İSTANBUL TEKNİK ÜNİVERSİTESİ
ELEKTRİK ELEKTRONİK FAKÜLTESİ

MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANILARAK BİR DÖVÜŞEN ROBOTUN EĞİTİLMESİ

KONTROL VE OTOMASYON MÜHENDİSLİĞİ TASARIMI II

Muhammet Işık

Kontrol ve Otomasyon Mühendisliği

Doç. Dr. Ahmet Onat

Ocak 2025

İSTANBUL TEKNİK ÜNİVERSİTESİ
ELEKTRİK ELEKTRONİK FAKÜLTESİ

MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANILARAK BİR DÖVÜŞEN ROBOTUN EĞİTİLMESİ

KONTROL VE OTOMASYON MÜHENDİSLİĞİ TASARIMI II

Muhammet Işık
(040120447)

Kontrol ve Otomasyon Mühendisliği

Doç. Dr. Ahmet Onat

Ocak 2025

Bu tezi bugüne gelmemde büyük emeği ve destekleri olan başta değerli eşim ve kızım olmak üzere, kıymetli anne babama ve her biri bu memleket için ayrı ayrı paha biçilemez birer değer olan kıymetli bölüm hocalarıma, son olarak da memleketin geleceği için çalışmanın bedelini canıyla ödemiş tüm rütbeli ve rütbesiz şehitlerimize ithaf ediyorum.

ÖNSÖZ

İTÜ'nün kapısından yıllar önce elimi kolumu sallayarak girdiğimde hiçbir şeyden haberi olmayan bir genç idim. Birkaç ay sıkı çalışmayla yerleşmiştim. İnsan emek vermeden elde ettiğinin, kıymetini de bilmiyor. Süreç içerisinde derslere adapte olamayınca odağım kaydı ve ticaretle ilgilenmeye başladım, öyle oldu ki küçücük işlerim sonradan tüm Türkiye'de kurumsal ve bireysel müşterilere hizmet veren bir işletme haline geldi.

Zaman ilerledikçe, bilişim ve teknik servis alanındaki tecrübelerimin yanına bir lisans eğitimi eklemeden memlekete kalbimdeki gibi hizmet edemeyeceğime üzülerek ikna oldum. Faaliyetlerimi yavaşlattım ve okuluma geri dönerek eğitimimi tamamladım. Şu çalışma esnasında verdiğim mücadeleyi, çektiğim sıkıntı stresi evvel hayatımın hiçbir aşamasında çekmediğimi samimiyetle söylemek isterim.

Akademik anlamda emek verip tadını aldıkça insan çalışmanın ve üretmenin önemini daha iyi kavrıyor, emeği hiçe sayanlara daha da düşmanlaşıyor. Şimdi ise yüksek lisans hedefleyerek akademide ilerleme gayreti edinmiş vaziyetteyim.

Pişmanlık, sahibine fayda vermez ama belki başkasına ibret olur, ışık olur. Genç kardeşlerime tavsiyem odur ki, okudukları okullardan sadece geçip gitmesinler, iz bıraksınlar. Her imkanı kullansınlar, projeler yapsınlar. Dinlenmedikleri her günü okullarında geçirsinler. Kendileri için bu okullar işlesin diye ayırılan bütçeleri hor görmesinler, israf etmesinler. Destek istemeyi, vermeyi ve teşekkür etmeyi öğrensinler.

Tezin hazırlanması esnasında yoğun bir şekilde insan ve makine beyni kullanımı yapılmış olmakla birlikte, sonuçlar sonradan da kullanılıp üstüne yeni bağlamlar inşa edilebilecek ve ilerletilebilecek şekilde en genel örneklerden birisi olan ters sarkaç sistemi üzerinde tasarlanmıştır. Çalışmam sırasında bana rehberlik eden, bilgi ve tecrübelerini esirgemeyen danışman hocam Sayın Ahmet Onat'a en içten teşekkürlerimi sunarım. Ayrıca, araştırmalarım boyunca her zaman yanımda olan aileme teşekkür eder, okurlara faydalı olmasını dilerim.

Ocak 2025 Muhammet Işık

İÇİNDEKİLER

Sayfa

KISALTMALAR

DRL **:** Deep Reinforcement Learning
YSA **:** Yapay Sinir Ağı
ANN **:** Artificial Neural Network
DQN **:** Deep Q Network
FIFO **:** First in First Out
MSE **:** Mean Squared Error
ReLU **:** Rectified Linear Unit
Adam **:** Adaptive Movement Estimation

SEMBOLLER

$\boldsymbol{\lambda}$	**:** Poisson rastgelelik katsayısı
ε	**:** Keşif orasnı
α	**:** Öğrenme oranı
γ	**:** İlerideki ödüllerin önem oranı
$\mathbf{x}$	**:** Araç konumu
$\boldsymbol{\theta}$	**:** Sarkaç açısı

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANILARAK BİR DÖVÜŞEN ROBOTUN EĞİTİLMESİ

ÖZET

Bu çalışma, dinamik denge kontrolü ve yapay zekâ temelli kontrol stratejilerinin gelişimini inceleyen kapsamlı bir araştırmayı içermektedir. Çalışmada ters sarkaç ve çift sarkaç sistemleri temel alınarak, bu sistemlerin yapay zekâ tabanlı kontrol yöntemleriyle denge ve etkileşim süreçleri detaylı bir şekilde analiz edilmiştir. Ters sarkaç sistemi, mühendislikte temel bir kontrol problemi olarak değerlendirilmiş ve derin pekiştirmeli öğrenme (DQN) algoritmasıyla sistemin kontrolü sağlanmıştır. Çalışma, robotik sistemlerin kontrol stratejilerini geliştirme ve optimize etme amacını taşımaktadır.

Araştırma, dinamik etkileşimler yoluyla robotik sistemlerin hem dengeyi koruma hem de rakip sisteme müdahale etme yeteneklerini geliştirmeyi hedeflemektedir. Çift ters sarkaç sisteminin modeli, birbirine darbe uygulayarak rakibin dengesini bozmayı amaçlayan iki robotik sistemi ele almaktadır. Her iki robot, aynı yapay sinir ağı ile kontrol edilmekte ve bu durum, sistemlerin birbirleriyle etkileşim içinde öğrenmelerini sağlamaktadır. Epsilon-greedy stratejisi ve deneyim belleği mekanizmaları, robotların çevreyi keşfetme ve öğrenme süreçlerini optimize etmek amacıyla kullanılmıştır. DQN algoritması ile robotların stratejik karar alma yetenekleri geliştirilmiş hem bireysel denge kontrolü hem de rekabetçi senaryolarda performansları değerlendirilmiştir.

Ters sarkaç ve çift sarkaç sistemlerinin dinamik yapıları ve kontrol problemleri ele alınmış, bu sistemlerin otomasyon ve robotik uygulamalardaki önemi vurgulanmıştır. Takviyeli öğrenme yöntemleri, dinamik ve karmaşık sistemlerin kontrolünde sağladığı avantajlar nedeniyle araştırmanın temelini oluşturmuştur. DQN algoritması, derin öğrenme ve klasik Q-öğrenmeyi birleştirerek karmaşık durum-aksiyon uzaylarında etkili bir çözüm sunmaktadır. Bu çalışmada, sistemin denge kontrolü ve rakip robotla etkileşim süreçleri, detaylı matematiksel modeller ve simülasyonlar aracılığıyla incelenmiştir.

Literatüde, ters sarkaç sistemleri ve takviyeli öğrenme algoritmaları üzerine yapılan önceki çalışmalar ele alınmıştır. Ters sarkaç sistemi, denge kontrolü ve stabilizasyon problemleri için temel bir model olarak geniş bir alanda kullanılmaktadır. Takviyeli

öğrenme yöntemlerinin, özellikle derin pekiştirmeli öğrenme algoritmalarının, karmaşık ve dinamik sistemlerin kontrolündeki başarıları çeşitli çalışmalarda kanıtlanmıştır. Bu bağlamda, DQN algoritmasının rekabetçi ortamlarda strateji geliştirme kabiliyetleri ve performansı analiz edilmiştir.

Yöntem kısmında, ters sarkaç sisteminin matematiksel modeli ve simülasyon ortamı detaylı bir şekilde açıklanmıştır. Sistem, Newton'un hareket yasalarına dayalı bir diferansiyel denklem seti ile modellenmiş ve Python ortamında simüle edilmiştir. Eğitim sürecinde, DQN algoritması kullanılarak robotların ödül tabanlı strateji geliştirme süreçleri optimize edilmiştir. Deneyim belleği, epsilon-greedy stratejisi ve ödül fonksiyonu gibi pekiştirmeli öğrenme bileşenleri, sistemin performansını artırmak için dikkatle tasarlanmıştır. Ödül fonksiyonları, sarkacın denge pozisyonuna yakınlığını ve enerji verimliliğini ödüllendirirken, gereksiz güç tüketimi ve kararsız durumları cezalandırmıştır.

Sonuçların analizi için tek ters sarkaç ve çift ters sarkaç sistemlerinin eğitim süreçleri değerlendirilmiştir. Tek sarkaç sisteminde, sistemin dengede kalma süresi artırılmış ve bozucu kuvvetler karşısında dayanıklılığı test edilmiştir. Poisson dağılımı ile modellenen rastgele bozucu kuvvetler, sistemin gerçek dünya uygulamalarında karşılaşabileceği dış etkileri simüle etmek için kullanılmıştır. Eğitim sürecinin başlangıcında bozucu kuvvetler, robotların performansını olumsuz etkilese de, sistem bu zorluklara hızlı bir şekilde adapte olmuştur. Eğitim ilerledikçe robotlar, bozucu etkiler altında bile dengede kalmayı başarmış ve kararlılığını artırmıştır.

Çift sarkaç sisteminde ise, robotların birbirleriyle etkileşimleri incelenmiştir. Çift ters sarkaç simülasyonu, iki robotun hem kendi dengesini koruma hem de rakip robotun dengesini bozma stratejilerini öğrenmesini hedeflemiştir. Bu süreçte, ortak bir yapay sinir ağı kullanılmış ve her iki robotun denge ve saldırı stratejileri eş zamanlı olarak optimize edilmiştir. Bu yaklaşım, robotların çoklu görev öğrenme kapasitesini test etmiş ve başarılı sonuçlar elde edilmiştir.

Sonuçlar, geliştirilen kontrol stratejilerinin dinamik sistemlerde yüksek performans sergilediğini göstermektedir. Tek ve çift sarkaç sistemlerinin eğitim süreçleri sonunda, sistemlerin dengede kalma süreleri artırılmış ve bozucu kuvvetler karşısında dayanıklılıkları kanıtlanmıştır. Çift sarkaç sisteminde, robotların rekabetçi senaryolarda etkili stratejiler geliştirdiği gözlemlenmiştir. Ayrıca, ödül

fonksiyonlarının ve hiperparametrelerin optimize edilmesi, sistemin kararlı ve enerji verimli bir şekilde çalışmasını sağlamıştır.

Bu çalışma, robotik sistemlerin kontrolünde yapay zeka tabanlı yöntemlerin etkinliğini ve potansiyelini ortaya koymaktadır. Çalışmanın bulguları, hem akademik hem de endüstriyel uygulamalarda yol gösterici bir temel sunmakta ve kontrol stratejilerinin geliştirilmesi için önemli bir kaynak oluşturmaktadır. Geliştirilen sistemlerin, dinamik ve rekabetçi ortamlar için adaptasyon yetenekleri, gelecekteki robotik uygulamalar için umut verici sonuçlar sunmaktadır.

TRAINING A FIGHTING ROBOT USING MACHINE LEARNING TECHNIQUES

SUMMARY

1 line spacing must be set for summaries. For B.Sc. projects, the summary in Turkish must have 300 words minimum, whereas the summary in English must have 300 This study presents a comprehensive investigation into the development of dynamic balance control and artificial intelligence-based control strategies. The research focuses on inverted pendulum and double pendulum systems, analyzing their balance and interaction processes in detail using AI-based control methods. The inverted pendulum system is regarded as a fundamental control problem in engineering and is controlled using the Deep Q-Network (DQN) algorithm. The study aims to enhance and optimize control strategies for robotic systems.

The research seeks to improve robotic systems' abilities to maintain balance and intervene with opposing systems through dynamic interactions. The double inverted pendulum model addresses two robotic systems attempting to disrupt each other's balance by applying impulses. Both robots are controlled by the same neural network, enabling them to learn through interaction. The epsilon-greedy strategy and experience replay mechanisms are utilized to optimize the robots' exploration and learning processes. The DQN algorithm develops strategic decision-making capabilities, evaluating both individual balance control and performance in competitive scenarios.

In the introduction, the dynamic structures and control problems of inverted and double pendulum systems are discussed, emphasizing their significance in automation and robotics applications. Reinforcement learning methods form the foundation of the research due to their advantages in controlling dynamic and complex systems. By combining deep learning with classical Q-learning, the DQN algorithm provides an effective solution for complex state-action spaces. In this study, the balance control of the system and its interaction with competing robots are examined through detailed mathematical models and simulations.

The literature review highlights previous studies on inverted pendulum systems and reinforcement learning algorithms. The inverted pendulum system is widely used as a fundamental model for balance control and stabilization problems. Reinforcement learning methods, particularly deep reinforcement learning algorithms, have

demonstrated success in controlling complex and dynamic systems. Within this context, the ability of the DQN algorithm to develop strategies and its performance in competitive environments are analyzed.

In the methodology section, the mathematical model and simulation environment of the inverted pendulum system are explained in detail. The system is modeled using a set of differential equations based on Newton's laws of motion and simulated in a Python environment. During training, the DQN algorithm is used to optimize the robots' reward-based strategy development processes. Components such as experience replay, epsilon-greedy strategy, and reward functions are carefully designed to enhance system performance. Reward functions encourage proximity to the pendulum's equilibrium position and energy efficiency while penalizing unnecessary power consumption and instability.

The results analysis evaluates the training processes of single and double inverted pendulum systems. In the single pendulum system, the balance retention time was increased, and resilience to disruptive forces was tested. Random disruptive forces modeled using a Poisson distribution were employed to simulate external effects encountered in real-world applications. Although disruptive forces initially negatively affected the robots' performance, the system quickly adapted to these challenges. As training progressed, the robots achieved balance even under disruptive influences and improved their stability.

In the double pendulum system, interactions between robots were examined. The double inverted pendulum simulation aimed to teach two robots to maintain their own balance while disrupting the balance of the opposing robot. A shared neural network was employed, and the balance and attack strategies of both robots were simultaneously optimized. This approach tested the robots' multi-task learning capacity, yielding successful outcomes.

The results demonstrate that the developed control strategies deliver high performance in dynamic systems. At the conclusion of the training processes for single and double pendulum systems, the systems' balance retention times were extended, and their resilience to disruptive forces was proven. In the double pendulum system, the robots exhibited effective strategy development in competitive scenarios. Furthermore,

optimizing reward functions and hyperparameters enabled the system to operate stably and efficiently.

This study highlights the effectiveness and potential of AI-based methods in controlling robotic systems. The findings provide a foundational guide for academic and industrial applications, offering valuable resources for developing advanced control strategies. The developed systems' adaptive capabilities in dynamic and competitive environments present promising results for future robotic applications.

1. GİRİŞ

Ters sarkaç ve çift sarkaç sistemleri, dinamik sistemlerin kontrolü ve denge problemlerinin anlaşılması için mühendislikte sıklıkla ele alınan temel modellerdir. Ters sarkaç sistemi, bir sarkacın dik konumda tutulmasını sağlarken, çift sarkaç sistemi bu zorluğu bir adım öteye taşıyarak daha karmaşık ve kaotik dinamik davranışlar sergiler. Her iki sistem de hareket dengeleme ve stabilizasyonu açısından otomasyon ve robotik uygulamalarında kritik bir test alanı oluşturur. Bu tür sistemler, kontrol algoritmalarının etkinliğini test etmek ve geliştirmek için vazgeçilmez birer platform sunmaktadır (Kirk, 2004).

Son yıllarda, takviyeli öğrenme (Reinforcement Learning, RL) algoritmaları, robotik sistemlerin kontrolü ve otonom karar verme problemlerinde önemli bir rol üstlenmiştir. RL, bir sistemin çevresiyle etkileşim kurarak, belirli ödül-fonksiyonlarına dayalı optimal bir politika öğrenmesini sağlar. Özellikle karmaşık ve kaotik sistemlerin kontrolünde RL tabanlı yöntemlerin etkin olduğu çeşitli çalışmalarla gösterilmiştir (Sutton & Barto, 2018).

Bu çalışma, çift sarkaç sisteminden farklı olarak, ters sarkaç prensibine dayalı ve birbirine darbe uygulayan iki robotun yer aldığı bir sistem tasarımını ve kontrolünü hedeflemektedir. Robotların, hem kendi dengesini sağlarken hem de karşı tarafa kontrollü bir şekilde darbe uygulaması üzerine bir senaryo oluşturulmuştur. Çalışmanın temelinde, iki robotun da aynı yapay sinir ağı (NN) ile kontrol edilerek birbirini ezmeden optimal çalışma stratejilerinin öğrenilmesi yatar. Bu kapsamda, DQN algoritması kullanılarak robotların kontrolü ve dövüş stratejilerinin geliştirilmesi amaçlanmıştır (Mnih et al., 2015).

Yaptığım çalışma, RL tabanlı kontrol sistemlerinin robotik uygulamalardaki potansiyelini ortaya koymayı, aynı zamanda robotlar arası etkileşimin dinamik denge ve kontrol performansına etkilerini incelemeyi hedeflemektedir. Sunulan sistem, hem robotik kontrol stratejilerinin geliştirilmesine katkı sağlayacak hem de RL algoritmalarının performans sınırlarının belirlenmesine olanak tanıyacaktır.

1.1 Tezin Amacı

Bu çalışmanın temel amacı, birbirine darbe uygulayarak denge bozmayı hedefleyen iki robotik sistemin kontrol stratejilerini geliştirmek ve optimize etmektir. Ters sarkaç prensibi üzerine kurulu olan bu sistemler, birbirleriyle dinamik bir etkileşim içinde çalışarak hem kendi dengesini korumaya hem de rakip sistemin dengesini bozmaya çalışmaktadır. Bu yapı, kontrol teorisi ve oyun teorisinin birleştiği bir problem alanı sunarak robotik sistemlerin karmaşık etkileşimler içindeki davranışlarını modellemek için uygun bir test senaryosu sağlamaktadır.

Robotların kontrol stratejilerini öğrenmesi, derin pekiştirmeli öğrenme (Deep Q-Network, DQN) algoritması ile gerçekleştirilmiştir. DQN, çevreden aldığı girdilere dayanarak en uygun aksiyonları seçmeyi öğrenen bir değer tabanlı algoritmadır. Bu yöntem, robotların sürekli değişen ve rekabetçi bir ortamda stratejik kararlar almasına olanak tanır. Her bir robot hem kendi denge durumunu hem de rakip robotun hareketlerini analiz ederek optimum hareket planını belirler.

DQN algoritması, epsilon-greedy stratejisi ile çalışır. Bu strateji, robotların başlangıçta çevreyi keşfetmesine olanak tanırken, öğrenme sürecinin ilerleyen aşamalarında daha önce öğrendikleri en iyi aksiyonları tercih etmelerini sağlar. Böylece, keşfetme ve istismar arasında dengeli bir öğrenme süreci gerçekleştirilir (Mnih et al., 2015).

