BİTİRME TASARIM PROJESİ SUNUMU

MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANILARAK BİR DÖVÜŞEN ROBOTUN EĞİTİLMESİ

Doç. Dr. Ahmet Onat

30.01.2025

Muhammet Işık (040120447)

KIYMETLİ JÜRİ ÜYELERİMİZ
PROF. DR. İLHAN KOCAARSLAN
DOÇ DR. KEMAL UÇAK

DEĞERLİ DANIŞMAN HOCAM
DOÇ. DR. AHMET ONAT

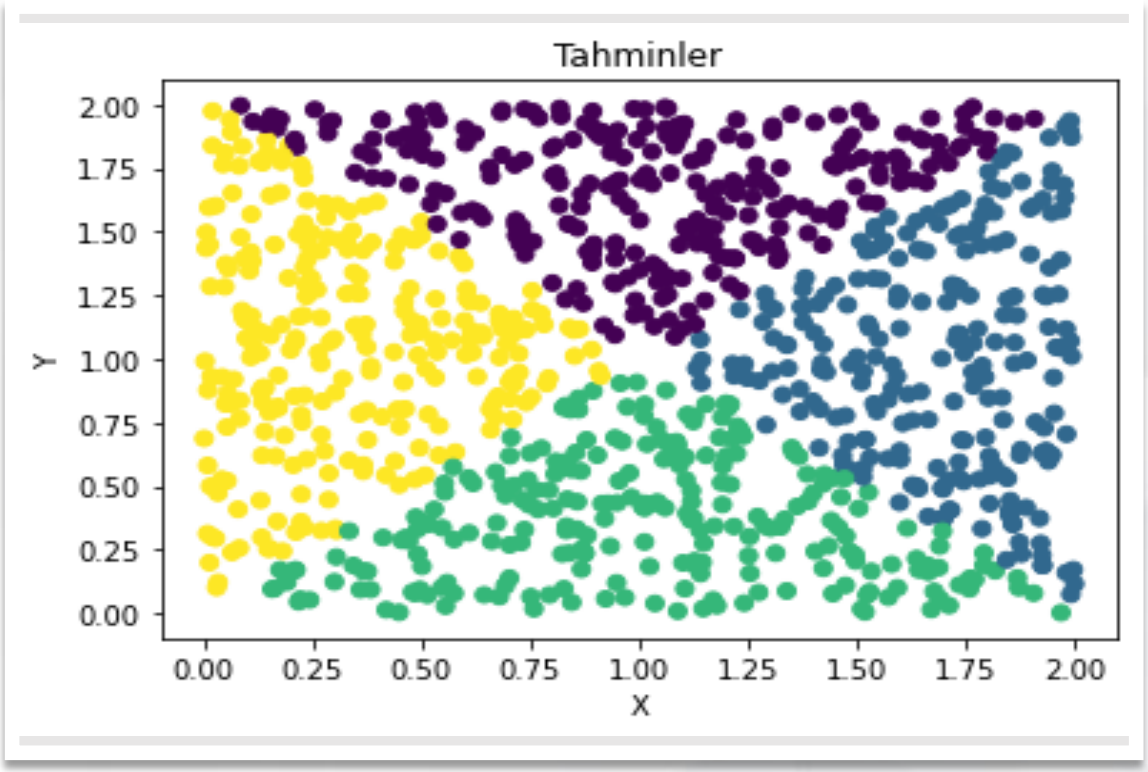
BTP SUNUMUMA HOŞ GELDİNİZ!

