), Birmingham, 372–377, 2017.
- Shu-Yin C., W. Jin-Long W., Posture Control for Humanoid Robot on Uneven Ground and Slopes Using Inertial Sensors, Advances in Mechanical Engineering, 2020.
- Yang S., Chen F., Zhang L., Cao Z., Wensing P., Liu Y., Pang J., Zhang W., Reachability-based Push Recovery for Humanoid Robots with Variable-Height Inverted Pendulum, 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021.
- Schuller R., Messesan G., Engelsberger J., Lee J., Ott C., Online Centroidal Angular Momentum Reference Generation and Motion Optimization for Humanoid Push Recovery, IEEE Robotics and Automation Letters, 2021.
- Messesan G., Engelsberger J., Ott C., Online DCM Trajectory Adaptation for Push and Stumble Recovery during Humanoid Locomotion, 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021.
- Ong H.Y., Chavez K., Hong A., Distributed Deep Q-Learning, Computer Science, 2015.
- Özupak Y., Design and Analysis of Different Transformer Models Used for Charging Electric Vehicles with Wireless Power Transmission System by Finite Element Method, Journal of Gazi University Faculty of Engineering and Architecture, 39 (2), 1113–1122, 2023.
- Huang Q., Dong C., Yu Z., Chen X., Li Q., Chen H., Liu H., Resistant Compliance Control for Biped Robot Inspired by Humanlike Behavior, IEEE/ASME Transactions on Mechatronics, 2022.
- Zhu K., Zhang T., Deep reinforcement learning based mobile robot navigation: A review, Tsinghua Science and Technology, 26 (5), 674–691, Oct. 2021.
- Assman T., Nijmeijer H., Takanishi A., Hashimoto K., Biomechanically motivated lateral biped balancing using momentum control, Eindhoven: Eindhoven University of Technology, Traineeship report. - DC 2011.035, 2012.
- Komura T., Leung H., Kudoh S, Kuffner S., A feedback controller for biped humanoids that can counteract large perturbations during gait, in Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 1989–1995, 2005.
- Özupak Y., Analysis and experimental verification of efficiency parameters affecting inductively coupled wireless power transfer systems, Heliyon, 10 (5), 2024.
- Pratt J., Carff J., Drakunov S., and Goswami A., Capture point: A step toward humanoid push recovery, in Proceedings of the 2006 6th IEEE-RAS International Conference on Humanoid Robots, 200–207, 2006.
- Pratt J., Capture point: A step toward humanoid push recovery, 2006 6th IEEE-RAS international conference on humanoid robots. IEEE, 2006.
- Oberdieck R., Pistikopoulos E. N., Explicit hybrid modelpredictive control: The exact solution, Automatica, 58, 152–159, 2015.
- Stephens B. J., State Estimation for Force-Controlled Humanoid Balance using Simple Models in the Presence of Modeling Error. In IEEE International Conference on Robotics and Automation, 2011.
- Stephens B. J., Atkeson C., Push recovery by stepping for humanoid robots with force controlled joints. 2010 10th IEEE-RAS International Conference on Humanoid Robots, 52–59, 2010.
- Mantilla L.C., Junca M. J., Deep Q-Learning, Universidad de Los Andes, Dogota, Colombia, 2021.