Bu çalışma, rekabetçi ve dinamik senaryolarda DQN algoritmasının etkinliğini analiz etmeyi ve robotların karmaşık görevler karşısında öğrenme performanslarını değerlendirmeyi hedeflemektedir. Ayrıca, bu sistemler aracılığıyla kontrol algoritmalarının sınırları araştırılarak, robotik sistemlerin adaptasyon yeteneklerini artırma potansiyeli ortaya konulacaktır.

1.2 Literatür Araştırması

Literatürde ters sarkaç sistemleri ve takviyeli öğrenme yöntemleri üzerine geniş bir çalışma alanı bulunmaktadır. Ters sarkaç sistemleri, özellikle denge kontrolü ve kararlılık analizlerinde temel bir model olarak sıkça kullanılmaktadır (Kirk, 2004).

Çift sarkaç sistemlerinin daha karmaşık dinamikleri nedeniyle, bu tür sistemler kaotik davranışların analizi için önemli bir platform sunmaktadır (Strogatz, 2018).

Takviyeli öğrenme algoritmaları, dinamik ve karmaşık sistemlerin kontrolünde büyük bir potansiyel sunmaktadır. Sutton ve Barto (2018), RL'nin temel prensiplerini açıklarken, özellikle ödül tabanlı öğrenme mekanizmasının adaptif kontrol sistemlerindeki rolüne dikkat çekmiştir. DQN algoritması, derin öğrenme ile klasik Q-öğrenmeyi birleştirerek karmaşık durum-aksiyon uzaylarında etkili bir kontrol sağlamaktadır (Mnih et al., 2015).

Rekabetçi ortamlar için RL tabanlı kontrol yöntemleri üzerine yapılan çalışmalarda, çoklu ajan sistemlerinin koordinasyon ve strateji geliştirme yetenekleri incelenmiştir. Örneğin, Silver ve arkadaşları (2017), oyun teorisi ile RL'yi birleştirerek stratejik karar alma problemlerine çözümler sunmuştur. Bu bağlamda, robotlar arası etkileşim ve dinamik dengelerin kontrolü, RL'nin sınırlarını test etmek için ideal bir uygulama alanı sunmaktadır.

1.3 Hipotez

Aynı yapay sinir ağı ile kontrol edilen iki robot, hem kendi dengesini koruma hem de rakip robotun dengesini bozma görevlerini eş zamanlı olarak başarıyla gerçekleştirebilir. DQN algoritması, rekabetçi ve dinamik bir ortamda robotların stratejik karar alma süreçlerini optimize edebilir. Eğitim süreci sonunda, iki robot arasında sürekli etkileşim ile dengede kalma sürelerinin artırılabileceği hipotez edilmektedir.

2. ÇALIŞMANIN YÖNTEMİ

Çalışma kapsamında, sarkaç sisteminin kontrolü amacıyla derin pekiştirmeli öğrenme (Deep Reinforcement Learning, DRL) yöntemiyle hareket edilmektedir. Bu bağlamda, sarkaç dinamikleri ve belirli parametreler ile etkileşimde bulunan bir DQN adamı kullanılmaktadır.

2.1 DQN Sisteminin Çalışma Şekli ve Mantığı

DQN, Q-learning algoritmasının derin öğrenme ile birleştirilmesiyle geliştirilmiş bir pekiştirmeli öğrenme yöntemidir. DQN, bir adam (yapay zeka bireyi) ile çevresi arasındaki etkileşimi öğrenme sürecini optimize etmek için kullanılır. Bu süreç, aşağıdaki adımları takip eder.

2.1.1 Aksiyon seçimi: keşif ve sömürü

DQN, adamının çevresiyle etkileşimde bulunarak en iyi eylemi öğrenmesini sağlar. Adam, epsilon-greedy stratejisi kullanarak keşif ve sömürüyi dengeler. Bu strateji şu şekilde işler:

Keşif (Exploration): Eğer rastgele bir sayı, ε değerinden küçükse, adam rastgele bir eylem seçer. Bu, çevrenin farklı yönlerini keşfetmesini sağlar.

Sömürü (Exploitation): Eğer rastgele sayı ε değerinden büyükse, adam mevcut Q-değerlerine dayanarak en yüksek Q-değerine sahip eylemi seçer.

Matematiksel olarak a_t mevcut s_t durumunda alınacak aksiyon ve Q(s_t,a) mevcut durum içerisindeki her ayrı aksiyon için hesaplanan bir Q değeri olmak üzere epsilon-greedy stratejisi 2.1 numaralı denklemdeki gibi ifade edilir.

$$a_t = \begin{cases} Rastgele\ aksiyon & (Olasılık: \varepsilon) \\ arg\ max_a\ Q(s_t, a) & (Olasılık: 1-\varepsilon) \end{cases} \tag{2.1}$$

Epsilon değeri ise genellikle $\boldsymbol{\varepsilon}$ epsilon olacak ve $\boldsymbol{\varepsilon_d}$ çarpanı $\boldsymbol{\varepsilon}$ 'nin bölüm başına azalma hızı olacak şekilde 2.2 numaralı denklemdeki gibi azaltılmaktadır.

$$\varepsilon = \varepsilon_{min} + (\varepsilon_{max} - \varepsilon_{min}) \cdot \varepsilon_d{}^{episode} \tag{2.2}$$

2.1.2 Deneyim belleği

Deneyim Belleği, DQN algoritmasının verimliliğini artıran önemli bir mekanizmadır. Bu mekanizma, adam ile çevresi arasındaki her bir etkileşim sonucunda oluşan (durum, eylem, ödül, yeni durum) setlerini bir bellekte saklar. Bu kayıtlar, öğrenme sürecinde tekrar kullanılabilir ve algoritmanın bazı temel problemlerini çözmeye yardımcı olur. Deneyim belleği şu şekilde çalışır:

Deneyimlerin Saklanması: Adam, çevresiyle etkileşime geçtiğinde her bir etkileşimi deneyim belleğine kaydeder. Bu etkileşimler, sürekli olarak güncellenir ve bellekte bir öncelik sırasına göre düzenlenebilir. Bellek kapasitesi sınırlı olduğunda, en eski deneyimler yerine yenileri eklenir (FIFO mantığı).

Rastgele Örnekleme: Eğitim sırasında, bellekten rastgele küçük bir örnekleme grubu seçilir ve modelin öğrenmesi için kullanılır. Rastgele örnekleme, ardışık verilerden kaynaklanan korelasyonu azaltır ve daha dengeli bir öğrenme sağlar.

Geçmiş Deneyimlerden Öğrenme: Model, yalnızca mevcut durumdan değil, geçmişte yaşanan tüm durumlardan bilgi öğrenir. Bu, modelin veri kullanımını artırır ve nadir ancak kritik olayları öğrenmesine olanak tanır.

Deneyim belleği, veri verimliliğini artırarak modelin daha etkili bir şekilde öğrenmesini sağlar. Aynı deneyimler birden fazla kez kullanıldığı için daha az veriden daha fazla bilgi çıkarılabilir. Ayrıca, bellekten rastgele örnekleme, eğitim sırasında veriler arasındaki korelasyonu azaltır ve modelin daha dengeli bir öğrenme süreci geçirmesine yardımcı olur. Özellikle karmaşık veya nadir görülen durumların öğrenilmesi kolaylaşır, çünkü geçmişteki bu olaylar eğitim sırasında tekrar tekrar kullanılabilir.

Örneğin, bir adam bir labirentte hedefe ulaşmaya çalışırken, her adımda karşılaştığı durumlar ve aldığı eylemler bellekte saklanır. Adam, bu deneyimlerden öğrenerek gelecekteki adımlarını daha etkili bir şekilde planlayabilir. Rastgele örnekleme sayesinde hem nadir hem de sık karşılaşılan durumlar üzerinde dengeli bir eğitim gerçekleştirilir ve adam, karmaşık kararlar almayı öğrenir. Bu süreç, hedefe daha hızlı ve etkili bir şekilde ulaşmayı mümkün kılar.

2.1.3 Ödül hesaplama

Reinforcement learning (takviyeli öğrenme) sistemlerinde ödül fonksiyonu, adam için davranış rehberi niteliğindedir. Adam, her adımda gerçekleştirdiği eylemlerin sonucunda bir ödül alır. Sarkacın hedeflenen denge pozisyonuna (genellikle dik konum) olan yakınlığı, ödülün büyüklüğünü belirler. Eğer sarkaç denge pozisyonundan uzaklaşıyorsa, ödül fonksiyonu negatif bir değer sağlayarak cezalandırır; buna karşın, dengeye yaklaşan durumlarda pozitif ödüllerle adam teşvik edilir. Bu yaklaşım, adamı yalnızca dengeyi sağlamaya değil, aynı zamanda sistemi olabildiğince stabil tutmaya yönlendirir. Sarkacın belirli bir açıyı aşması gibi istenmeyen durumlar için ağır cezalar uygulanabilir, böylece öğrenme sürecinde bu tür durumlar minimize edilir (Sutton & Barto, 2018).

Ödül fonksiyonlarının doğru tasarımı, öğrenme sürecinin etkinliğini doğrudan etkiler. Örneğin, ödül fonksiyonuna açısal hız veya pozisyon gibi diğer durum parametrelerinin dahil edilmesi, daha hassas kontrol stratejilerinin geliştirilmesini sağlar. Sarkacın yalnızca dik pozisyona ulaşması değil, bu pozisyonda kalabilmesi de kritik olduğundan, ödül fonksiyonu sürekliliği dikkate almalıdır. Örneğin, bir sarkaç sisteminde, ödülün yalnızca açı ile değil, aynı zamanda hız veya enerji tüketimi gibi faktörlerle de ilişkilendirilmesi, gerçek dünya uygulamalarında daha dengeli ve enerji verimli sistemler geliştirilmesine olanak tanır (Mnih et al., 2015).

2.1.4 Q-Değeri güncelleme ve model iyileştirme

DQN, örneklem grubundan öğrenme yöntemi ile deneyim belleğinden rastgele örnekler alarak Q-değerlerini günceller. Örneklem grupları, modelin daha stabil ve hızlı öğrenmesini sağlar. Denklem 2.3 üzerinde görüldüğü gibi, r_t alınan ödül, γ gelecekteki ödüllere verdiğimiz önemin oranı ve Q(s_(t+1), a') da gelecek bir durum-aksiyon çifti için Q değeri olmak üzere y_t hedef Q-değeri şu şekilde hesaplanır:

$$y_t = r_t + \gamma \max_{a'} Q(s_{t+1}, a') \tag{2.3}$$

Q-değerleri, modelin çıktısı olan Q-fonksiyonu ile güncellenir. Bu işlem sırasında, kayıp fonksiyonu Denklem 2.4'te görüldüğü şekilde minimize edilir:

$$Loss(\theta) = E\left[\left(y_t - Q(s_t, a_t;\ \theta)\right)^2\right] \tag{2.4}$$

Denklem 2.4'te yer alan E, beklenen değer operatörüdür ve kayıp fonksiyonunun tüm öğrenme verileri üzerinden ortalamasını temsil eder. DQN algoritmasında bu veriler, genellikle deneyim tekrar oynatma (experience replay) yöntemi kullanılarak seçilir. Bu yöntem, algoritmanın geçmiş deneyimlerden öğrenmesine olanak tanır ve korelasyonlu verilerin model parametrelerini olumsuz etkilemesini engeller.

Bu kayıp fonksiyonu, mean squared error (MSE) olarak bilinir ve modelin tahmin ettiği Q-değerlerini hedef Q-değeriyle karşılaştırarak güncellenmesini sağlar.

Bütün bu bileşenleriyle birlikte değerlendirildiğinde DQN algoritması hem basit hem de güçlü bir pekiştirmeli öğrenme yöntemidir ve bu çalışmada sarkaç sisteminin kontrolü için etkili bir çözüm sunmaktadır.

2.1.5 Poisson dağılımıyla rastgele bozucu darbeler uygulanması

Poisson dağılımı, belirli bir zaman aralığında veya mekânsal bir bölgede belirli bir olayın kaç kez gerçekleşeceğini modelleyen olasılık dağılımıdır. Özellikle, olayların bağımsız olduğu durumlarda sıklıkla kullanılır. Dakikada bir çağrı merkezine gelen çağrı sayısını modellemek, bir üretim bandında belirli bir sürede meydana gelen hataların sayısını modellemek, trafik lambasında belirli bir süre içinde geçen araç sayısını modellemek gibi alanlarda sıkça kullanılır. Çalışma sonucunda dövüşen robotların modellenmesi hedeflendiği için, karşıdan gelecek olan rastgele darbelerin zamanı ve değerleri poisson dağılımıyla belirlenmiştir.

Poisson dağılımı, olayların meydana gelme oranı (λ) sabit olan durumlarda uygundur. Possion dağılımı, k belirli zamanda gerçekleşen olay sayısı ve e euler sayısı olmak üzere Denklem 2.5'te ifade edilmiştir.

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{2.5}$$

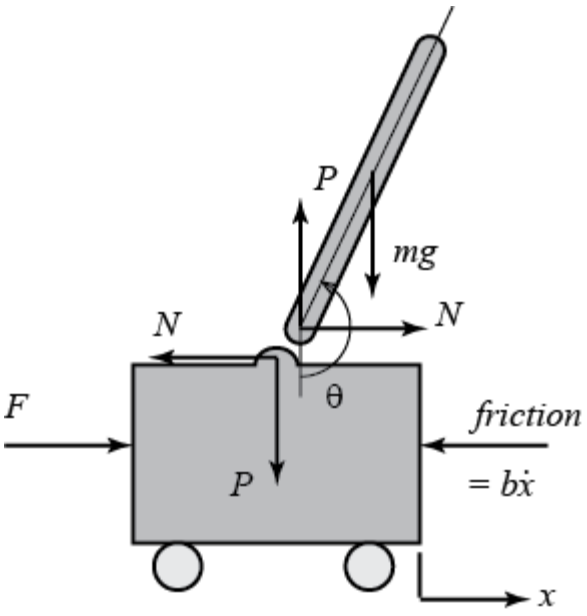
Adamın belirlediği kuvvetlere ek olarak rastgele bir bozucu kuvvet eklemek için Poisson dağılımını kullanıldı. Burada Poisson dağılımı, bozucu kuvvetlerin hangi sıklıkta gerçekleşeceğini modellemek için uygundur. Poisson dağılımı ile belirli bir zaman aralığında kaç kez bozucu kuvvet uygulanacağı hesaplandı. Her adımda bozucu kuvvetin etkin olup olmadığını kontrol etmek için λ parametresine göre bir olasılık belirlendi. Ardından bozucu kuvvet büyüklüğü rastgele bir aralıkta belirlendi ve son aşamada adamın dengesini bozma amacıyla denge kuvvetine eklendi.

3. MODEL

Çalışma esnasında kullanılan ters sarkaç ve çift sarkaç sistemlerinin matematik modelleri, bilgisayar ortamındaki implementasyonları, aynı zamanda DQN sisteminin durum ve model bilgileri, tüm sistemlerin parametreleri ve lüzumlu diğer bilgiler bu bölümde verilmektedir.

3.1 Ters Sarkaç Matematik Modeli

Sarkaç sistemi, Newton'un hareket yasalarına dayanan bir matematik modeli ile tanımlanır. Sarkaç, bir kütlenin (m) bir çubuğun (l) ucuna bağlı olduğu ve yatay düzlemde hareket ettiği bir sistemdir. Bu sistemin dinamiği, sarkacın açısal konumunu ve hızını, arabaya uygulanan kuvvetin etkisiyle birlikte dikkate alarak belirlenir. (Tilbury et al., 2021) Şekil 3.1, bir ters sarkaç modellenirken kullanılan tüm fiziksel değişkenlerin ifadesini içermektedir.



Şekil 3.1 : Ters sarkaç sisteminin matematiksel modeli.

Arabaya ait serbest cisim diyagramındaki yatay yöndeki kuvvetlerin toplamı alınarak aşağıdaki hareket denklemini elde edilir:

$$M\ddot{x} + b\dot{x} + N = F \tag{3.1}$$

Dikey yönde de kuvvetlerin toplamı alınabilir, ancak bu işlemden faydalı bir bilgi edinilemez. Sarkacın serbest cisim diyagramındaki yatay yöndeki kuvvetlerin toplamını alarak reaksiyon kuvveti (N) için şu ifade görülür:

$$N = m\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta \tag{3.2}$$

Bu denklemi birinci denkleme yerleştirildiğinde, sistemin ilk denklemi elde edilir:

$$(M+m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta \tag{3.3}$$

Bu sistemin ikinci hareket denklemini elde etmek için, sarkaca dik kuvvetlerin toplamı alınır. Bu eksende çözüm yapmak matematiği büyük ölçüde basitleştirir. Buna göre ilerlendiğinde şu denkleme ulaşılır:

$$P\sin\theta + N\cos\theta - mg\sin\theta = ml\ddot{\theta} + m\ddot{x}\cos\theta \tag{3.4}$$

Yukarıdaki denklemin içindeki (P) ve (N) terimlerinden kurtulmak için, sarkacın ağırlık merkezine göre momentlerin toplamını alındığında şu ifade elde edilir:

$$-Pl\sin\theta - Nl\cos\theta = I\ddot{\theta} \tag{3.5}$$

Bu iki ifade birlikte ele alındığında, sistemin ikinci ana denklemi şekillenecektir:

$$(I+ml^2)\ddot{\theta} + mgl\sin\theta = -ml\ddot{x}\cos\theta \tag{3.6}$$

Bu hesaplar sırasında ulaştığımız denklemleri ve programlama dillerindeki diferansiyel denklem çözücüleri kullanarak ters sarkaç modeli bilgisayar ortamına aktarılabilmektedir. Kullanılan modelde 0 derece sarkacın aşağı denge noktasını işaret ettiği için pekiştirmeli öğrenme sırasında açı hedefini 0 derece yerine π dereceye göre olacak şekilde hizalama yapılmıştır. Bu sayede ödül fonksiyonunu minimize edebilmeye imkan bulunabilmiştir. Model üzerinde π dereceye ulaşıldığında, bu nokta yazılım tarafından 0 derece olarak tanınacaktır.

3.1.1 Ters sarkaç sistem parametreleri

Sistemimizdeki temel parametreler Çizelge 3.1 içerisinde aktarılmıştır. Bu parametreler, sarkacın hareketini etkileyen önemli unsurlardır ve sistemin performansını doğru simüle etmek için gereklidir.

Çizelge 3.1 : Ters sarkaç sistem parametreleri.