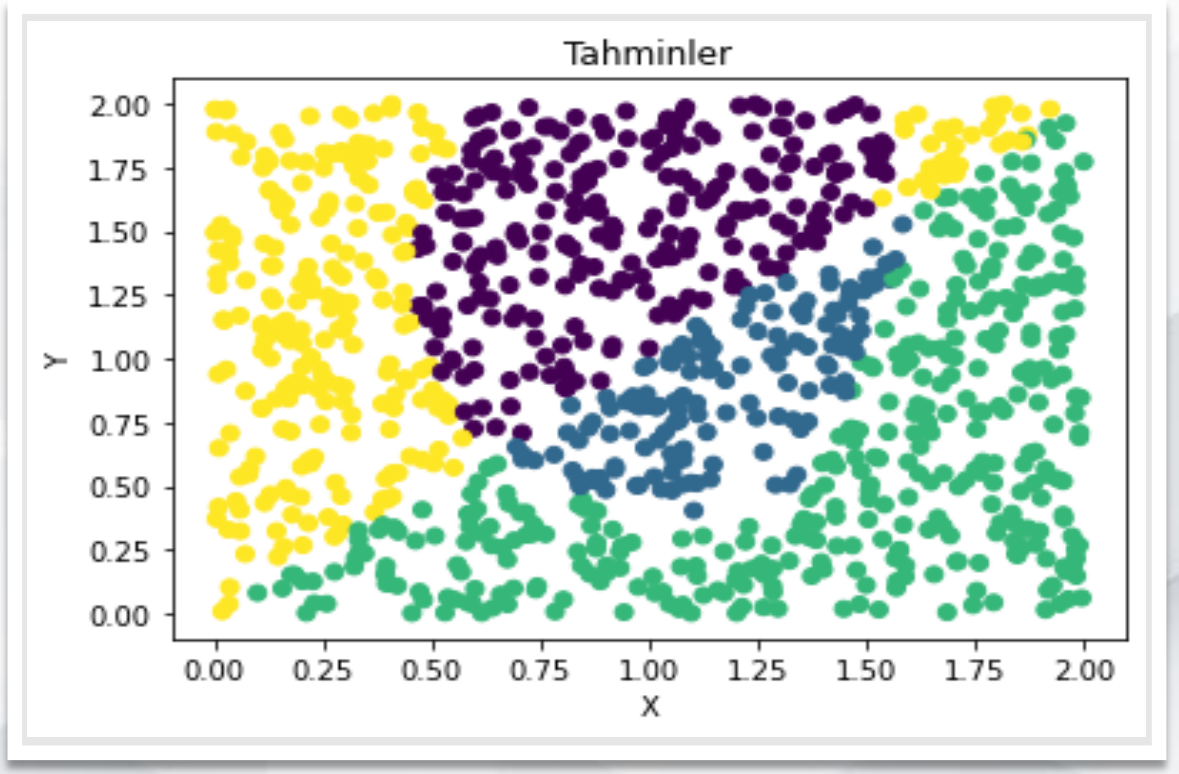
Geçmiş Dönem Özeti

Proje esnasında, giriş dersinde edindiğim bilgilerle kurduğum temeli yoğun bir şekilde kullandım. Python programlama dilinde ilerledim ve proje boyunca gerekli tüm bileşenleri bu dil üzerinde geliştirdim.

- Koordinat sistemi üzerinde bir sınıflandırma problemi çözdüm,
- LQR kontrolcüsüyle bir ters sarkaç kontrolü yaptım,
- LQR kontrol verileriyle bir yapay zeka kontrolcü eğittim,
- Sınıflandırma ve kontrol problemlerini önce tamamen matematiksel temellere dayanarak, ardından da yapay zeka eğitimi için özelleşmiş detaylı yazılım kütüphanelerini kullanarak çözdüm ve aradaki farkları gördüm.

Geçmiş Dönem Özeti



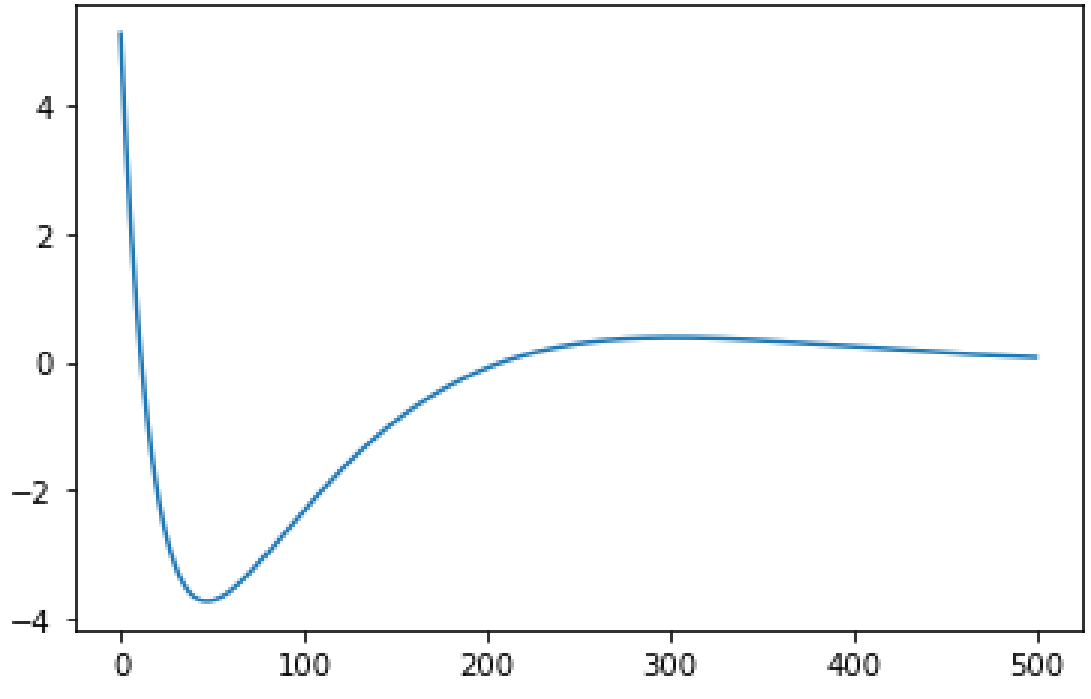


Sınıflandırma Problemi Çalışması

Geçmiş Dönem Özeti

Temsil	Anlamı	Değeri
M	Araç ağırlığı	0.5 [kg]
m	Sarkaç ağırlığı	0.2 [kg]
b	Araç içi sürtünme katsayısı	0.1 [N/m/sec]
l	Sarkacın ağırlık merkezine olan mesafe	0.3 [m]
I	Sarkacın eylemsizlik momenti	0.006 [kg/m2]
F	Araca uygulanan kuvvet	F [N]
x	Aracın anlık pozisyonu	x [m]
θ	Sarkacın dik eksenle yaptığı açı	θ [°]
φ	Aracın anlık ivmesi	φ [m/s^2]

Örnek sarkaç parametreleri

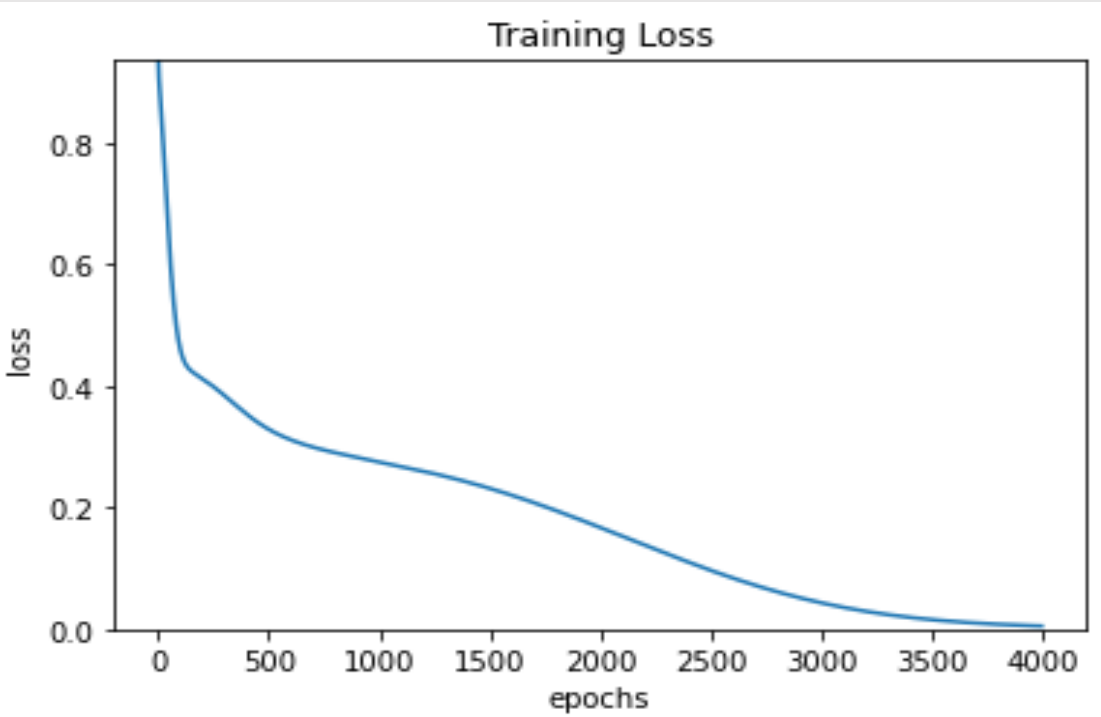


Başlangıç durum vektörü x = [-0.178, 3.870, 0.073, -0.067]^T için kontrolcünün ürettiği kuvvet çıkışı