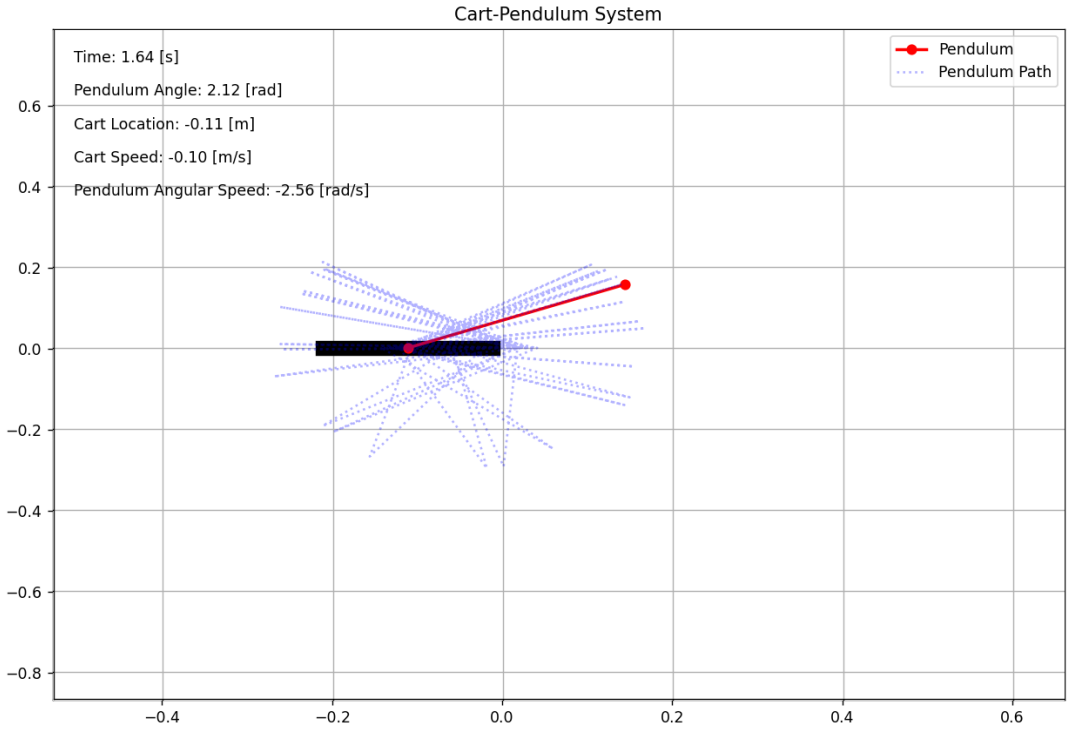
Temsil	Anlamı	Değeri
M	Arabanın kütlesi	0.5 [kg]
m	Sarkaç kütlesi	0.2 [kg]
b	Sürtünme katsayısı	0.1 [N/m/s]
l	Sarkaçın ağırlık merkezine uzaklığı	0.3 [m]
I	Sarkaçın atalet momenti	0.006 [kg·m²]
g	Yerçekimi ivmesi	9.81 [m/s²]
F	Araca uygulanan kuvvet	[N]
x	Aracın anlık pozisyonu	[m]
θ	Sarkacın dik eksenle yaptığı açı	[rad]
V	Aracın doğrusal hızı	[m/s]
w	Sarkacın açısal hızı	[rad/s]

3.1.2 Diferansiyel denklem fonksiyonu

Dinamik fonksiyonu, sarkaç sisteminin dinamiklerini tanımlayan bir diferansiyel denklem setini içerir. Girdi olarak sistemin mevcut durumunu (konum, hız, açı, açısal hız), zaman, fiziksel parametreleri ve dış kuvveti alır.

Bu fonksiyonda açının sinüs ve kosinüs değerleri hesaplanır. $\ddot{x}$ ve $\ddot{\theta}$, sırasıyla arabada ve sarkacın açısında hızlanmayı temsil eder. Sonuç olarak, sistemin bir sonraki durumu odeint fonksiyonu ile entegre edilerek hesaplanır.

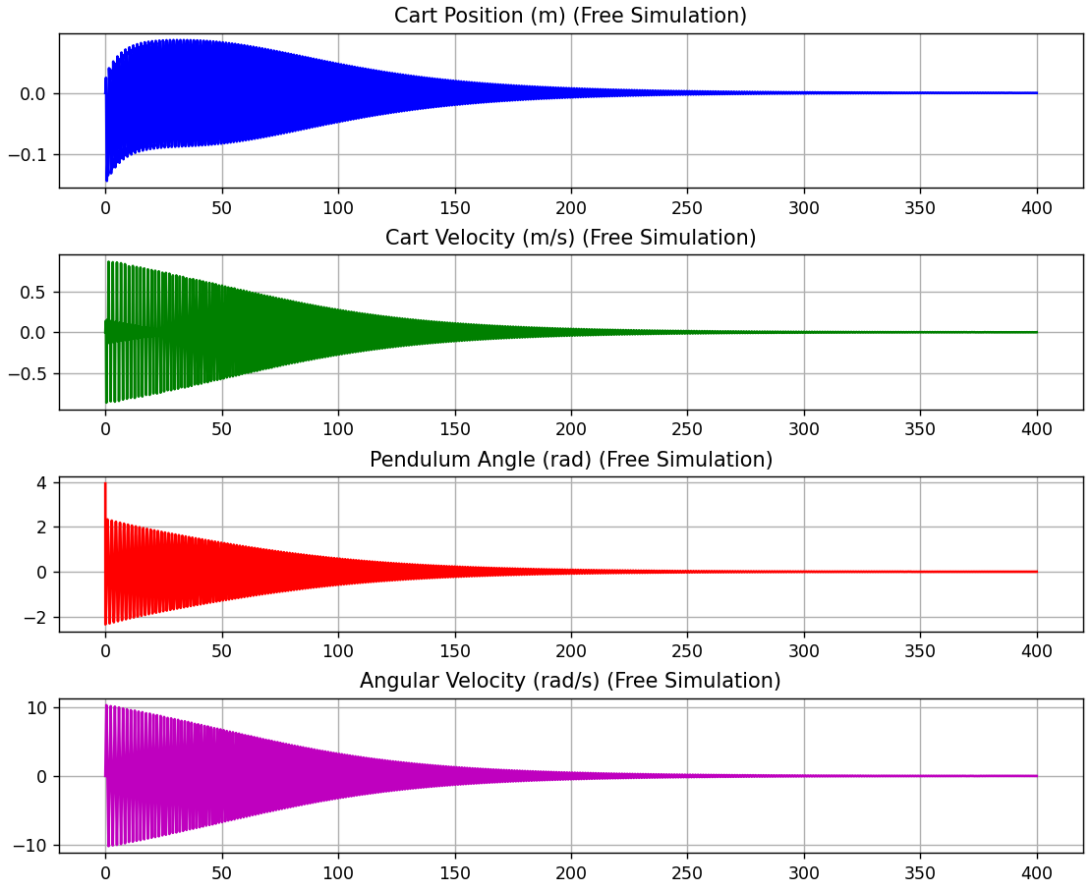
Sarkacın durumlarını arka arkaya ekleyerek hareketini daha rahat gözlemleyebilmek için bir simülasyon ortamı oluşturuldu. Bu simülasyon ortamı kullanılarak sarkacın kararsızlığa gittiği durumlar, eğitimin aşamaları ve karşılaşılan problemler gözlemsel olarak izlenebilir hale getirilmiştir. Aynı zamanda ödüllerin ve durumların grafikleri oluşturularak daha tutarlı bir ilerleme kontrolü yapılabilmektedir. Şekil 3.2'de, sistemin simülasyonu için tasarlanan kodun çıktısından bir örnek görülmektedir.



Şekil 3.2 : Python ortamında yazılan sarkaç simülasyonunun bir görüntüsü.

3.1.3 Sarkaç adım fonksiyonu

Adım fonksiyonu, verilen bir durum için sarkacın dinamiklerini güncellemek amacıyla bir zaman adımı boyunca sistemin yeni durumunu hesaplar. Bu adımda mevcut durum ve dış kuvvet ile birlikte zaman aralığı belirlenir, ardından dinamik durum fonksiyonu çağrılarak yeni durum hesaplanır. Şekil 3.3 üzerinde, kendisine hiçbir kuvvet uygulanmaksızın dik konumdan çok az bir sapma verilerek serbest bırakılmış bir ters sarkacın adım fonksiyonu sayesinde hesaplanan düşme sürecine ait bir grafik paylaşılmaktadır.

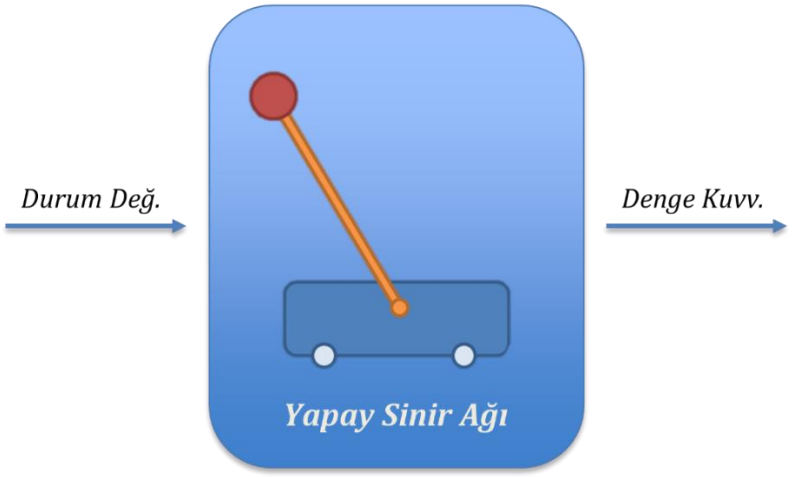


Şekil 3.3 : Python ortamında yazılan serbest sarkaç simülasyonunun sonucu.

3.2 Ters Sarkaç DQN Kurulumu, Modellemesi ve Adamın Eğitimi

Bu projede, derin Q-öğrenmesi (DQN) algoritmasını kullandım. Q-öğrenmesi, her durum-eylem çiftine bir ödül değeri (Q-değeri) atayarak çalışır. Adam, durumuna göre hangi eylemi seçeceğine karar vermek için bu Q-değerlerini kullanır. Derin öğrenme yöntemlerini kullanarak Q-değerlerini tahmin eden bir sinir ağı tasarlandı.

Sinir ağı, giriş olarak durum vektörünü alır ve çıkış olarak her eylem için Q-değerlerini döndürür. Ağın mimarisi üç tam bağlı sıralı katmandan oluşur ve her katmanda ReLU aktivasyon fonksiyonu kullanılmıştır. Şekil 3.4 üzerinde sisteme ait bir model gösterilmektedir.



Şekil 3.4 : Ters sarkaç kontrolü için tasarlanan DQN YSA'nın temsili modeli.

3.2.1 Ters sarkaç dqn sistem parametreleri

DQN parametreleri, sistemin etkili bir şekilde çalışmasını sağlamak için dikkatlice ve belirli bir süreç sonucunda seçilmiş ve Çizelge 3.2 üzerinde belirtilmiştir.

Çizelge 3.2 : DQN adamı sistem parametreleri.

Parametre	Değeri
Durum Uzayı	[x, x_dot, theta, theta_dot]
Aksiyon Uzayı	[-10, -7.5, -5, -2.5, 0, +2.5, +5, +7.5, +10]
Zaman Adımı	0.035 saniye (35 ms)
İndirim Oranı (γ)	0.99
Epsilon (ε) Başlangıç Değeri	1.0
Epsilon (ε) Minimum Değeri	0.01
Epsilon (ε) Azalma Oranı	0.9999
Öğrenme Hızı (α)	0.0001
Mini-batch Boyutu	64
Hafıza Boyutu	50,000
Giriş Katmanı Boyutu	4 (Durum uzayı boyutunda)
Ara Katman Boyutları	256, 256, 128
Ara Katman Aktivasyon	Relu
Çıkış Katmanı Boyutu	9 (Aksiyon uzayı boyutunda)
Çıkış Katmanı Aktivasyon	Lineer
Kayıp Fonksiyonu	MSE
Optimize Edici	Adam
Eğitim Bölüm Sayısı	2000
Bölüm Başına En Fazla Adım	200
Hedef Model Güncelleme Sıklığı	Her 2 adımda bir
Ödül (Ceza) Fonksiyonu	$-(0.1 \times s \times Q \times sT + 0.01 \times F2)$
Bölüm Bitirme Durumu	$\Theta \geq 0.785$ rad veya $x \geq 5$

Bölüm Bitirme Cezası	– 0.1 × (Kalan adım) ceza)	(En fazla -20

3.2.2 DQN adami yazilimsal yapisi

Modelin üzerine kurulduğu yazılımsal yapı, derin pekiştirmeli öğrenme için gerekli temel yapı taşlarını içerir.

Yapıcı Metod: Adamın durumu, eylemi, hafızası, öğrenme oranı ve epsilon değerleri gibi temel bileşenleri tanımlar.

Modelin İnşası: Model kurma metodu, Q-değerlerini tahmin etmek için bir yapay sinir ağı oluşturur. Ağı, yoğun (dense) katmanlar ile yapılandırarak, relu aktivasyon fonksiyonu kullanır.

Eylem Seçimi: aksiyon alma fonksiyonu, epsilon-greedy stratejisi ile keşif ve sömürü arasında bir denge sağlar. Epsilon değeri ile belirli bir olasılıkla rastgele bir eylem seçer, aksi takdirde tahmin edilen en yüksek Q-değerine göre hareket eder.

Hafızada Tutma: hafıza metodu, geçmiş deneyimleri hafızada saklar.

Yeniden Oynama: oyun tekrar metodu, rastgele seçilen deneyimlerden oluşan küçük bir örneklem uzayı ile modeli günceller. Burada, geçerli ödül ve gelecekteki en yüksek beklenen ödül ile hedef değer hesaplanır.

3.2.3 Durum uzayı

Sistemimizin durum uzayı dört değişkenden oluşmaktadır: $[x, \dot{x}, \theta, \dot{\theta}]$. Bu değişkenler arabanın yatay düzlemdeki pozisyonu, arabanın yatay düzlemdeki hızı, sarkacın dikey eksenden sapma açısı, sarkacın açısal hızı olarak ifade edilebilir.

Bu durum değişkenleri, adamımıza sarkacın o anki durumunu gösterir. Adam, bu durumu kullanarak uygun eylemi seçer ve sarkacı dik pozisyonda tutmaya çalışır.

3.2.4 Eylem uzayı

Adamın toplamda 9 olası eylem seçeneği vardır: arabaya sağa doğru [10, 7.5, 5, 2.5] kuvvetlerinden birisini uygulamak, arabaya sola doğru [-10, -7.5, -5, -2.5] kuvvetlerinden birisini uygulamak ya da arabaya kuvvet uygulamamak.

Bu eylemlerden birisi, her adımda adam tarafından seçilir ve sarkacın dengede tutulması için gerekli olan kuvvet uygulanır.

3.2.5 Hiperparametreler

Hiperparametreler, bir takviyeli öğrenme modelinin performansını ve öğrenme sürecini doğrudan etkileyen kritik ayarlardır. Seçimlerinizi aşağıdaki gibi açıklayabiliriz:

Zaman Adımı (0.035 saniye): Zaman adımı, simülasyonun her bir adımında çevrenin ne kadar ilerlediğini belirler. 35 milisaniyelik bir zaman adımı, sistemin dinamiklerini yeterince detaylı yakalamak için seçilmiştir. Daha kısa bir adım, hesaplama yükünü artırabilir ve öğrenmeyi yavaşlatabilir; daha uzun bir adım ise sarkacın hızlı hareketlerini veya önemli durum değişikliklerini kaçırabilir. Binlerce bölüm üzeründen yüzbinlerce adım süren bir eğitim yapıldığında, bu süre hem yeterince detaylı bilgi sağlar hem de hesaplama süresini makul bir seviyede tutar.

İndirim Oranı (γ=0.99): İndirim oranı, gelecekteki ödüllerin mevcut duruma ne kadar katkıda bulunduğunu belirler. γ=0.99 değeri, adamın uzun vadeli ödülleri dikkate almasını sağlar. Örneğin, dengeye ulaşmayı hedefleyen bir adam, yalnızca kısa vadeli dengeyi değil, uzun vadeli kararlılığı da önceliklendirir. Bu, sarkacın dengede tutulduğu süreyi artırmaya yönelik bir stratejiyi teşvik eder.

Epsilon (ε) Başlangıç ve Azalma Değerleri: İlk önce adam, çevresini keşfetmek için rastgele hareketler yapar. Bu Başlangıç Değeri (1.0), adamın başlangıçta farklı durumları denemesini sağlar ve keşfi destekler. Eğitim ilerledikçe adam, daha önce öğrendiği bilgilere dayanarak karar verir. Minimum değer (0.01), tamamen rastgele hareketlerden kaçınırken bir miktar keşfi sürdürmeyi sağlar. 2000 bölüm boyunca epsilonun yavaşça bir oranda (0.9999) azalması, adamın keşiften sömürüye doğru dengeli bir geçiş yapmasını sağlar. Çok hızlı bir azalma, erken aşamalarda yeterince keşif yapılmamasına; çok yavaş bir azalma ise optimal politikanın geç öğrenilmesine yol açabilir.

Öğrenme Hızı (α=0.0001): Öğrenme hızı, ağırlık güncellemelerinin boyutunu kontrol eder. Küçük bir değer (α=0.0001), güncellemelerin kararlı olmasını sağlar ve öğrenmenin aşırı osilasyonlara yol açmasını engeller. Düşük öğrenme hızı, özellikle karmaşık durum uzaylarında daha hassas öğrenmeyi destekler. Sarkacın hassas kontrol gerektirdiği durumlarda bu önemlidir.

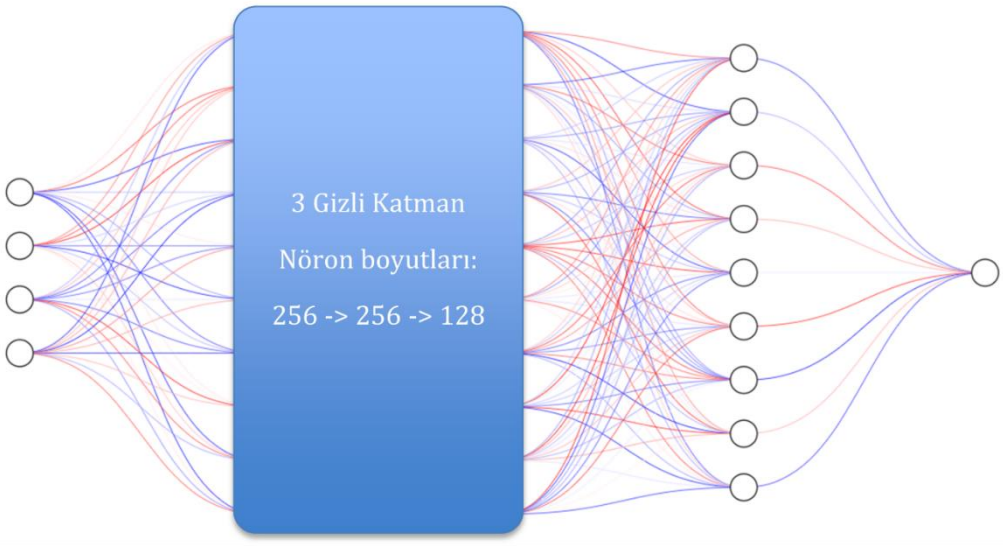
Örneklem uzayı Boyutu (64): Mini-batch boyutu, modelin güncelleme yaptığı örnek sayısını belirler. 64 boyutu, hesaplama verimliliği ile model kararlılığı arasında iyi bir denge sağlar. Çok küçük batch boyutları gürültülü güncellemeler yaparken, çok büyük batch boyutları öğrenmeyi yavaşlatabilir.

Hafıza Boyutu (50,000): Deneyim belleği, modelin geçmişteki deneyimlerini depoladığı alanı ifade eder. 50,000 boyutu, adamın farklı durumlar ve eylemlerle ilgili yeterli çeşitliliği öğrenmesine imkan tanır. Bu, özellikle 2000 bölüm ve 200 adım gibi uzun bir eğitim sürecinde geçmiş deneyimlerin yeniden kullanımı açısından kritiktir.