Geçmiş Dönem Özeti

Parametre	Değeri
Sistemin girdi sayısı	5
Gizli katman 1 için nöron adedi	20
Gizli katman 2 için nöron adedi	20
Çıkış boyutu	1
Öğrenme oranı	0.0001
Yapay sinir ağı eğitim sayısı	1000
Toplam veri sayısı	50,000
Örnekleme zamanı	0.01 s

Yapay sinir ağı eğitim parametreleri



Yapay sinir ağının çıktıları

Çalışmanın Amacı

Çalışmam esnasında bir ters sarkaç ile temsil edilen robotik sistemin önce pekiştirmeli öğrenme özelinde derin Q öğrenmesi ağı (DQN) sistemini kullanarak dengesini sağlamayı öğrenmesini, ardından karşısına geçen bir rakibe her karar alma anında bir kuvvet uygulamak suretiyle dövüşmesini sağlamak üzere rakibi devirmesini öğrenmesi üzerinde uğraştım.

Çalışmam esnasında Michigan Üniversitesi ekibinin hazırladığı bir kontrol örneğinde kullanılan ters sarkaç parametrelerini ve matematik modelini bir değişiklik yapmadan kullandım. Yalnızca bu modelin dik konumu π dereceye denk geldiğinden, bu değerin bizim kontrolcümüze 0 derece olarak yansıması için normalize edildi. Bunun sebebi, eğitim esnasında maliyet fonksiyonunu sıfıra götürebilmekten ibarettir.

Çalışmanın Yöntemi: DQN

DQN, Q-öğrenmesi algoritmasının derin öğrenme ile birleştirilmesiyle geliştirilmiş bir RL yöntemidir. Sistem, bir yapay zeka adamının çevresiyle etkileşimine dayanır. Adam, eğitimin ilk aşamalarında yüksek oranda rastgele aksiyonlar alırken, yaptıklarının sonucuyla yüzleştikçe, geçmiş tecrübelerini kullanmaya başlar. Eğitim esnasında, eğitim boyunca kademeli olarak azalan ε oranında rastgele kararlar alınır.

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

Alınacak kararın belirlenmesi, belirli bir durumda mevcut aksiyondan elde edilen ödül ve ulaşılan yeni durumun hedefimize ne kadar yakın olduğuna dayalı olarak gerçekleştirilir.

$$Q(s_t, a_t) \leftarrow r_t + \gamma \max_{a'} Q(s_{t+1}, a')$$

DQN Nasıl Öğreniyor?

Pekiştirmeli öğrenme algoritmaları aslında bir insanı çok yüksek gerçekçilik oranıyla taklit ederler. Bunu birkaç yaşındaki bir çocuğun öğrenmesi üzerinden örneklendirelim. Bu durumda $\varepsilon = 0.90$ olsun.

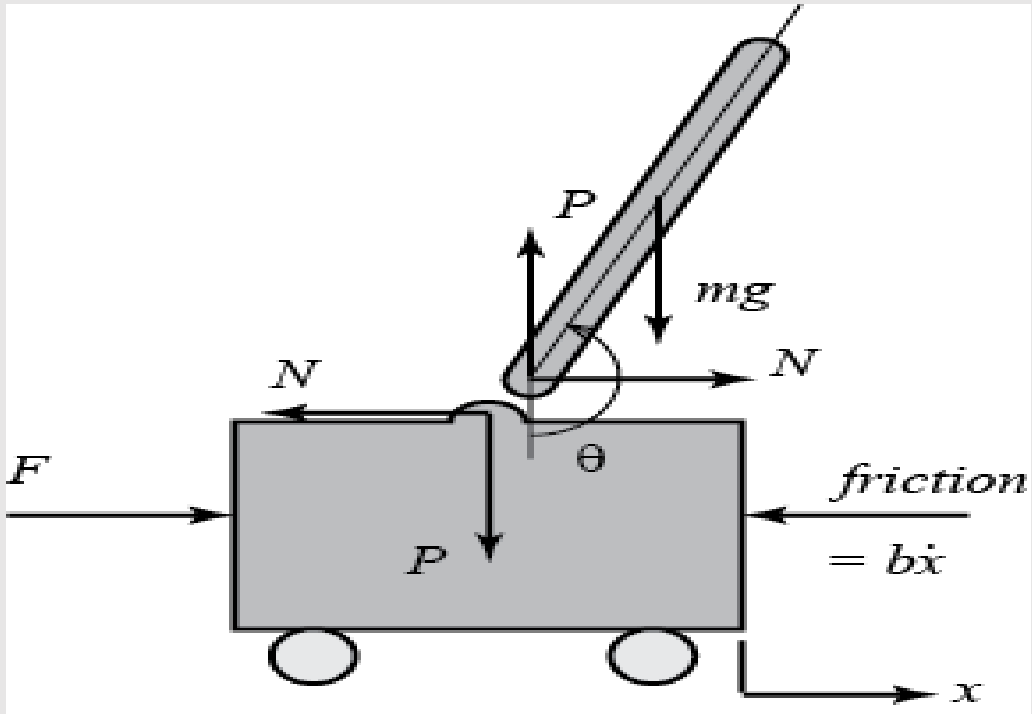
Hafıza: {

durum:	***kanepenin üstündeyim***
aksiyon:	***yere atlıyorum (%90 oranda rastgele)***
ödül:	***babam şeker yememe cezası verdi***
ulaşılan durum:	***ayağımı burktum ve bileğim acıyor***
oyun bitti mi:	***bitti*** ***}***

Çocuk bu aksiyonun bir ceza getirdiğini, ayrıca kendisini kötü bir duruma ulaştırdığını öğrendi ancak hala keşfetme arzusuna hakim olamıyor, ε yeterince azalana, belki 25'li yaşlarına kadar farklı şekillerde rastgele aksiyonlar denemeye devam ediyor. En sonunda kendisine koyulan hedefe (ödüle) göre hafızasındaki her bir bileşene birer Q değeri atayarak ya sorunsuz bir şekilde atlamayı, ya da atlamamayı öğreniyor.