Bu hiperparametreler, simülasyonun dinamiklerini anlamaya yönelik bir denge sağlar ve eğitim sürecinin hem verimli hem de etkili bir şekilde tamamlanmasına olanak tanır. Parametreler, denge kontrolü gibi karmaşık problemler için optimize edilmiş olup, modelin daha hızlı ve kararlı bir şekilde öğrenmesine yardımcı olur.

3.2.6 Modelin ağ yapısı

Modelin ağ yapısı hem durumu hem de eylemi etkili bir şekilde temsil edecek şekilde tasarlanmıştır. Bu yapı, DQN için optimize edilmiştir. Şekil 3.5, modelin katman yapısını anlaşılır bir şekilde ifade etmektedir.



Şekil 3.5 : Ters sarkaç kontrolü için tasarlanan DQN modelinin ağ yapısı.

Giriş Katmanı (Boyut: 4): Giriş katmanı, durum uzayını temsil eder. Bu durumda, sistemin dört durumu (örneğin, sarkacın pozisyonu, açısı, hızları) giriş olarak alınır. Bu bilgiler, modelin çevresini anlaması için gereklidir.

Ara Katmanlar (Boyutlar: 256, 256, 128): Modelin ara katmanları, karmaşık durum-eylem ilişkilerini öğrenmek için derin bir yapıya sahiptir. İlk iki katmanda 256 nöron ve üçüncü katmanda 128 nöron bulunmaktadır. Bu boyutlar, modelin yüksek kapasiteli bir öğrenme sürecine olanak tanıyarak, doğrusal olmayan ilişkileri keşfetmesini sağlar. ReLU aktivasyon fonksiyonu ise hızlı hesaplama ve gradyan sönümleme sorunlarını azaltma özellikleriyle derin öğrenme modellerinde yaygın olarak kullanılır. ReLU'nun doğrusal olmayan yapısı, modelin karmaşık veri ilişkilerini öğrenmesine yardımcı olur.

Çıkış Katmanı (Boyut: 9): Çıkış katmanı, eylem uzayını temsil eder. Bu durumda, adam toplamda 9 farklı aksiyon seçeneğine (örneğin, farklı kuvvet veya moment uygulamaları) sahiptir. Çıkış katmanı, her aksiyon için bir Q-değeri döndürür. Lineer aktivasyon fonksiyonu, Q-değerlerini kesintisiz bir şekilde temsil eder ve bu değerlerin herhangi bir sınırlama olmaksızın optimize edilmesine olanak tanır.

Kayıp Fonksiyonu (MSE): Hedef Q-değerleri ile tahmin edilen Q-değerleri arasındaki farkı minimize etmek için MSE kullanılır. Bu, modelin doğru eylemleri seçmesine yardımcı olur.

Optimize Edici (Adam): Adam optimize edicisi, öğrenme hızını uyarlayarak ve momentum kullanarak daha hızlı ve kararlı bir optimizasyon sağlar. Bu, DQN gibi karmaşık modellerde daha iyi performans sunar.

Bu yapıyla model, durum ve aksiyonlar arasındaki ilişkileri öğrenmek için yeterli kapasiteye sahiptir. Katman boyutlarının ve aktivasyon fonksiyonlarının seçimi, modelin hem hesaplama maliyetini kontrol altında tutar hem de gerekli öğrenme kapasitesini sağlar.

3.2.7 Eğitim süreci

Eğitim süreci, takviyeli öğrenme (reinforcement learning) algoritmalarının adamı eğitmek için çevreyle etkileşimini simüle ettiği bir süreçtir. Bu süreçte adam, belirli bir hedefe ulaşmaya çalışırken her adımda ödül veya ceza alır ve bu ödül/ceza, adamı hedefe yönlendiren stratejilerin öğrenilmesine katkı sağlar.

Eğitim Bölüm Sayısı ve Adım Sayısı: Adam, 2000 farklı bölümü (episode) tamamlayarak öğrenir. Her bölüm, adamı başlangıç durumundan hedefe yönlendiren bir denemedir. Adam, her bölümde kendi stratejisini geliştirir ve ödüller alarak bu stratejiyi daha etkili hale getirir. Her bölümde adam en fazla 200 adım atabilir. Bu limit, çevreyi keşfetmek için gerekli zamanı sınırlar ve adamı hızlı bir şekilde doğru stratejiler geliştirmeye zorlar. Bu, eğitim sürecinin verimliliğini artırır.

Hedef Model Güncelleme Sıklığı: Modelin hedef ağı güncellenme sıklığı her 2 adımda bir olacak şekilde belirlenmiştir. Bu, modelin daha stabil bir şekilde öğrenmesini sağlar çünkü her adımda yapılan güncellemeler arasında daha fazla veri birikmesine olanak tanır. Böylece model, eğitim sürecinde daha az değişkenlikle karşılaşır ve daha doğru bir strateji geliştirir.

Ödül (Ceza) Fonksiyonu: Bu fonksiyon, sarkacın durumu ve güç uygulamaları arasındaki ilişkiyi puanlar. Burada s sarkacın durumunu, Q ağırlık matrisini, s^T sarkacın durumunun transpoz vektörünü ve F ise güç kullanımını temsil eder. Bu fonksiyon adamı adaptif bir şekilde hedeften uzaklaşıldıkça ve gereksiz yüksek kuvvet uyguladıkça daha fazla cezalandırarak, mümkün olduğunca düşük ceza ile daha iyi bir kontrol stratejisi geliştirmeye teşvik eder.

Bölüm Bitirme Durumu: Adam, sarkacı dengeye getirene kadar devam eder. Eğer açısı belirli bir değeri (45 derece, yani 0.785 rad) aşarsa veya sarkacın pozisyonu hedefe ulaşarak 5 birim uzaklıkta olursa, bölüm sona erer. Bu hedefler, sarkacın başarılı bir şekilde dengeye ulaşmasını simüle eder.

Bölüm Bitirme Cezası: Bölüm erken sona ererse (yani adam sarkaç dengeye gelemeden devrilirse), kalan bölüm sayısına bağlı bir ceza uygulanır. Bu ceza, adamın zamanın verimli kullanılmasını sağlamak amacıyla, erken bitirilen bölümlere karşılık gelir. Bu ceza, adamın daha iyi stratejiler geliştirmesini teşvik eder.

Bu eğitim süreci, adamı her adımda daha iyi stratejiler geliştirmeye zorlar ve uzun vadede en verimli dengeleme stratejilerini öğrenmesini sağlar. Adam, her adımda

çevreyle etkileşime girerek durumunu değerlendirir, ödülleri ve cezaları göz önünde bulundurarak aksiyonlarını optimize eder.

3.2.8 Ters sarkaca uygulanacak rastgele darbelerin simüle edilmesi

Ters sarkaç simülasyonunda robotların muharebesine doğru ilerlerken, rastgele zamanlarda karşıdan gelebilecek darbeleri simüle etmek için **Poisson dağılımı** kullanıldı. Bu dağılım, belirli bir zaman diliminde gerçekleşen olayların sayısını modellemek için uygundur. Poisson dağılımı kullanarak, her bir simülasyon adımında beklenen darbe sayısını belirleriz. Bu darbelerin zamanları ve kuvvetleri rastgele seçilir.

Poisson dağılımı, belirli bir ortalama darbe sayısına göre darbelerin gerçekleşmesini simüle eder. Örneğin, 200 adımdan oluşan her bölümde ortalama $\lambda = 10$ darbe beklediğimizde, Poisson dağılımı bu sayının etrafında rastgele sayılar üreterek oluşacak darbe sayısını belirleyecektir. Yani, her simülasyonda darbe sayısı değişebilir, ancak ortalama değer sabit olur.

Çizelge 3.3 : Poisson dağılımı ile rastgele darbe üretimi.

Parametre	Değeri
Poisson Lambda (λ)	10
Darbe Sayısı	Ortalaması λ olacak şekilde seçiliyor
Darbe Adımları	Maksimum adım sayısından seçiliyor
Darbe Kuvvetleri	Aksiyon uzayından seçiliyor

Darbeler, simülasyonun belirli adımlarında meydana gelir. Bu adımlar, zaman çizelgesinden rastgele seçilir. Seçilen adımlar, darbenin ne zaman uygulanacağını belirler. Bu adımlar, tüm adımlar arasında eşit olasılıkla seçilir ve sıralanır. Her darbe için bir kuvvet değeri belirlenir. Bu kuvvetler genellikle belirli bir aralıktan rastgele seçilir. Kuvvetlerin pozitif veya negatif olması, darbenin yönünü belirler.

Simülasyon her adımda devam ederken, belirli adımlarda darbeler uygulanır. Eğer o adımda bir darbe varsa, kuvvet bu adımda sisteme eklenir. Bu yaklaşım, sarkaç gibi dinamik sistemlerde dış etkilerin rastgele olabileceği durumları simüle etmek için kullanılır. Poisson dağılımı, darbelerin sayısını modellerken, rastgele kuvvetler sistemin karmaşıklığını artırır ve gerçek dünyadaki düzensizlikleri taklit eder.

3.2.9 Sonuçların analizi

Eğitim sürecinde elde edilen sonuçlar, adamı değerlendirmek ve eğitimin etkinliğini görselleştirmek amacıyla kaydedilir. Bu kayıtlar, adamın öğrenme sürecini izlemek ve eğitim stratejilerinin doğruluğunu analiz etmek için kullanılır. Eğitim tamamlandıktan sonra, adam ve eğitimle ilgili veriler belirli dosyalar olarak kaydedilir. Örneğin, adam nesnesi kayıt fonksiyonu ile kaydedilirken, animasyon çizdirmek için gerçek durumlar, faz portrelerinin ve zaman göre durum grafiklerin hesabı için ödül durumları ve ödüller numpy dosyaları olarak saklanır. Bu veriler, daha sonra analiz ve görselleştirme amacıyla kullanılmak üzere saklanır.

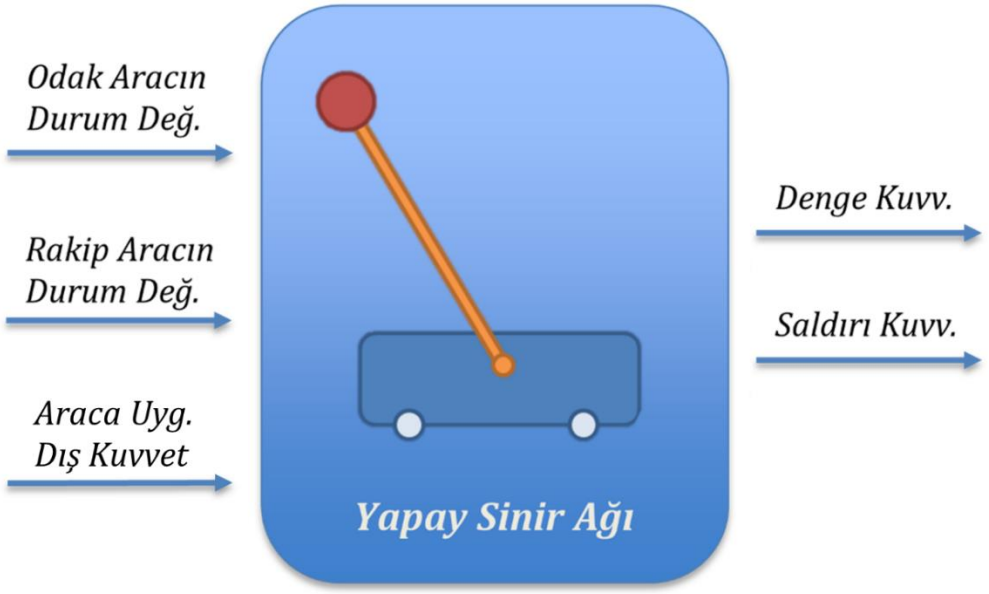
Kaydedilen verilerin görselleştirilmesi için bir menü sunulur. Bu menüde, kullanıcı farklı analiz ve simülasyon türlerini seçebilir. Örneğin, **kuvvetsiz simülasyon animasyonu** ve **grafikleri** ile adamların çevreyle etkileşimi görsel olarak izlenebilir. Eğitim sürecinin nasıl ilerlediğini gözlemlemek için **eğitim süreci animasyonu** seçeneği kullanılabilir. Bu animasyon, adamı gerçek zamanlı olarak gözlemlemenizi sağlar ve adamın hareketlerini, ödüllerini ve strateji değişimlerini izlemek için yararlıdır. Ayrıca, **eğitim süreci durum grafikleri** ile **ödül ve faz portresi grafikleri** gibi görselleştirmeler, adamın her bölümdeki durumunu ve aldığı ödülleri zaman içinde görmenize olanak tanır. Bu grafikler, adamın öğrenme sürecinin nasıl evrildiğini, ödüllerin nasıl değiştiğini ve stratejilerinin ne kadar verimli olduğunu anlamanızı sağlar.

Son olarak kaydedilen adam durumunu bir test modunda sıfır epsilon değeriyle çalıştırmaya devam etmek, kullanıcıya modelin eğitim sonrası performansını gerçek zamanlı olarak test etme fırsatı sunar. Bu simülasyon, adamın eğitildikten sonra ne kadar başarılı olduğunu, belirli hedeflere ulaşmada ne kadar verimli olduğunu gösterir. Bu görselleştirmeler sayesinde, eğitim süreci boyunca elde edilen sonuçların analiz edilmesi ve modelin doğruluğu hakkında bilgi edinilmesi sağlanır.

3.3 Çift Sarkaç DQN Kurulumu, Modellemesi ve Adamın Eğitimi

Tek başına bir sarkacın dengede durması sağlandıktan sonra, iki sarkacın bağımsız bir şekilde ayrı ayrı dengelerini korumaları ve aynı zamanda birbirleriyle etkileşime girerek karşılıklı olarak birbirlerine devirme amacıyla kuvvet uygulamaları hedefiyle bir çift sarkaç simülasyonu tasarlanmıştır.

Çift sarkaç simülasyonunda, sistem parametrelerinin büyük bir kısmı aynı kalmakla birlikte, durum uzayı ve çıktı uzayı iki katına çıkmıştır. Bunun nedeni, her iki sarkacın da durumuyla aracın kendisine uygulanan saldırı kuvvetini değerlendirmek ve her birinin dengede kalma kuvvetinin yanı sıra karşı tarafa uygulanacak saldırı kuvvetini seçmek zorunluluğudur. Bu bölümde, yalnızca önceki bölümde yapılan geliştirmeler ve uygulamaya konan yeni yöntemler ele alınmaktadır.



Şekil 3.6 : Çift ters sarkaç kontrolü için tasarlanan DQN YSA'nın temsili modeli.

3.3.1 Çift sarkaç DQN sistem parametreleri

DQN sistemimizde kullanılan temel parametreler çizelge 3.4'te gösterilmiştir. Bu parametreler, DQN sisteminin etkili bir şekilde çalışmasını sağlamak için dikkatlice ve belirli bir süreç ilerlemenin sonucunda seçilmiştir. Adamın eğitimi esnasında kullanılan hiperparametrelerde herhangi bir değişiklik yapılmamıştır.

Özetlemek gerekirse, yeni model iki aracı ve saldırı değerlerini kontrol etmek üzere tasarlandığından ağ yapısı biraz daha karmaşıklaştırıldı ve katmanlar genişletildi, ayrıca keşif sürecinin daha uzun sürmesi için epsilon azalma oranı yavaşlatıldı. Aynı zamanda ödül fonksiyonuna saldırmak için kullanılan değerle ilgili bir terim eklendi.

Çizelge 3.4 : Çift sarkaç sistemi için DQN adamı sistem parametreleri.

Parametre	Değeri
Durum Uzayı	[Odak uzayı, Rakip uzayı, Gelen darbeler toplamı]
Standart Durum Uzayı	[x, x_dot, theta, theta_dot]
Aksiyon Uzayı	[Denge Uzayı, Saldırı Uzayı]
Denge Uzayı	[-10, -7.5, -5, -2.5, 0, +2.5, +5, +7.5, +10]
Saldırı Uzayı	[-1.5, -1.13, -0.75, -0.38, 0, +0.38, +0.75, +1.13, +1.5]
Epsilon (ε) Azalma Oranı	0.99995
Öğrenme Hızı (α)	0.0001
Mini-batch Boyutu	128
Hafıza Boyutu	100,000
Ara Katman Boyutları	512, 512, 256
Giriş Katmanı Boyutu	9
Çıkış Katmanı Boyutu	18
Model Güncelleme Sıklığı	Her adımda bir
Ödül Fonksiyonu	Aracın kendi ödülü – 0.3 × Karşı aracın ödülü
Standart Ödül Fonksiyonu	$-(0.1 \times s \times Q \times sT + 0.01 \times F_{denge}^2 + 0.005 \times F_{saldırı}^2)$

3.3.2 DQN adamı yazılımsal yapısı

Eğitim noktasında yeni sistem önceki durumdan devam edebilme, ikili ortam, dövüş ortamı ve poisson ile rastgele bozucu darbe uygulanması durumlarını açıp kapatabilme imkanıyla tasarlanmıştır. Kavga modu açıkken ve kapalıyken karşı tarafın durumunun ana araç durumu üzerindeki etkisi hesaplamalara dahil edilmekte veya edilmemektedir. Yeni eğitim sistemiyle öncelikle araçların kendi başlarına dengelerini sağlaması, sonrasında kavgayı öğrenmesi amaçlanmaktadır. Bu şekilde dengede durmayı başaramadan adamın sadece karşı tarafın düşmesinden faydalanmak suretiyle kolay yoldan bir zafer elde etmesi engellenmeye çalışılmıştır.

Adam sistemi özelinde tek değişiklik kurucu fonksiyon ve eylem seçimi fonksiyonunda yapılmıştır. Kurucu fonksiyon yeni giriş ve çıkış uzayına ve ağ yapısına göre düzenlenmiştir. Eylem fonksiyonunda ise giriş ve çıkış uzayı ana araç ve karşı araç için olmak üzere ortadan ikiye bölünmüştür. Kavga modu kapalıysa eylem fonksiyonuna karşı tarafın durumu hakkında bilgi gitmemekte ve herhangi bir saldırı kuvveti kararı verilmemektedir. Kavga modunun açılması önceden yapılan denge eğitiminin ardından gerçekleştiği için, başlangıçta saldırı kuvvetini rastgele verse de denge kuvvetini hafızadan seçmeye devam edecek şekilde tasarlanmıştır.

Fonksiyon, yine amacına uygun olarak epsilon-greedy stratejisi ile keşif ve istismar arasında bir denge sağlar. Epsilon değeri ile belirli bir olasılıkla rastgele bir eylem seçer, aksi takdirde modelden tahmin edilen en yüksek Q-değerine göre hareket eder.

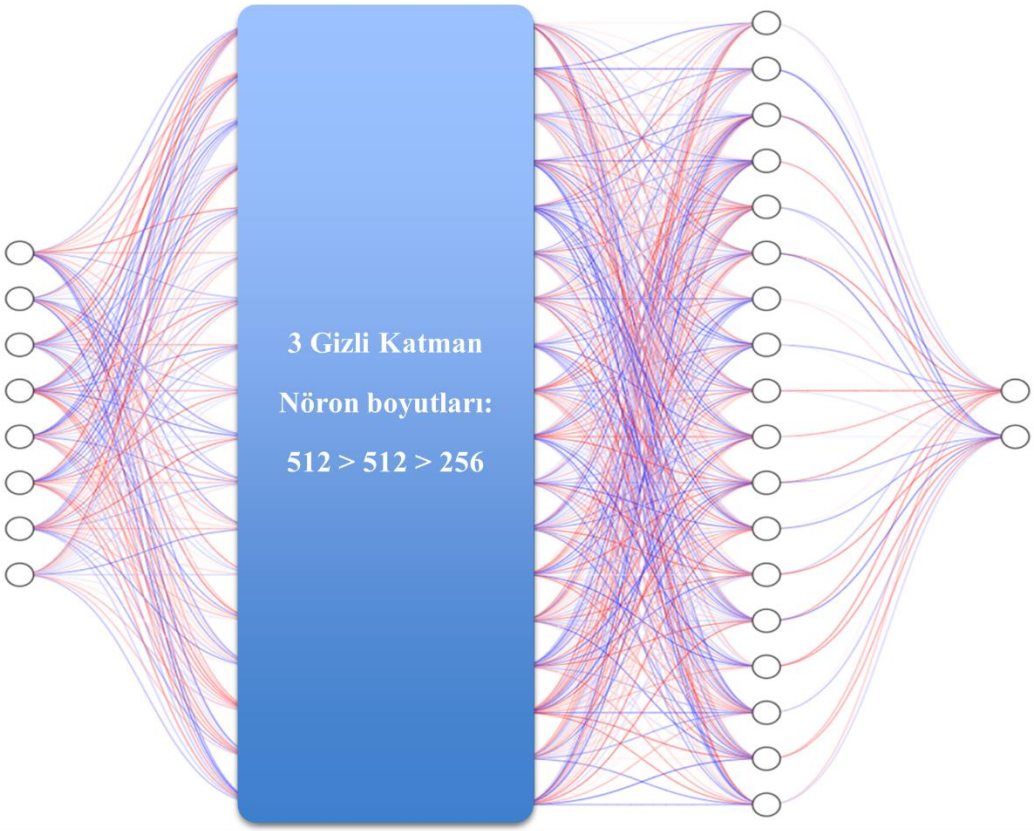
3.3.3 Durum ve eylem uzayları

Sistemimizin durum uzayı $[x_1, \dot{x}_1, \theta_1, \dot{\theta}_1, x_2, \dot{x}_2, \theta_2, \dot{\theta}_2, F]$ olmak üzere toplamda 9 değişkenden oluşmaktadır. Bu değişkenler adamımıza sarkacın kendisinin ve rakibinin o anki durumunu ve araca etkiyen dış kuvvetlerin toplamını gösterir. Dış kuvvetler, karşıdan gelen saldırı değerini ve rastgele bir bozucu kuvveti içerebilir, toplam bir değerdir. Adam bu durumu kullanarak uygun eylemi seçer ve sarkacı dik pozisyonda tutmaya, kavga modu açıksa da karşısındaki rakibi devirmeye çalışır.

Eylem uzayı tarafında ise adamın toplamda 18 olası eylem seçeneğinden 2 seçeneği seçmesi gerekir. Uzay, standart eylem uzayının iki tanesinin yanyana koyulmasından oluşur. Bunlar uzayın ilk yarısından seçilen savunma ve ikinci yarısından seçilen saldırı kuvvetleridir. Bu eylemler, her adımda adam tarafından seçilir ve sarkacın dengede tutulması ve rakibin devrilmesi için gerekli olan kuvvet uygulanır.

3.3.4 Modelin ağ yapısı

Modelin ağ yapısı hem durumu hem de eylemi etkili bir şekilde temsil edecek ve yeni sistemin karmaşıklığına kolay adapte olacak şekilde tasarlanmıştır.



Şekil 3.7 : Çift ters sarkaç kontrolü için tasarlanan DQN modelinin ağ yapısı.

Giriş Katmanı (Boyut: 9): Giriş katmanı, durum uzayını temsil eder. Bu durumda, iki ayrı sarkaç sistemin dört durumu (örneğin, sarkacın pozisyonu, açısı, hızları) ve araca dışarıdan etkiyen kuvvetlerin toplamı olmak üzere toplam 9 boyutlu bir giriş olarak alınır.

Çıkış Katmanı (Boyut: 18): Çıkış katmanı, eylem uzayını temsil eder. Bu durumda, adam saldırı ve denge olmak üzere iki farklı aksiyon uzayının birleşimi olarak toplamda 18 farklı aksiyon seçeneğine sahiptir. Çıkış katmanı, her aksiyon için bir Q-değeri döndürür.

Bu yapıyla model, durum ve aksiyonlar arasındaki ilişkileri öğrenmek için yeterli kapasiteye sahiptir. Katman boyutlarının ve aktivasyon fonksiyonlarının seçimi, modelin hem hesaplama maliyetini kontrol altında tutar hem de gerekli öğrenme kapasitesini sağlar.

3.3.5 Eğitim süreci

Eğitim süreci, takviyeli öğrenme algoritmalarının adamı eğitmek için çevreyle etkileşimini simüle ettiği bir süreçtir. Bu süreçte adam, belirli bir hedefe ulaşmaya çalışırken her adımda ödül veya ceza alır ve bu ödül/ceza, adamı hedefe yönlendiren stratejilerin öğrenilmesine katkı sağlar.

Eğitim Bölüm Sayısı ve Adım Sayısı: Adam, sonsuz döngü olarak kullanıcı durdurana kadar her seferinde eski tecrübelerinden devam ederek rastgele oluşturulmuş bölümleri tamamlayarak öğrenir. Her bölüm, adamı başlangıç durumundan hedefe yönlendiren bir denemedir. Modelin hedef ağı güncellenme sıklığı adımda bir olacak şekilde belirlenmiştir.

Ödül (Ceza) Fonksiyonu: Tekil ödül fonksiyonuna karşı tarafa uygulanılan kuvvet bir çarpanla eklenmiş olup, kavga modunun açık olup olmadığına göre adam karşı tarafın devrilmesinden de pozitif puan kazanmaktadır. Bu yöntem adamı adaptif bir şekilde hedeften uzaklaşıldıkça ve gereksiz yüksek kuvvet uyguladıkça daha fazla cezalandırarak, karşı taraftaki sarkaç devrildikçe ödüllendirecek şekilde mümkün olduğunca yüksek ödül ile daha iyi bir strateji geliştirmeye teşvik eder.

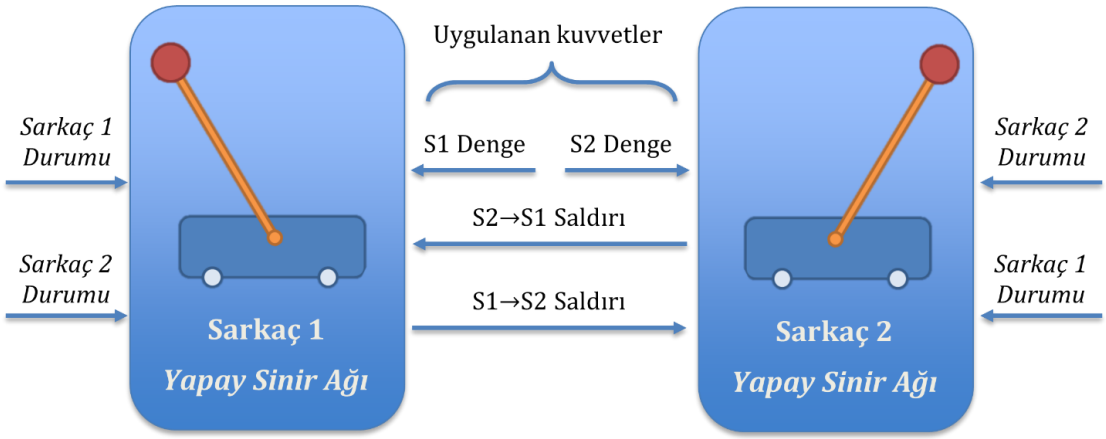
Kavga modu kapalı olduğu halde birleşik ödül uygulanırsa adam zaten karşı tarafın düşüşünden ödül aldığı için ayrıca dengede durmaya devam etmek için bir kuvvet uygulamaktan vazgeçer. Bundan dolayı kavga modu kapalı ve açık durumda ödül hesabı farklıdır. Bu eğitim süreci, adamı her adımda daha iyi stratejiler geliştirmeye zorlar ve uzun vadede en verimli dengeleme stratejilerini öğrenmesini sağlar. Adam, her adımda çevreyle etkileşime girerek durumunu değerlendirir, ödülleri ve cezaları göz önünde bulundurarak aksiyonlarını optimize eder.

3.3.6 Sonuçların analizi

Eğitim sürecinde elde edilen sonuçlar, adamı değerlendirmek ve eğitimin etkinliğini görselleştirmek amacıyla kaydedilir. Bu kayıtlar, adamın öğrenme sürecini izlemek ve eğitim stratejilerinin doğruluğunu analiz etmek için kullanılır. Eğitim tamamlandıktan sonra, adam ve eğitimle ilgili veriler, iki sarkacın ayrı ayrı istatistikleri belirli dosyalar olarak kaydedilir. Kaydedilen verilerin görselleştirilmesi için bir menü sunulur. Bu menüde, kullanıcı farklı analiz ve simülasyon türlerini seçebilir.

3.3.7 Çift sarkacın birbirleriyle etkileşimi

İki ters sarkaç sisteminin birbirleriyle etkileşimi esnasında sistemde her bir sarkaç, kendi durumunu ve diğer sarkacın durumunu giriş olarak alan aynı yapay sinir ağı tarafından kontrol edilmektedir. Bu ağ, her sarkaç için denge kuvveti ve saldırı kuvveti olmak üzere iki farklı kontrol çıkışı üretmektedir. İki farklı yapay sinir ağının kullanıldığı sistemlerde sistemlerden birisinin diğerine kısa süre içerisinde ezici bir üstünlük geliştirdiğinden ötürü bütün sistem tek bir yapay sinir ağı tarafından kontrol edilmek üzere tasarlanmıştır.



Şekil 3.8 : Çift ters sarkacın kavga modunda birbiriyle etkileşimleri.

Sistemin temel çalışma prensibi, iki farklı mod üzerine kurgulanmıştır. Kavga modunun devre dışı olduğu durumda, her sarkaç yalnızca kendi dengesini korumaya odaklanır. Bu modda saldırı kuvvetleri sıfır olarak ayarlanmakta ve sarkacın durumu yalnızca kendi denge kuvveti tarafından belirlenmektedir. Kavga modunun aktif olduğu durumda ise, her sarkaç hem kendi dengesini korumaya çalışmakta hem de diğer sarkaca etki edebilmektedir. Bu modda yapay sinir ağı, her sarkaç için hem denge kuvveti hem de saldırı kuvveti üretmektedir.

Bu çift modlu kontrol sistemi, tek bir yapay sinir ağının çoklu görev öğrenme kapasitesini test etmektedir. Ağın hem tekil sarkaç dengeleme görevini başarıyla gerçekleştirmesi hem de uygun koşullarda rakip sarkacı etkisiz hale getirme stratejileri geliştirmesi beklenmektedir.

4. BULGULAR

4.1 Temel Hiperparametrelerin Değişimi ve Sonuçlar

Öğrenme süreci boyunca, farklı hiperparametreler üzerinde çeşitli denemeler yaparak optimal sonuçlar elde etmeye ve sistemin performansını iyileştirmeye çalıştım. Aşağıda, farklı parametre değerlerinin nasıl çalıştığını ve sonuçlarını detaylıca ele alıyorum.

4.1.1 Öğrenme oranı (α)

Başlangıçta öğrenme oranını 0.001 olarak ayarlanmıştı. Eğitimler çok uzun sürdüğü için değer 0.1 olarak değiştirildi ancak bu sefer de ağın hızlı bir şekilde öğrenmek yerine aşırı sıçramalar yaptığı gözlendi. Bu durum, adamı istikrarsız hale getirdi ve ödülde büyük dalgalanmalar oluştu. Bu, seçilen öğrenme oranının çok büyük olduğunu gösteriyordu. Daha sonra öğrenme oranı 0.01 olarak ayarlandı ve bölümler boyunca belirli aralıklarla azaltılması sağlandı. Bu şekilde daha tutarlı sonuçlar elde edildi. Sürecin en sonunda eğitim sürecinde zaten çok fazla adım olduğundan ve hızlıca bir baz değere ulaşıldığından dolayı kademeli azaltma stratejisinden vazgeçtim ve $\alpha = 0.0001$'de karar kıldım.

- **Başlangıç değeri**: 0.001
- **Denenen değerler**: 0.1**,** 0.01 ile başlayarak kademeli azalan bir değer
- **Son değer**: 0.0001

Bu değişiklik sonrasında adam, daha istikrarlı bir şekilde öğrenmeye başladı ve ödül fonksiyonu daha düzenli hale geldi. Buradan hareketle dinamik ve hızlı hareket eden sarkaç gibi sistemlerde düşük öğrenme adımlarıyla ilerlemenin eğitimi daha kararlı hale getirdiği sonucuna vardım.

4.1.2 İskonto faktörü (γ):

Gamma, adamımızın gelecekteki ödülleri ne kadar dikkate aldığını belirleyen bir parametreydi. Başlangıçta $\gamma = 0.99$ olarak ayarlamak suretiyle daha sonraki hedeflerin de önemli olduğu belirlenmiş oldu. Daha iyi bir sonuca ulaşmak adına γ değerini düşürerek sistem tekrar gözlendi ancak bu durumda alınan sonuç eskisi kadar tatmin edici olmadığından dolayı başlangıç değerine geri dönüldü.

- **Başlangıç değeri**: 0.99
- **Denenen değerler**: 0.90, 0.95
- **Son değer:** 0.99

4.1.3 Keşif Oranı (ε) azalması:

Başlangıçta adam, $\boldsymbol{\varepsilon} = 1.00$ olacak şekilde yüksek bir keşif oranına sahipti, bu da adamın her durumu rastgele denemesine neden oldu. Öğrenme ilerledikçe adam daha iyi bir politika oluşturmaya başladığında, keşif oranı kademeli olarak azaltıldı ve $\boldsymbol{\varepsilon} = 0.01$ minimum seviyesine kadar düştü. Bu süreç, adamımızın daha az rastgele hareket edip, daha fazla öğrendiği politika doğrultusunda hareket etmesini sağladı. Azalma oranının buradaki rolü ise uzun eğitim süreci boyunca keşif oranının dengeli bir şekilde azaltılmasıydı. Eğitim süreci boyunca farklı azalma oranları denendi ancak bazı değerlerde keşif oranı çok hızlı bir şekilde düşerek adam daha ortamı tanıyamadan kendi bilgilerini kullanmaya başlamak zorunda kaldı ve bu durum eğitimi kesintiye uğrattı. Bundan dolayı en yavaş düşüş için 0.9999 nihai azalma oranı değeri kullanılmış oldu.

- **Başlangıç azalma değeri**: 0.90
- **Denenen değerler:** 0.70, 0.99, 0.999
- **Son değer**: 0.9999
- **Çift sarkaç sistemi için son değer:** 0.99995

Epsilon azalma oranındaki bu yükselme, adamımızın daha uzun sürede daha iyi bir politika öğrenmesine olanak tanıdı. İlk bölümlerde rastgele keşif yapılırken, ilerleyen bölümlerde daha belirgin ve kararlı bir politika izlemeye başladı.

4.1.4 Toplu işlem boyutu (batch size):

Toplu işlem boyutu, sinir ağının eğitiminde kullanılan verilerin miktarını belirleyen önemli bir hiperparametreydi. Batch size, veya toplu işlem boyutu, sinir ağlarının eğitiminde kullanılan bir hiperparametredir ve eğitim verilerinden kaç örneğin her bir adımda işlem göreceğini belirler. Tam eğitim setini tek seferde işlemek yerine, eğitim daha küçük ve yönetilebilir parçalar halinde yapılır ve bu parçalara batch (toplu işlem) denir.

Deneyimler hafızada saklandıktan sonra, bu deneyimlerden rastgele örnekler alarak sinir ağını eğitmek için 64 değeri kullanıldı. Bu boyut, verilerin birbirine bağımlılığını azaltarak daha genel bir öğrenme sağladı. Daha küçük boyutlar denendiğinde (örneğin, 32), adam hızlı bir şekilde öğrenmekte zorlandı.

- **Başlangıç değeri**: 64
- **Denenen değerler**: 32
- **Son değeri**: 64
- **Çift sarkaç sistemi için son değer:** 128

4.2 Ödül Fonksiyonu Gelişimi ve Sonuçlar

Bu proje boyunca ödül fonksiyonunda çeşitli denemeler yapılarak sistemin performansı iyileştirildi. Aşağıda, farklı ödül fonksiyonlarının nasıl çalıştığını ve sonuçlarını detaylı olarak ele alınmaktadır.

4.2.1 Basit kosinüs temelli ödül fonksiyonu

Ödül fonksiyonu, sarkacın açısının kosinüsüne dayandırıldı. Amaç, sarkacın dik pozisyonda olduğu durumlarda yüksek ödül vermekti. Ödül fonksiyonu 4.1 numaralı denklemdeki gibiydi.

$$reward = \cos\theta \tag{4.1}$$

- Adam, sarkacı kısa süreler boyunca dengede tutmayı başardı, ancak uzun vadede tutarlı sonuçlar elde edemedi.
- Bu yaklaşım, adamı sadece açısal sapmaya odaklandırdığı için, arabayı sabit bir pozisyonda tutmakta zorluk çekti.

Kosinüs temelli ödül fonksiyonu, matris tabanlı karesel ceza fonksiyonundan önce elde ettiğim en başarılı fonksiyondu. Sistem, başarı koşulunu yüksek bir oranda sağlamayı başardı ancak eğitimi uzun vadede negatif etkilediğinden, matris çarpımlı ödül fonksiyonuyla birlikte sarkaç devrilince bölümün bittiği bir yönteme geçildi.

4.2.2 Açı ve açısal hız tabanlı ödül fonksiyonu

Sarkacın açısal sapması ve açısal hızına dayalı bir ceza fonksiyonu eklendi, değişkenlerin katsayıları zamanla değiştirelerek farklı katsayılarla önemleri değiştirildi ve ayrıca eğitimler yapıldı. Aracın konumu ve hızı dikkate alınmadığından uzun vadede başarılı sonuçlar elde edilemedi.

$$reward = -W * \theta^2 - B * |\dot{\theta}| \tag{4.2}$$

- Öğrenme süreci yavaşladı ve adam dengeyi sağlamada başarılı olamadı.
- Konum ve hız hesaba dahil edilmediği için adam bazı adımlarda dik durmayı başarsa da araç sürekli kayarak sonsuza gittiği için dik durma durumu korunamadı.

4.2.3 Tüm durumları dikkate alan ağırlıklandırılmış ödül fonksiyonu

Son denemede, durum vektörü ve ağırlık matrisi Q kullanılarak daha karmaşık bir ödül fonksiyonu oluşturuldu. Bu fonksiyon, durum vektöründeki her bir bileşene farklı ağırlıklar atayarak cezalandırma yaptı. Özellikle sarkacın doğrusal ve açısal konumu

daha büyük ağırlıklarla cezalandırıldı. Gerçek bir sisteme uygulanma durumunda güç tasarrufu sağlanması için düşük bir ağırlıkla uygulanan güç de cezalandırıldı. Denklem 4.3'te gösterilen ödül fonksiyonu incelendiğinde buradaki Q matrisi, dört durum değişkenine farklı ağırlıklar atayan bir köşegen vektör matrisidir.

$$reward = -W_1 * x * Q * x^T - W_2 * F^2 \tag{4.3}$$

- Bu ödül fonksiyonu, adamı sadece birkaç değişkeni azaltmaya değil, aynı zamanda tüm durumu kontrol etmeye yönlendirdi.
- Sarkacın dik pozisyonda daha uzun süre tutulabilmesi sağlandı ve öğrenme hızı arttı.
- Adamın performansı önemli ölçüde iyileşti, özellikle uzun vadeli dengeleme görevinde başarılı sonuçlar elde edildi.