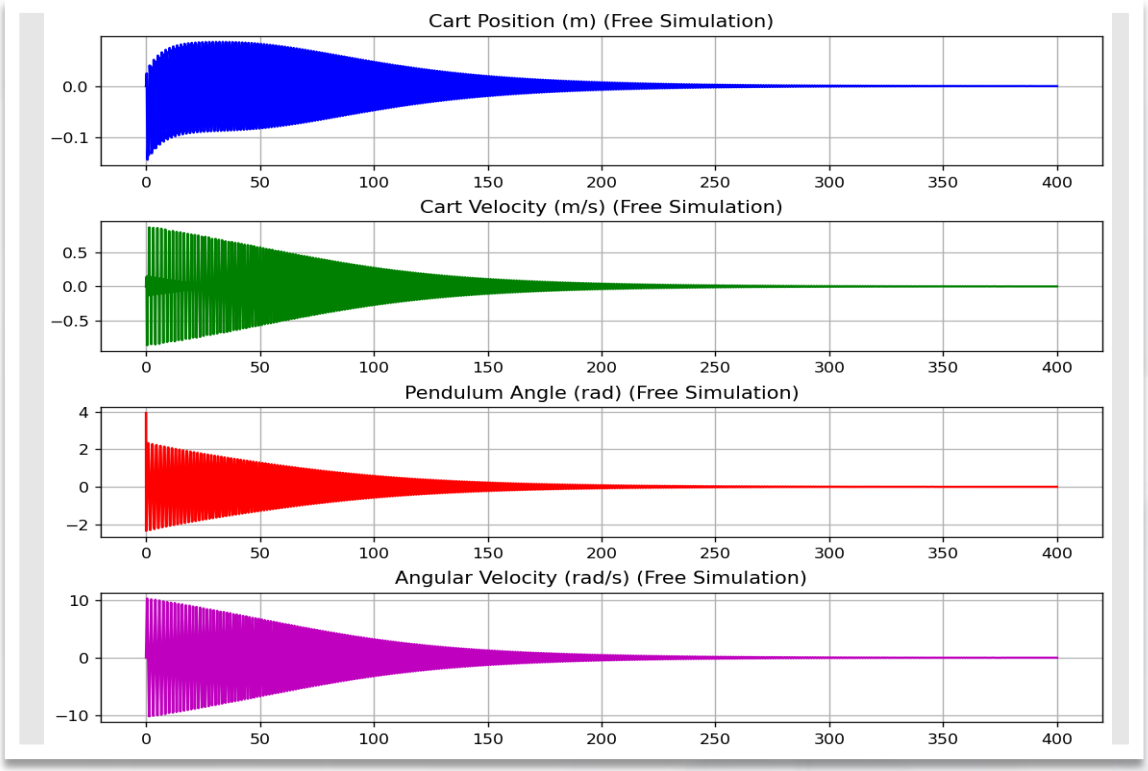
Çalışmanın Modeli

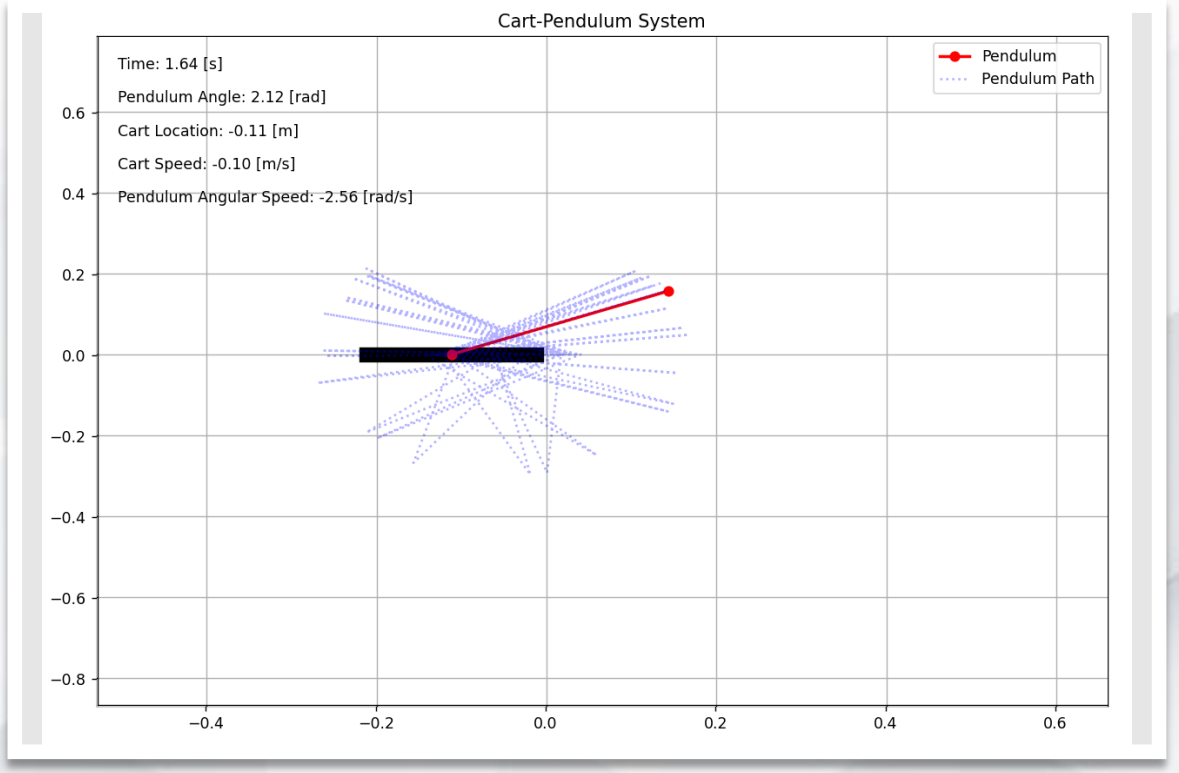
Temsil	Anlamı	Değeri
M	Araç ağırlığı	0.5 [kg]
m	Sarkaç ağırlığı	0.2 [kg]
b	Araç içi sürtünme katsayısı	0.1 [N/m/sec]
l	Sarkacın ağırlık merkezine olan mesafe	0.3 [m]
I	Sarkacın eylemsizlik momenti	0.006 [kg/m2]
F	Araca uygulanan kuvvet	F [N]
x	Aracın anlık pozisyonu	x [m]
θ	Sarkacın dik eksenle yaptığı açı	θ [°]
φ	Aracın anlık ivmesi	φ [m/s^2]