Kuadratik olarak tanımlanan ve bir Q matrisi aracılığıyla her değerin önemini belirten, birbiriyle eşit oranda sonucu etkilemesi için bir faktör ile büyüterek sonucu belirleyen bir ceza fonksiyonuyla çalışan ve sarkaç devrildiğinde bölümü bitiren bir eğitim sistemi en başarılı eğitim sonuçlarına ulaştı. Eğitim sırasında ödüller çok yüksek oranda sıfıra yakın oldu ve özellikle son bölümlerde neredeyse tamamına yakını çok küçük değerlerden oluşmaktaydı.

4.2.4 Ödül ağırlık matrisi (q) gelişimi ve elde edilen sonuçlar

Ödül fonksiyonunda çarpım işlemine katılan Q matrisi, dört durum değişkenine farklı ağırlıklar atadı. Böylece her değişkeni tek tek çarpım olarak göstermek yerine değişkenlerin ağırlıkları daha kolay bir şekilde matrisle belirtilmiş olundu.

$$Q_1 = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \tag{4.4}$$

Arabanın yatay pozisyonu ve açısal hızı için ortalama ağırlıklar kullanıldı. Açı için ise çok büyük bir ağırlık verildi. Bu, sarkacın açısal sapmasını daha fazla cezalandırarak dik pozisyonda kalmasını sağlamayı hedefliyordu. Ancak uzun vadede oranlar arasındaki yüksek farklar ödül fonksiyonunda dengesiz davranışlara, aynı zamanda sarkacın ve aracın da sonsuza yakınsamasıyla sonuçlandı.

$$Q_2 = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \tag{4.5}$$

Asıl ağırlık açıya ve dengeli şekilde hızlara verildiğinde konum da dolaylı olarak stabilize edilmiş oldu. Bu yeni ağırlık matrisiyle ödüller daha dengeli oldu ve sarkacın kararlılığı daha uzun süreler boyunca sağlanmış oldu.

4.2.5 Çift ters sarkacın mücadele ödül fonksiyonu gelişimi

İki adet ters sarkaç birlikte mücadele ettiğinde, aracın savaşmayı öğrenmesi için rakip sarkacın devrilmesinin de odaktaki araca pozitif bir etkisi olması gerekir. Bu etki, dengeden ne kadar uzaklaştırıldıysa o kadar büyük olmalıdır. Bundan dolayı tek ters sarkaç için tasarlanan ödül fonksiyonunu baz alarak, karşı aracın ödülünün negatif değeri ödül olarak deviren araca eklenir.

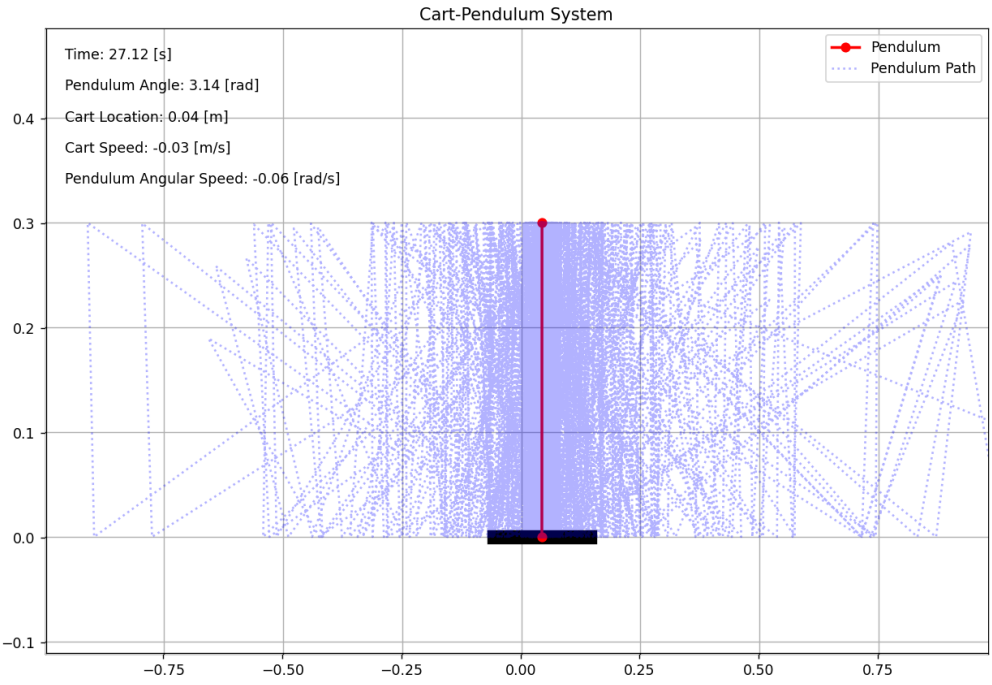
Burada karşı aracın cezasının ödül olarak direkt eklenmesi adamı hiçbir şey yapmadan rakip aracın devrilmesini bekleyerek buradan fayda sağlama davranışına ittiği için, ödül fonksiyonu yine rakip durumundan etkilenecek fakat karşı aracın ödülü düşük bir ağırlıkla eklenerek tüm amacın karşıyı devirmek olmasına engel olacak şekilde tasarlanmalıdır.

$$reward_{\ total} = W_1 * reward_{main} - W_2 * reward_{opponent} \tag{4.6}$$

.

4.3 Ters Sarkacın Kendi Başına Dengeyi Sağlama Süreci

Geliştirilen pekiştirmeli öğrenme sisteminde tek ters sarkacın denge kontrolü başarıyla gerçekleştirilmiştir. Eğitim sürecinde kullanılan hiperparametreler, sistemin kararlı ve verimli bir şekilde öğrenmesini sağlayacak şekilde optimize edilmiştir. Öğrenme oranının 0.0001 gibi düşük bir değerde tutulması, adamın öğrenme sürecinde aşırı değişimlerden kaçınmasını sağlarken, 0.99995 olarak belirlenen epsilon azalma değeri, keşif ve sömürü dengesi açısından sisteme esneklik kazandırmıştır. Gamma değerinin 0.99 olarak belirlenmesi ise, gelecekteki ödüllerin mevcut kararlarda yeterli ağırlığa sahip olmasını sağlamıştır.

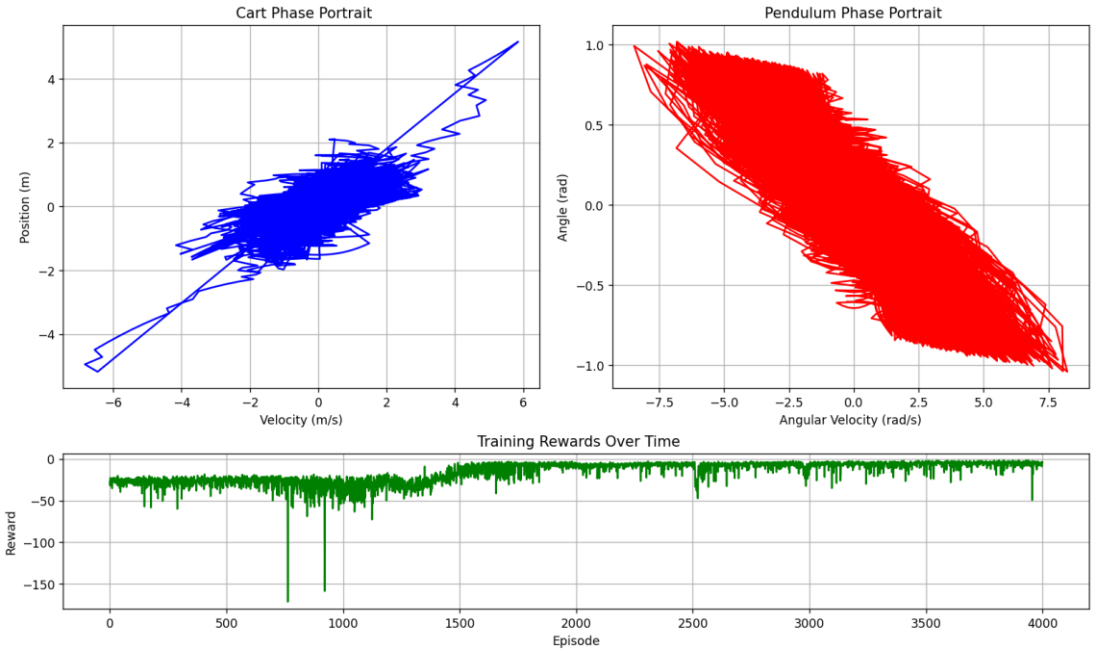


Şekil 4.1 : Ters sarkacın eğitim esnasında izlediği yolları gösteren bir diyagram

Eğitim sürecinin performans analizi, sistemin öğrenme kabiliyetini net bir şekilde ortaya koymaktadır. Başlangıçta yüksek olan ceza değerleri, eğitim ilerledikçe belirgin bir düşüş eğilimi göstermiş ve sıfıra yakınsamıştır. Bu düşüş trendi, adamın ters sarkaç sisteminin dinamiklerini başarıyla öğrendiğini ve optimal kontrol stratejilerini geliştirdiğini göstermektedir. 2500 episode'luk eğitim süreci sonunda, adam saniyeler

mertebesinde dengede kalabilme yeteneği kazanmıştır. Bu süre, sistemin pratik uygulamalar için yeterli kararlılığa ulaştığını göstermektedir.

Geliştirilen sistemin öğrenme sürecindeki en dikkat çekici özelliklerinden biri, eğitimin erken aşamalarında bile anlamlı ilerleme kaydedebilmesidir. Adam, başlangıçtaki başarısız dengeleme denemelerinden hızla öğrenmiş ve çevresel dinamiklere uyum sağlamıştır. Bu hızlı adaptasyon kabiliyeti, seçilen hiperparametrelerin ve ödül fonksiyonunun etkinliğini doğrulamaktadır. Sonuç olarak, geliştirilen sistem, tek ters sarkaç probleminin kontrol gereksinimlerini başarıyla karşılamış ve istikrarlı bir performans sergilemiştir.

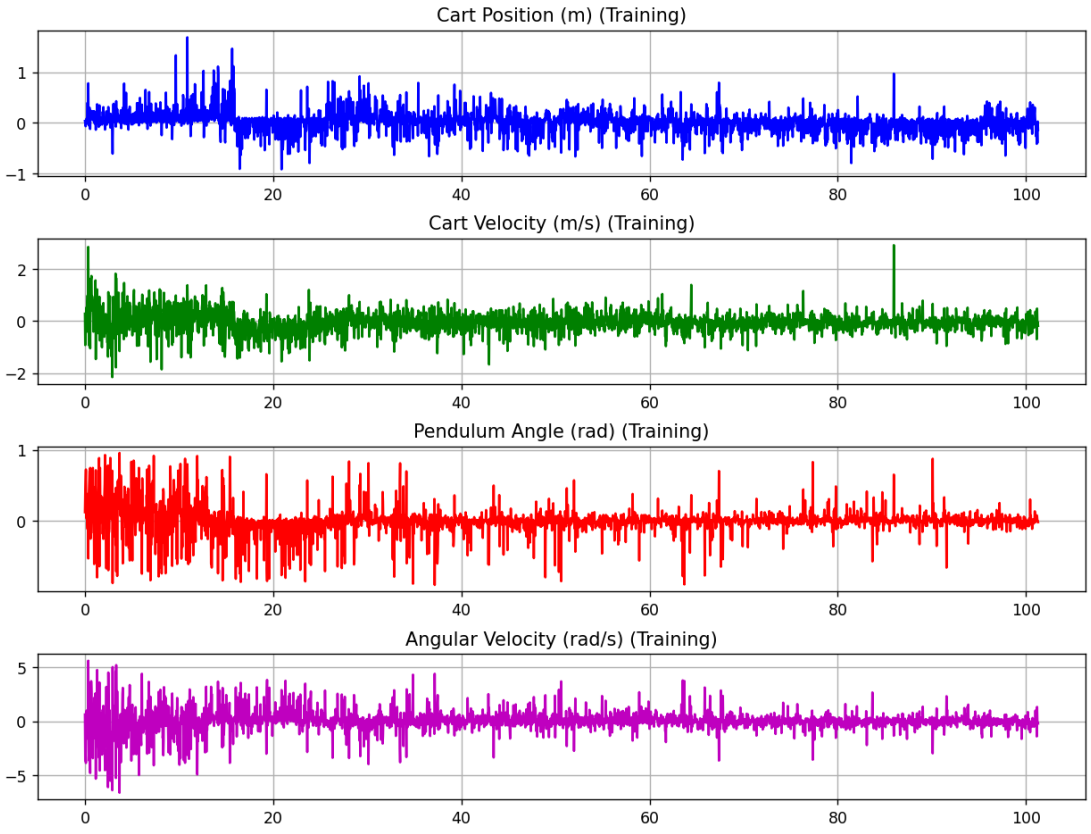


Şekil 4.2 : Ters sarkacın eğitim boyunca faz portreleri ve ödül değişimi grafikleri

4.4 Ters Sarkacın Rastgele Darbeler Karşısında Dengeyi Sağlama Süreci

Sistemin sağlamlığını test etmek amacıyla, eğitim sürecine Poisson dağılımlı rastgele bozucu kuvvetler eklenmiştir. Bu bozucu kuvvetler, gerçek dünya uygulamalarındaki beklenmedik dış etkileri simüle etmek üzere tasarlanmıştır. Eğitim sürecinin bu aşamasında, adam daha zorlu ve dinamik bir çevresel koşulla karşı karşıya bırakılmış,

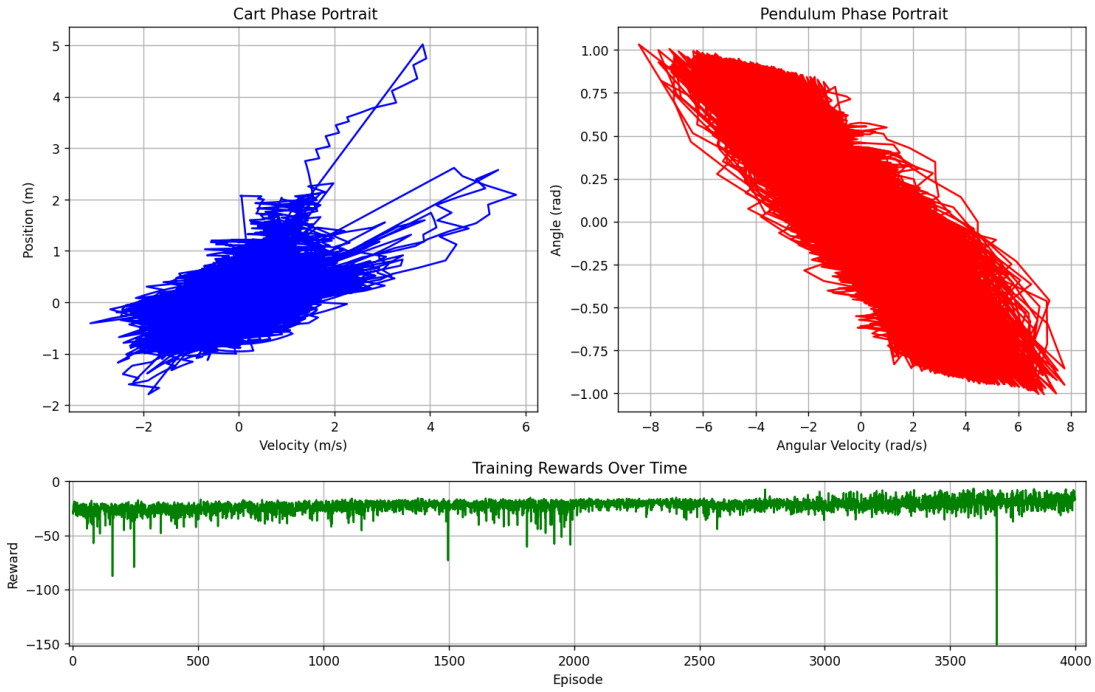
böylece sistemin adaptasyon kabiliyeti sınanmıştır. Başlangıç aşamasında bozucu kuvvetlerin varlığı adamın dengeleme performansını olumsuz etkilese de, eğitim ilerledikçe sistem bu yeni dinamiklere uyum sağlamayı başarmıştır. İlk aşamada kasıtlı olarak dış kuvvet ajan durumlarına dahil edilmediği için eğitim uzun sürmüştür. Çift ters sarkaç için yeniden oluşturulan yazılımda adam dış kuvvetleri dikkate alan ve kaldığı yerden eğitime devam edebilen bir yapıda olduğu için, çok daha hızlı bir şekilde adapte olduğu gözlemlenebilecektir.



Şekil 4.3 : Ters sarkacın rastgele darbeler aldığı eğitim boyunca izlediği durum değişimi aşamaları grafiği

Bozucu kuvvetlerin varlığında gerçekleştirilen eğitim süreci, adamın öğrenme kapasitesini ve esnekliğini ortaya koymuştur. Hiperparametrelerde herhangi bir modifikasyon yapılmaksızın, sistem çevresel bozulmalara karşı etkili stratejiler geliştirmeyi başarmıştır. Bu durum, adamın öğrendiği kontrol stratejilerinin yalnızca ideal koşullarda değil, aynı zamanda bozulmuş durumlarda da geçerli olduğunu

göstermektedir. Sistemin bozucu kuvvetlere karşı gösterdiği hızlı adaptasyon yeteneği, öğrenilen politikaların sağlamlığını ve genellenebilirliğini kanıtlamaktadır.



Şekil 4.4 : Ters sarkacın rastgele darbeler aldığı eğitim boyunca faz portreleri ve ödül değişimi grafikleri

Pekiştirmeli öğrenme adamının bozucu kuvvetlere karşı geliştirdiği bu adaptif davranış, sistemin pratik uygulamalardaki potansiyelini göstermektedir. Adam, öngörülemeyen çevresel değişimlere karşı robust bir kontrol stratejisi geliştirmeyi başarmış ve bu zorlu koşullar altında bile dengeleme görevini başarıyla yerine getirmiştir. Bu sonuçlar, geliştirilen sistemin gerçek dünya uygulamalarındaki belirsizliklere ve bozucu etkilere karşı dayanıklı olduğunu ve pratik kullanım için uygun bir çözüm sunduğunu göstermektedir.