Örnek sarkaç sisteminin parametreleri

Çalışmanın Modeli



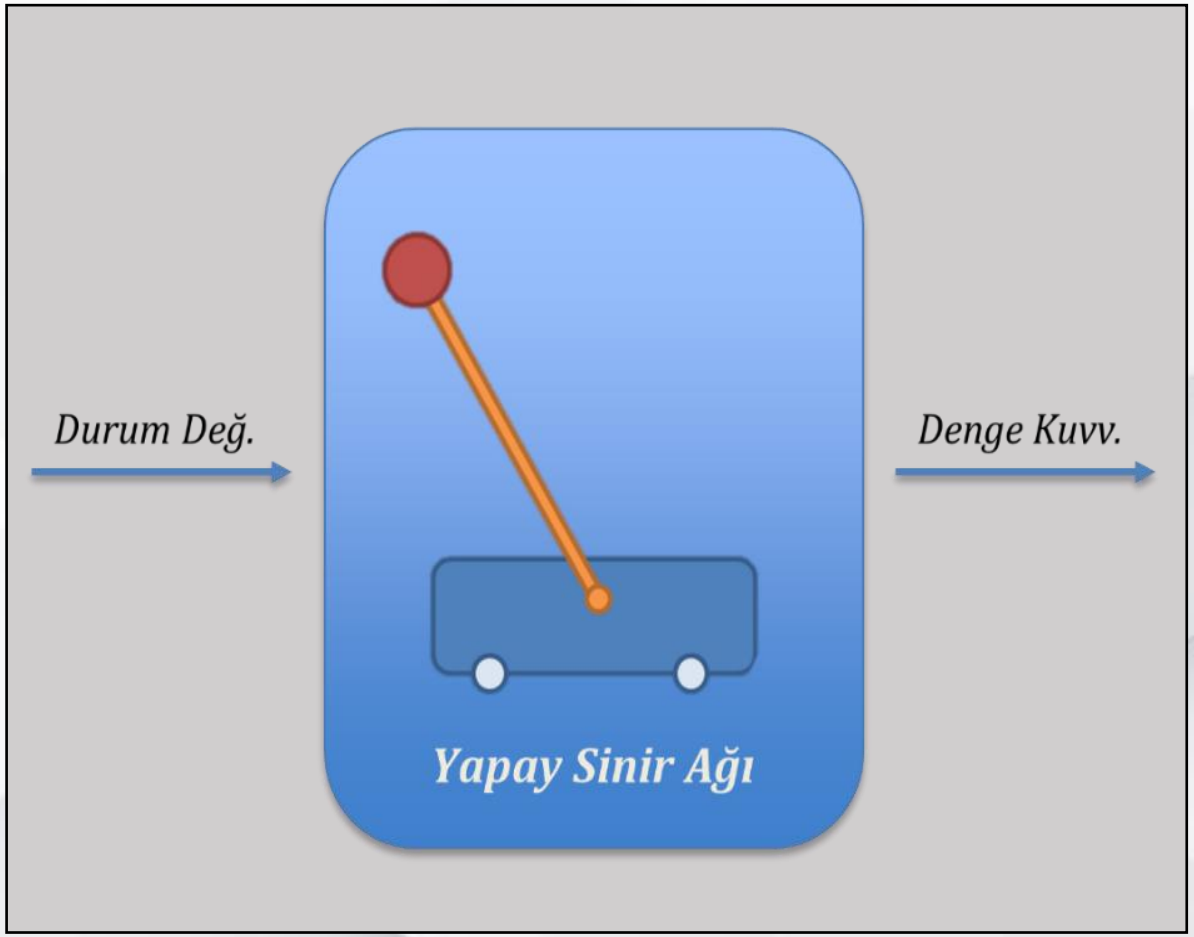


Ters sarkaç sisteminin serbest düşüş grafiği ve sarkacın görsel olarak izlenmesi için oluşturulan simülasyon ortamı

Çalışmanın Modeli: DQN

Çalışmanın ilk aşamasında ters sarkaç kendi başına dengede durmayı öğrendi. Bunun için öncelikle sadece sarkacı dengede tutmak için tasarlanmış bir YSA kullanıldı. Sistem giriş değerleri olarak sarkacın durum değişkenlerini aldı ve bir denge kuvveti üreterek sarkacın dengesini sağlamaya çalıştı.

Sonraki aşamalarda ise poisson dağılımına göre rastgele zamanlarda belirlenen rastgele kuvvetler adama bildirilmeden beklenmeyen bir bozucu olarak sisteme uygulandı ve adam buna rağmen dengeyi sağlamayı öğrendi.