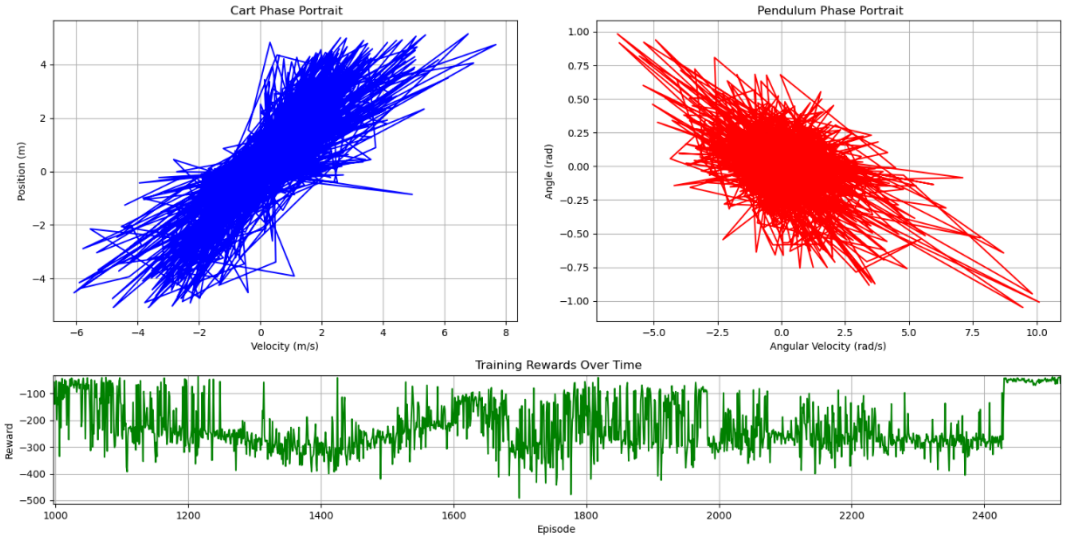
4.5 Çift Ters Sarkacın Kendi Başına Dengeyi Sağlama Süreci

Çift ters sarkaç sisteminde dengeyi sağlayacak kontrol mekanizmalarının geliştirilmesi için ilk aşamada sistem, herhangi bir dış müdahale veya bozucu etkiden bağımsız bir şekilde çalıştırılmıştır. Bu süreçte, yalnızca sarkaç sisteminin kendi dinamikleri göz

önüne alınmıştır. Karşı tarafa ait tüm durumlar [0, 0, 0, 0] vektörü olarak verilmiş ve çıkışın saldırı tarafı ve dışarıdan gelen kuvvetlerin gösterimi sabit 0 N kuvvetine ayarlanmıştır. Böylece, kontrol sistemi sadece kendi durumunu gözlemlemiş ve bu bilgilere dayanarak denge kuvvetini belirlemiştir.

Eğitim sürecinde, sistem toplamda 2600 bölüm boyunca çalıştırılmıştır. Her bölümde kontrol mekanizmasının, sistemin devrilmeden maksimum 200 adım boyunca dengeyi sürdürebilmesi hedeflenmiştir. Başlangıç aşamalarında keşif mekanizmasının da etkisiyle özellikle epsilon minimuma inene kadarki süreçte geçen ilk 900 bölümde sistemin performansı düşük seviyelerde seyretmiş ve sarkaç sıklıkla devrilmiştir. Ancak eğitim ilerledikçe, kontrol mekanizması daha başarılı stratejiler geliştirmiştir.

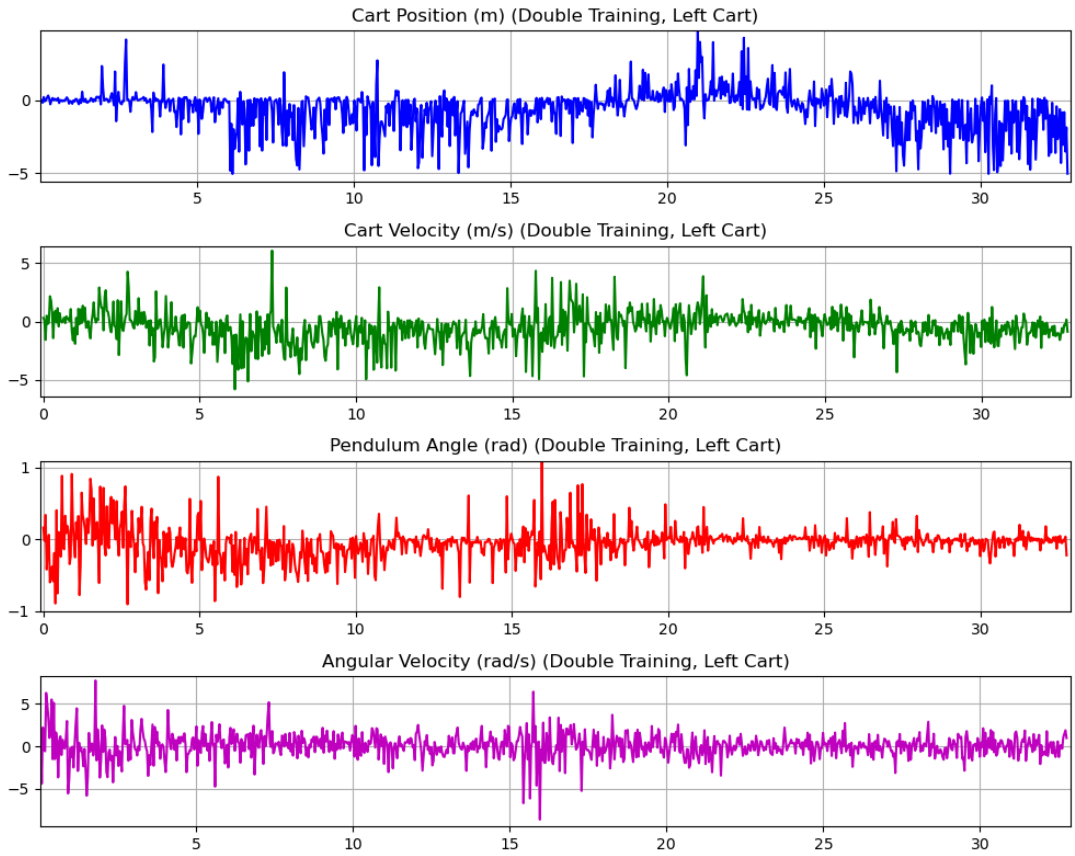
Şekil 4.5'te görüldüğü üzere, başlangıçta düzensiz ve düşük seviyelerde olan ödüller, ilerleyen bölümlerde istikrarlı bir artış göstermiştir. Eğitimin belirli aşamasında erken bölüm bitirme cezası artırıldıktan sonra ödüllerde doğal bir düşüş görülse de, adam bu düşüsü zamanla toparlamış ve cezadan kaçarak eğitimi tutarlı hale getirmiştir. Bu artış eğilimi, kontrol mekanizmasının denge sağlama yeteneğinin geliştiğini kanıtlar niteliktedir. Maksimum adım sayısına ulaşılması ve ödül grafiğindeki bu artış, eğitimin başarıyla tamamlandığının göstergeleri olarak değerlendirilmiştir. Ayrıca sunulan faz portresi, sistemin durum değişkenlerinin (açı ve konumun) faz uzayındaki hareketini göstermektedir. Başlangıçta kaotik olan sistem davranışı, eğitim sürecinin sonlarına doğru kararlı bir denge noktası etrafında düzenli salınımlara dönüşmüştür. Bu durum, sistemin denge kontrolünü başarıyla öğrendiğinin bir diğer kanıtıdır.



Şekil 4.5 : Darbesiz eğitim sürecinde sistemin açı ve konum hız portreleri ve ödül değerinin bölüm sayısına göre değişimi.

Temel denge davranışlarının öğrenilmesinin ardından, sisteme Poisson dağılımına göre rastgele bozucu kuvvetler uygulanarak yeni bir eğitim süreci başlatılmıştır. Bu süreçte, sistemin dengeyi bozacak darbelere karşı kendisini toparlaması ve savunma stratejileri geliştirmesi amaçlanmıştır. Eğitimin başlangıcında, epsilon parametresi sıfırlanarak yeniden ε = 1.00 olarak ayarlanmış ve keşif süreci yeniden başlatılmıştır. Eğitim esnasında adamın eski durumu, hafızası, ağırlıkları ve tüm değerleri korunarak önceki eğitimden kaldığı yerden devam etmesi sağlanmıştır. Ayrıca giriş değerlerinde ek bir kuvvet görüntülenmeye başlanmış ve adam kendisine yapılmakta olan bir darbeyi tanır hale gelmiştir. Önceki yaklaşımımızdan farklı olarak bu noktada adamın kendisine uygulanan darbeyi de bir giriş olarak görmüş olması, darbelere karşı daha hızlı strateji geliştrebilmesine yardımcı olmuştur.

Şekil 4.6'da görüldüğü üzere, rastgele darbelerin etkisiyle sistem durumlarında (açı ve konum) ani sapmalar meydana gelmiştir. Ancak kontrol mekanizması, bu sapmaların ardından sistemi dengede tutmayı başarmıştır. Bu toparlanma yeteneği, eğitimin ilerleyen aşamalarında daha da gelişmiştir.



Şekil 4.6 : Poisson dağılımlı bozucu kuvvetler altında sistem durumlarının zamana göre değişimi.

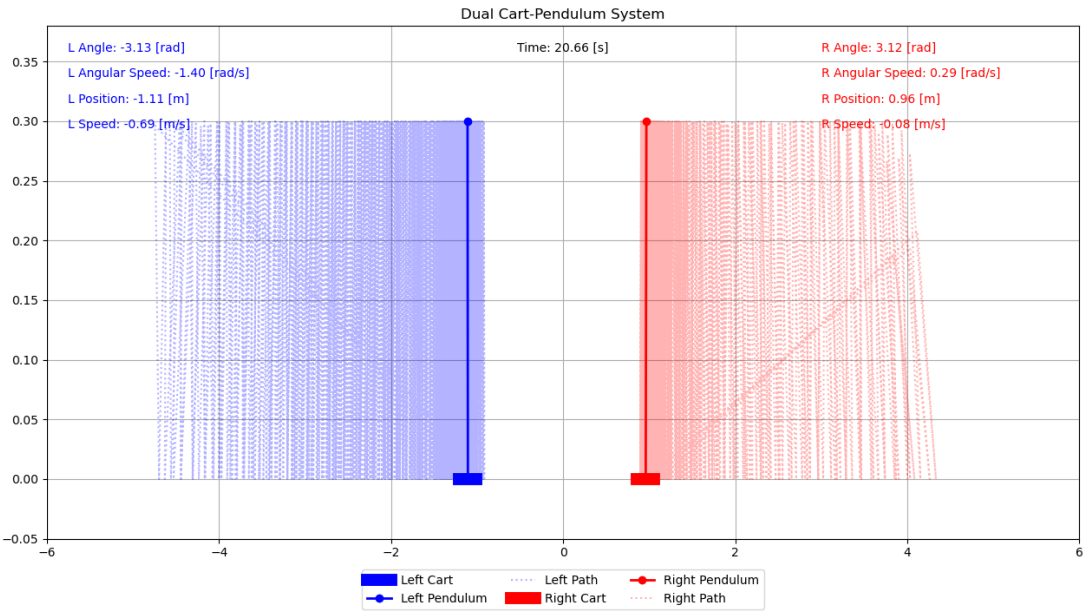
Başlangıçta dalgalı bir seyir izleyen ödül değerleri, 1500 bölümlük eğitim sürecinin sonunda daha istikrarlı bir seviyeye ulaşmıştır. Bu durum, sistemin rastgele darbelere karşı etkili savunma stratejileri geliştirdiğine emin olmak için yeterlidir.

Sonuç olarak, her iki eğitim aşamasında da sistem başarılı bir öğrenme performansı sergilemiştir. İlk aşamada temel denge davranışları kazanılmış, ikinci aşamada ise bu davranışlar bozucu etkilere karşı güçlendirilmiştir. Bu sonuçlar, geliştirilen kontrol mekanizmasının hem temel denge durumunda hem de bozucu etkiler altında etkin bir şekilde çalıştığını kanıtlamaktadır.

4.6 Çift Ters Sarkacın Birbirlerine Kuvvet Uygulama Süreci

Denge kontrolünün başarıyla öğrenilmesinin ardından, sistem iki aracın etkileşimini içerecek şekilde genişletilmiştir. Bu aşamada, aynı dinamik denklemlerden beslenen iki araç, tek bir kontrol mekanizması tarafından yönetilmiştir. Önceki aşamadan farklı olarak, kontrol sistemi artık karşı tarafın durum bilgisine erişebilmekte, kendisine uygulanan kuvveti algılayabilmekte ve karşı tarafa uygulayacağı kuvveti değerlendirebilmektedir.

Etkileşim aşamasına geçildiğinde, keşif sistemi sıfırlanmış ancak önceki aşamada elde edilen tecrübeler ve ağırlıklar korunmuştur. Bu süreçte özgün bir yaklaşım benimsenmiştir: sistem, saldırı stratejilerini keşfederken denge kontrolü için geçmiş tecrübelerinden faydalanmıştır. Güvenli bir etkileşim ortamı sağlamak amacıyla, saldırı uzayı denge uzayının %15'i ile sınırlandırılmıştır. Bu düzenlemeyle kontrol mekanizması, denge için [-10, +10] N aralığından, saldırı için ise [-1.5, +1.5] N aralığından kuvvet değerleri seçebilmiştir. Şekil 4.7'de bilgisayar ortamında oluşturulan bir simülasyon ortamı görüntülenebilmektedir.



Şekil 4.7 : Çift ters sarkaç müsabaka sırasında izledikleri pozisyonları gösteren simülasyon animasyonundan bir görüntü.

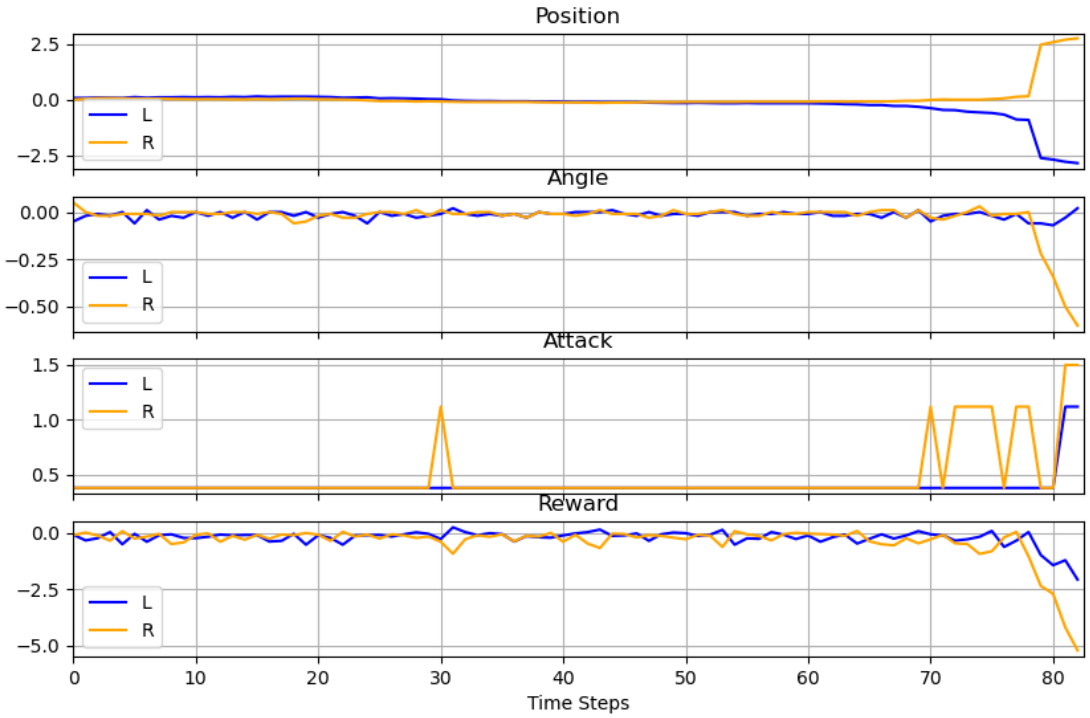
Eğitimin ilk 900 bölümünde keşif oranı minimum seviyeye inmiş ve toplamda 1300 bölümlük bir süreç tamamlanmıştır. Bu eğitimin sonunda, aynı kontrol mekanizması tarafından yönetilen iki araç, birbirlerine üstünlük sağlayamadan maksimum 200 adımdan oluşan bölümleri çoğunlukla yenişemeden ve yüksek puanlarla tamamlamıştır. Bu sonuçlar, kontrol mekanizmasının her iki aracı da dengeli ve kararlı bir şekilde yönettiğini göstermektedir. Şekil 4.8'de görüntülenen grafikte her bölümün toplam ödülleri gösterilmiştir. Bölümü erken kaybedip devrilen aracın ödülü erken bitirme cezasıyla dramatik olarak yenen araçta 0, yenilen araçta -300 seviyesinde ulaşmakta, yenişemeden bitirilen bölümlerde ise standart olarak dengede durma davranışı öğrenildiği için her bölümde -1 civarında alınan ödüllerin toplanmasıyla -200 seviyesinde bitirilmektedir.

Şekil 4.8 : Çift ters sarkaçın test edilmesi sırasında alınan toplam ödüllerinin bölümlere göre değişimi.

Eğitim tamamlandıktan sonra bir test süreci başlatılmıştır. Test sırasında kontrolcü yeni veri kullanmamış, durumlar hafızaya kaydedilmemiş ve hiçbir şekilde yeni keşif yapılmamıştır (epsilon = 0.0). Test sürecinde maksimum adım sayısı 1000 olarak ayarlanmış ve toplamda 300 bölüm gerçekleştirilmiştir. Bu bölümlerde ortalama 320 adım süren müsabakalarda, iki araç arasında kararlı ve tutarlı bir denge sağlanmıştır. En uzun süren bölüm 700 adıma ulaşmıştır.

Şekil 4.9'da sunulan test sonuçları, heyecanlı bir çekişmenin portresini çizmektedir. Bölüm sonuna doğru bir tarafın açı sınırını aşarak devrilmesi ve erken devrilen aracın devrilme cezasıyla ödülünün aşağıya çekilmesi net bir şekilde izlenebilmektedir.

Müsabaka sırasında ajanların birbirlerine uyguladıkları kuvvetler de diyagramda görüntülenmektedir.



Şekil 4.9 : Test sürecinde en uzun bölümlerden olan, 700 adım süren bölümde araçların konum, açı, saldırı kuvveti ve adım başı ödüllerinin değişimleri.

Test bölümlerinde gerçekleşen müsabakalar ortalama 320 adım sürmüş ve çoğunlukla kıl payı yenişmelerle sonuçlanmıştır. En uzun süren karşılaşma 700 adıma ulaşmıştır. Grafikte görüldüğü üzere, her iki aracın açı ve konum değişimleri ile birbirlerine uyguladıkları kuvvetler, kontrol mekanizmasının hem denge kontrolünü hem de etkileşim stratejilerini başarıyla öğrendiğini kanıtlamaktadır.

Sonuç olarak, geliştirilen kontrol mekanizması hem denge kontrolünü hem de kontrollü etkileşimi başarıyla gerçekleştirebilmiştir. Sistem, önceki aşamada öğrendiği denge stratejilerini korurken, yeni öğrendiği etkileşim stratejileriyle iki aracı dengeli bir şekilde yönetebilmiştir.

5. SONUÇ VE ÖNERİLER

Bu tez çalışmasında, derin pekiştirmeli öğrenme yaklaşımı kullanılarak tek ve çift ters sarkaç sistemlerinin kontrolü gerçekleştirilmiş ve kapsamlı deneysel sonuçlar elde edilmiştir. Çalışmanın temel odak noktası olan DQN algoritmasının karmaşık kontrol problemlerindeki etkinliği, sistematik deneyler ve analizler yoluyla kanıtlanmıştır.

Gerçekleştirilen hiperparametre optimizasyonu çalışmaları, sistemin öğrenme performansını önemli ölçüde etkilediğini göstermiştir. Özellikle öğrenme oranının 0.0001 seviyesinde tutulması, sistemin aşırı salınımlar yapmadan kararlı bir şekilde öğrenmesini sağlamıştır. Epsilon azalma oranının 0.99995 olarak belirlenmesi, özellikle çift sarkaç sisteminde etkileşimlerin karmaşıklığı nedeniyle daha uzun keşif süreleri gerektiren durumlarda, keşif ve sömürü arasındaki hassas dengenin korunmasına olanak tanımıştır.

Çalışma kapsamında geliştirilen ödül fonksiyonu, özellikle Q matrisi kullanılarak yapılan matris tabanlı karesel ceza yaklaşımıyla, sistemin etkin bir öğrenme süreci gerçekleştirmesinde kritik rol oynamıştır. Matematiksel model, sarkaç kütlesi, araba kütlesi ve sürtünme katsayısı gibi temel fiziksel parametreleri içermekte olup, bu parametreler sistemin doğru simülasyonu ve kontrolü için büyük önem taşımıştır.

5.1 Sonuçlar

Tek ters sarkaç sisteminde gerçekleştirilen 2500 bölümlük eğitim süreci sonunda, geliştirilen kontrol mekanizması saniyeler mertebesinde kararlı denge kontrolü sağlayabilmiştir. Sistem, Poisson dağılımlı rastgele bozucu kuvvetlere karşı gösterdiği dayanıklılık ile gerçek dünya uygulamaları için uygunluğunu kanıtlamıştır. -10N ile +10N arasında değişen dış kuvvetlere maruz kaldığında bile dengesini koruyabilmesi, sistemin adaptif yeteneklerini göstermektedir.

Çift ters sarkaç sisteminde ise, tek bir kontrol mekanizması ile iki aracın etkileşimli kontrolü başarıyla gerçekleştirilmiştir. Yapay sinir ağı mimarisi, artan karmaşıklığı yönetmek için genişletilmiş, giriş katmanları dokuz durum değişkenini işlerken çıkış katmanları 18 olası eylemi yönetmiştir. Test süreçlerinde gözlemlenen ortalama 320 adım süren müsabakalar ve maksimum 700 adıma ulaşan karşılaşmalar, sistemin kararlı ve tutarlı bir performans sergilediğini göstermiştir. Sistem, denge kuvvetlerini korurken saldırı kuvvetlerini de -1.5N ile +1.5N arasında etkin bir şekilde yönetebilmiştir.

Q matrisi aracılığıyla farklı durum değişkenlerinin optimal ağırlıklandırılması, kontrol performansının iyileştirilmesinde önemli bir faktör olmuştur. Bu yaklaşım, sistemin hem denge kontrolünü sağlamasında hem de bozucu etkilere karşı dayanıklı bir yapı geliştirmesinde etkili olmuştur. Elde edilen performans seviyesi, sistemin endüstriyel ve akademik uygulamalar için yeterli olgunluğa eriştiğini göstermektedir.

5.2 Öneriler

Gerçekleştirilen çalışmanın sonuçları ışığında, gelecekteki araştırmalar için bir dizi önemli öneri sunulmaktadır. Öncelikle, geliştirilen sistemin gerçek fiziksel ortamlarda test edilmesi, simülasyon sonuçlarının doğrulanması ve pratik uygulanabilirliğin değerlendirilmesi açısından kritik önem taşımaktadır. Bu amaçla, laboratuvar ortamında prototip uygulamalar geliştirilmeli ve sistem performansı gerçek koşullar altında analiz edilmelidir.

Sistemin performansının daha da iyileştirilmesi için, farklı derin öğrenme mimarileri ve pekiştirmeli öğrenme algoritmaları ile karşılaştırmalı analizler yapılması önerilmektedir. DDPG, TD3 veya SAC gibi modern algoritmaların sistem üzerindeki etkinliğinin incelenmesi, kontrol performansının optimize edilmesine katkı sağlayabilir. Ayrıca, transfer öğrenme ve meta-öğrenme gibi ileri düzey yaklaşımların sisteme entegre edilmesi düşünülebilir.

Çift ters sarkaç sisteminde, araçlar arasındaki etkileşimin daha karmaşık senaryolar altında incelenmesi önemli bir araştırma alanı olarak öne çıkmaktadır. Farklı başlangıç koşulları, çevresel etkiler ve çoklu araç etkileşimlerinin sisteme dahil edilmesi, kontrol mekanizmasının kapasitesini artırabilir. Bunun yanında, ödül fonksiyonunun adaptif olarak ayarlanması ve sistem parametrelerinin çevrimiçi optimizasyonu gibi konular da gelecek çalışmalarda ele alınmalıdır.

5.3 Bilimsel Katkılar

Bu tez çalışması, kontrol sistemleri ve yapay zeka alanlarına bir dizi özgün katkı sağlamıştır. İlk olarak, tek ve çift ters sarkaç sistemlerinin kontrolünde kullanılan matris tabanlı karesel ceza fonksiyonu yaklaşımı, literatürde benzer sistemler için kullanılabilecek etkili bir ödül mekanizması sunmaktadır. Bu yaklaşım, sistem durumlarının optimal ağırlıklandırılmasına olanak tanıyarak, kontrol performansının iyileştirilmesinde önemli bir rol oynamıştır.

Çalışmanın bir diğer önemli katkısı, bozucu kuvvetlerin varlığında sistem performansının iyileştirilmesi için geliştirilen özgün eğitim stratejisidir. Bu strateji, sistemin gerçek dünya uygulamalarındaki gürbüzlüğünü artırmış ve belirsiz koşullar altında bile kararlı kontrol sağlanmasını mümkün kılmıştır. Özellikle Poisson dağılımlı rastgele bozucu kuvvetlere karşı geliştirilen adaptif davranış, benzer kontrol problemleri için örnek teşkil edebilecek niteliktedir.

Çift ters sarkaç sisteminde, tek bir kontrol mekanizması ile iki aracın etkileşimli kontrolü için sunulan yenilikçi yaklaşım, literatüre özgün bir katkı niteliğindedir. Bu yaklaşım, çoklu sistem kontrolü ve etkileşimli öğrenme alanlarında yeni araştırma fırsatları sunmaktadır. Ayrıca, hiperparametre optimizasyonu için geliştirilen sistematik metodoloji, benzer derin pekiştirmeli öğrenme uygulamalarında referans olarak kullanılabilecek değerli bir kaynak oluşturmaktadır.

Sonuç olarak, bu çalışmanın ortaya koyduğu bulgular ve geliştirilen metodolojiler, derin pekiştirmeli öğrenme yaklaşımlarının karmaşık kontrol problemlerindeki potansiyelini göstermekte ve gelecekteki araştırmalar için sağlam bir temel oluşturmaktadır. Elde edilen sonuçlar, özellikle robotik sistemler ve otonom kontrol uygulamaları gibi alanlarda pratik değer taşımakta ve endüstriyel uygulamalar için umut verici perspektifler sunmaktadır.

ETİK KURALLAR UYUM BEYANI

Aşağıda belirtilen mühendisliğin temel ilkelerini biliyor ve kabul ediyorum.

Muhammet	**Işık**	**İMZA/TARİH:**

Mühendisler; mühendislik mesleğinin doğruluğunu, onurunu ve değerini insanlığın refahının artması için kendi bilgi ve becerilerini kullanarak, dürüst ve tarafsız olarak halka, kendi işverenlerine ve müşterilerine sadakatle hizmet ederek, mühendislik mesleğinin yeteneğini ve prestijini artırmaya çabalayarak, kendi disiplinlerinin mesleki ve teknik birliğini destekleyerek yüceltir ve geliştirirler.

- Mühendisler, mesleki görevlerini yerine getirirken toplumun güvenliğini, sağlığını ve rahatını en önde tutacaktır.
- Mühendisler, sadece yetkili oldukları alanlarda hizmet vereceklerdir.
- Mühendisler, sadece objektif ve gerçekçi raporlar düzenleyeceklerdir.
- Mühendisler, mesleki konularda işveren veya müşteri için güvenilir vekil veya yardımcı olarak davranacaklar ve çıkar çatışmalarından kaçınacaklardır.
- Mühendisler mesleki itibarlarını hizmetlerinin gereğine göre tesis edecekler ve diğer meslektaşlarıyla haksız rekabete girmeyeceklerdir.
- Mühendisler, meslek doğruluğunu, onurunu ve değerini yüceltmek ve geliştirmek için çalışacaklardır.
- Mühendisler, mesleki gelişmelerini kendi kariyerleriyle devam ettirecekler ve kendi kontrolleri altındaki mühendislerin mesleki gelişmeleri için olanaklar sağlayacaklardır.

Bu raporda herhangi bir kaynaktan alıntı yapılmış kısımlar %15'den az, ve paragraf halinde birebir alıntı sayısının ise sıfır olduğunu beyan ediyorum.

Muhammet	**Işık**	**İMZA/TARİH:**

IEEE ETİK KURALLARI

IEEE Etik Kuralları
IEEE Code of Ethics

IEEE üyeleri olarak bizler bütün dünya üzerinde teknolojilerimizin hayat standartlarını etkilemesindeki önemin farkındayız. Mesleğimize karşı şahsi sorumluluğumuzu kabul ederek, hizmet ettiğimiz toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmayı söz verdiğimizi ve aşağıdaki etik kuralları kabul ettiğimizi ifade ederiz.

1. **Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;**
2. **Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;**
3. **Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;**
4. **Her türlü rüşveti reddetmek;**
5. **Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;**
6. **Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;**
7. **Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriyi kabul etmek ve eleştiriyi yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;**
8. **Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği, veya cinsiyet ifadesi üzerinden ayırımcılık yapma durumuna girişmemek;**
9. **Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;**
10. **Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.**

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.

STANDARTLAR VE KISITLAR : Raporda aşağıda istenenler doldurularak eklenecektir.

Bitirme Projesinin hazırlanmasında Standartlar ve Kısıtlarla ilgili olarak, aşağıdaki soruları cevaplayınız.

1. **Projenizin tasarım boyutu nedir? Açıklayınız.**
   Var olan bir projenin üzerine koyularak ilerlenmiş olup, ters sarkaçları birbiriyle yarıştırma ve aynı sinir ağı tarafından idare etmeyi amaçlamıştır.
2. **Projenizde çözüm ürettiğiniz mühendislik problemini ve çözümünüzü kısaca açıklayınız?**
   Projede ters sarkaca uyguladığımız yöntemle bir robot veya başka bir yapı da bir amaca uygun şekilde kendi kendine öğrenme yoluyla kullanılabilir.
3. **Lisans eğitiminiz süresince almış olduğunuz derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?**
   Akıllı kontrol sistemleri, programlama, sistem modelleme ve simülasyon, bilgisayarlı kontrol sistemleri, mikrokontrol sistemleri gibi derslerde öğrendiğim bilgileri bu projede kullanma şansım oldu?
4. **Projenizi gerçekleştirirken kullandığınız modern araçlar/yazılımlar/programlar vb. nelerdir? Hangi amaçlarla kullandığınızı kısaca açıklayınız.**
   Python yazılım dili ve makine öğrenmesi üzerine yazılmış tensorflow gibi kütüphaneleri kullandım.
5. **Ders dışında çeşitli disiplinleri içeren sertifikanız var mı? (Örneğin CUDA, Udemy, Coursera gibi online platformlarda bilgi sahibi olmak)**
   Programlama üzerine sertifikalarım ve çalışmalarım vardır.
6. **Kullandığınız veya dikkate aldığınız mühendislik standartları/normları nelerdir?**
   Proje konunuzla ilgili olarak kullandığınız ve kullanılması gereken standartları burada kod ve isimleri ile sıralayınız.
7. **Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir?**
   a)Fiziksel kısıtlamalar
   Ortam modeli için gerçek bir sistemin matematik modeline dayalı kısıtlamalar kullanılmıştır. Kontrol algoritması için ise motor ürünlerinin kabiliyetleri dikkate alınarak uygulanacak kuvvetlerin belirlenmesi sağlanmıştır.

Proje Ekibi (Yürütücüsü/Lideri): Muhammet Işık

Proje Konusu: Makine Öğrenmesi Teknikleri Kullanılarak

Bir Dövüşen Robotun Eğitilmesi

Proje Danışmanı: Doç. Dr. Ahmet Onat

Not: Gerek görülmesi halinde bu sayfa istenilen maddeler için genişletilebilir.

KAYNAKLAR

Kirk, D. E. (2004). *Optimal Control Theory: An Introduction*. Prentice Hall.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., … & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529-533.

Krause, P., Teel, A. R., & Zabarankin, M. (2009). *Nonlinear Control of Dynamic Systems*. Princeton University Press.

Slotine, J. J. E., & Li, W. (1991). *Applied Nonlinear Control*. Prentice-Hall.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on Machine Learning* (pp. 387-395).

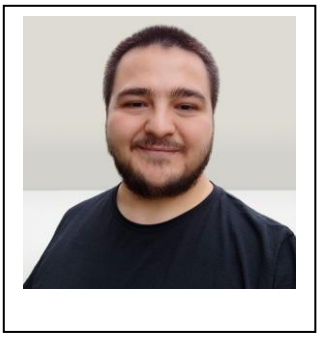
Anderson, B. D. O., & Moore, J. B. (1990). *Optimal Control: Linear Quadratic Methods*. Courier Corporation.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade* (pp. 437-478). Springer, Berlin, Heidelberg.

Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. The *International Journal of Robotics Research*, 32(11), 1238-1274.

Tilbury, D., Messner, W., Hill, R., Taylor, J. D., Das, S., Hagenow, M., & The MathWorks. (2021). *Control Tutorials for MATLAB and Simulink (CTMS): Inverted Pendulum*. University of Michigan. Erişim adresi: https://ctms.engin.umich.edu/CTMS/?example=InvertedPendulum&section=SystemModeling

ÖZGEÇMİŞ



Ad-Soyad **: Muhammet Işık**

Doğum Tarihi ve Yeri **: 13.05.1994 / Konya**

E-posta **:** isikmuhamm@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** **:** 2025, İstanbul Teknik Üniversitesi, Elektrik Elektronik Fakültesi, Robotik ve Otonom Sistemler Mühendisliği Bölümü

MESLEKİ DENEYİM VE ÖDÜLLER:

- Mesleki deneyimi yoktur.

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- Yoktur.

DİĞER YAYINLAR, SUNUMLAR VE PATENTLER:

- Yoktur.

IEEE ETİK KURALLARI

IEEE Etik Kuralları
IEEE Code of Ethics

IEEE üyeleri olarak bizler bütün dünya üzerinde teknolojilerimizin hayat standartlarını etkilemesindeki önemin farkındayız. Mesleğimize karşı şahsi sorumluluğumuzu kabul ederek, hizmet ettiğimiz toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmayı söz verdiğimizi ve aşağıdaki etik kuralları kabul ettiğimizi ifade ederiz.

11. **Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;**
12. **Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;**
13. **Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;**
14. **Her türlü rüşveti reddetmek;**
15. **Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;**
16. **Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;**
17. **Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriyi kabul etmek ve eleştiriyi yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;**
18. **Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği, veya cinsiyet ifadesi üzerinden ayırımcılık yapma durumuna girişmemek;**
19. **Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;**
20. **Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.**

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.

ETİK KURALLAR UYUM BEYANI

Aşağıda belirtilen mühendisliğin temel ilkelerini biliyor ve kabul ediyorum.

Muhammet	**Işık**	**İMZA/TARİH:**

Mühendisler; mühendislik mesleğinin doğruluğunu, onurunu ve değerini insanlığın refahının artması için kendi bilgi ve becerilerini kullanarak, dürüst ve tarafsız olarak halka, kendi işverenlerine ve müşterilerine sadakatle hizmet ederek, mühendislik mesleğinin yeteneğini ve prestijini artırmaya çabalayarak, kendi disiplinlerinin mesleki ve teknik birliğini destekleyerek yüceltir ve geliştirirler.

- Mühendisler, mesleki görevlerini yerine getirirken toplumun güvenliğini, sağlığını ve rahatını en önde tutacaktır.
- Mühendisler, sadece yetkili oldukları alanlarda hizmet vereceklerdir.
- Mühendisler, sadece objektif ve gerçekçi raporlar düzenleyeceklerdir.
- Mühendisler, mesleki konularda işveren veya müşteri için güvenilir vekil veya yardımcı olarak davranacaklar ve çıkar çatışmalarından kaçınacaklardır.
- Mühendisler mesleki itibarlarını hizmetlerinin gereğine göre tesis edecekler ve diğer meslektaşlarıyla haksız rekabete girmeyeceklerdir.
- Mühendisler, meslek doğruluğunu, onurunu ve değerini yüceltmek ve geliştirmek için çalışacaklardır.
- Mühendisler, mesleki gelişmelerini kendi kariyerleriyle devam ettirecekler ve kendi kontrolleri altındaki mühendislerin mesleki gelişmeleri için olanaklar sağlayacaklardır.

Bu raporda herhangi bir kaynaktan alıntı yapılmış kısımlar %15'den az, ve paragraf halinde birebir alıntı sayısının ise sıfır olduğunu beyan ediyorum.

Muhammet	**Işık**	**İMZA/TARİH:**

STANDARTLAR VE KISITLAR : Raporda aşağıda istenenler doldurularak eklenecektir.

Bitirme Projesinin hazırlanmasında Standartlar ve Kısıtlarla ilgili olarak, aşağıdaki soruları cevaplayınız.

8. **Projenizin tasarım boyutu nedir? Açıklayınız.**
   Var olan bir projenin üzerine koyularak ilerlenmiş olup, ters sarkaçları birbiriyle yarıştırma ve aynı sinir ağı tarafından idare etmeyi amaçlamıştır.
9. **Projenizde çözüm ürettiğiniz mühendislik problemini ve çözümünüzü kısaca açıklayınız?**
   Projede ters sarkaca uyguladığımız yöntemle bir robot veya başka bir yapı da bir amaca uygun şekilde kendi kendine öğrenme yoluyla kullanılabilir.
10. **Lisans eğitiminiz süresince almış olduğunuz derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?**
    Akıllı kontrol sistemleri, programlama, sistem modelleme ve simülasyon, bilgisayarlı kontrol sistemleri, mikrokontrol sistemleri gibi derslerde öğrendiğim bilgileri bu projede kullanma şansım oldu?
11. **Projenizi gerçekleştirirken kullandığınız modern araçlar/yazılımlar/programlar vb. nelerdir? Hangi amaçlarla kullandığınızı kısaca açıklayınız.**
    Python yazılım dili ve makine öğrenmesi üzerine yazılmış tensorflow gibi kütüphaneleri kullandım.
12. **Ders dışında çeşitli disiplinleri içeren sertifikanız var mı? (Örneğin CUDA, Udemy, Coursera gibi online platformlarda bilgi sahibi olmak)**
    Programlama üzerine sertifikalarım ve çalışmalarım vardır.
13. **Kullandığınız veya dikkate aldığınız mühendislik standartları/normları nelerdir?**
    Proje konunuzla ilgili olarak kullandığınız ve kullanılması gereken standartları burada kod ve isimleri ile sıralayınız.
14. **Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir?**
    a)Fiziksel kısıtlamalar
    Ortam modeli için gerçek bir sistemin matematik modeline dayalı kısıtlamalar kullanılmıştır. Kontrol algoritması için ise motor ürünlerinin kabiliyetleri dikkate alınarak uygulanacak kuvvetlerin belirlenmesi sağlanmıştır.

Proje Ekibi (Yürütücüsü/Lideri): Muhammet Işık

Proje Konusu: Makine Öğrenmesi Teknikleri Kullanılarak

Bir Dövüşen Robotun Eğitilmesi

Bu proje *Doç. Dr. Ahmet Onat* tarafından onaylanmıştır. *(.....İmza.....)*

Not: Gerek görülmesi halinde bu sayfa istenilen maddeler için genişletilebilir.