Ters sarkaç kontrolü için tasarlanan DQN YSA'nın temsili modeli

DQN 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
Örneklem 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, Optimize Edici	MSE, Adam
Eğitim Bölüm Sayısı	2000 bölüm @ 200 adım
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	$-0.1 \times$ (Kalan adım) (En fazla -20 ceza)

DQN Yazılımsal Yapısı

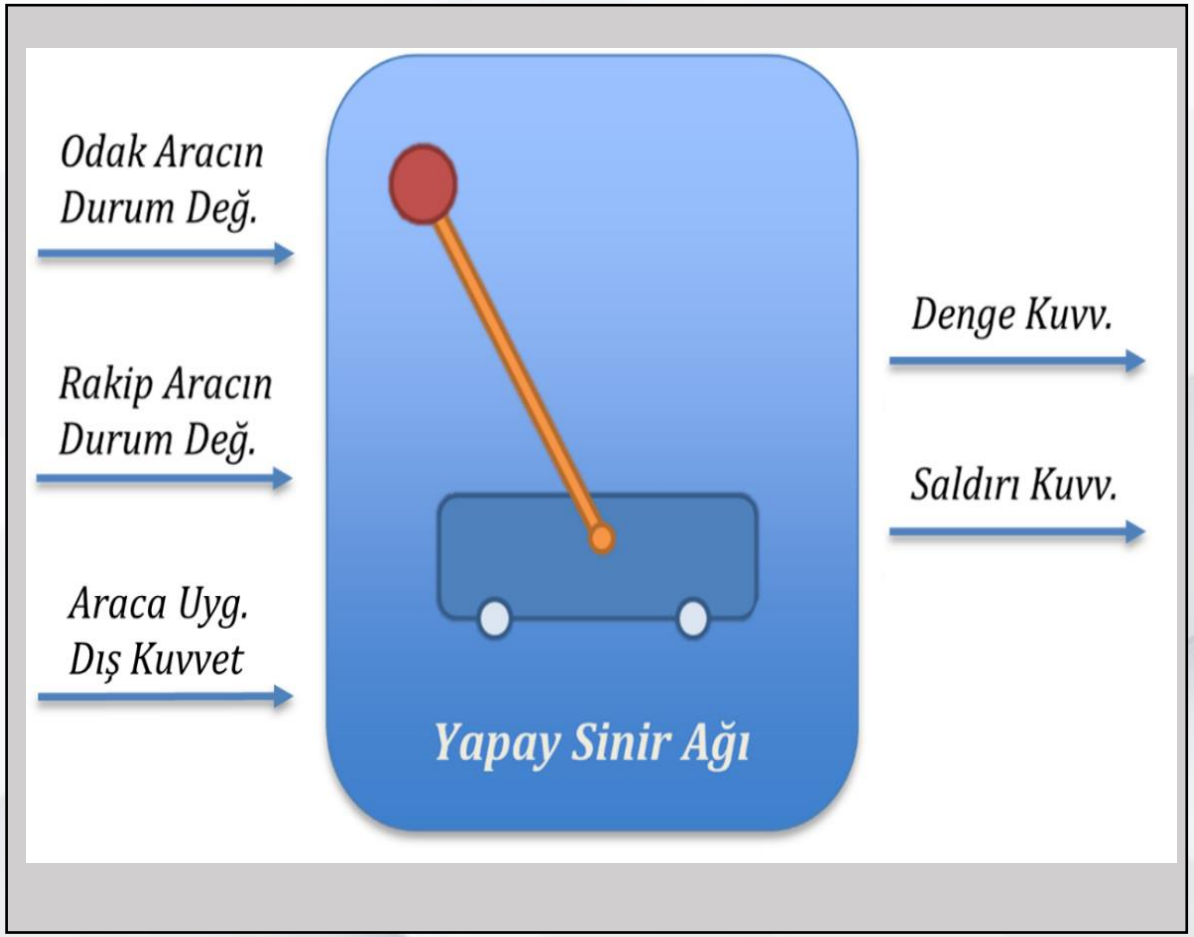
Modelin üzerine kurulduğu yazılımsal yapı, derin pekiştirmeli öğrenme için gerekli temel yapı taşlarını içerir.

- **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.

Çift Sarkaç Modeli

Çift sarkaç sisteminin kurulmasının ardından eğitimin ilk aşamasında bir sarkaç dengede durmayı öğrendi. Bunun için sarkacı dengelemek ve karşı tarafa bir saldırı uygulamak için tasarlanmış bir adam kısıtlanarak kullanıldı.

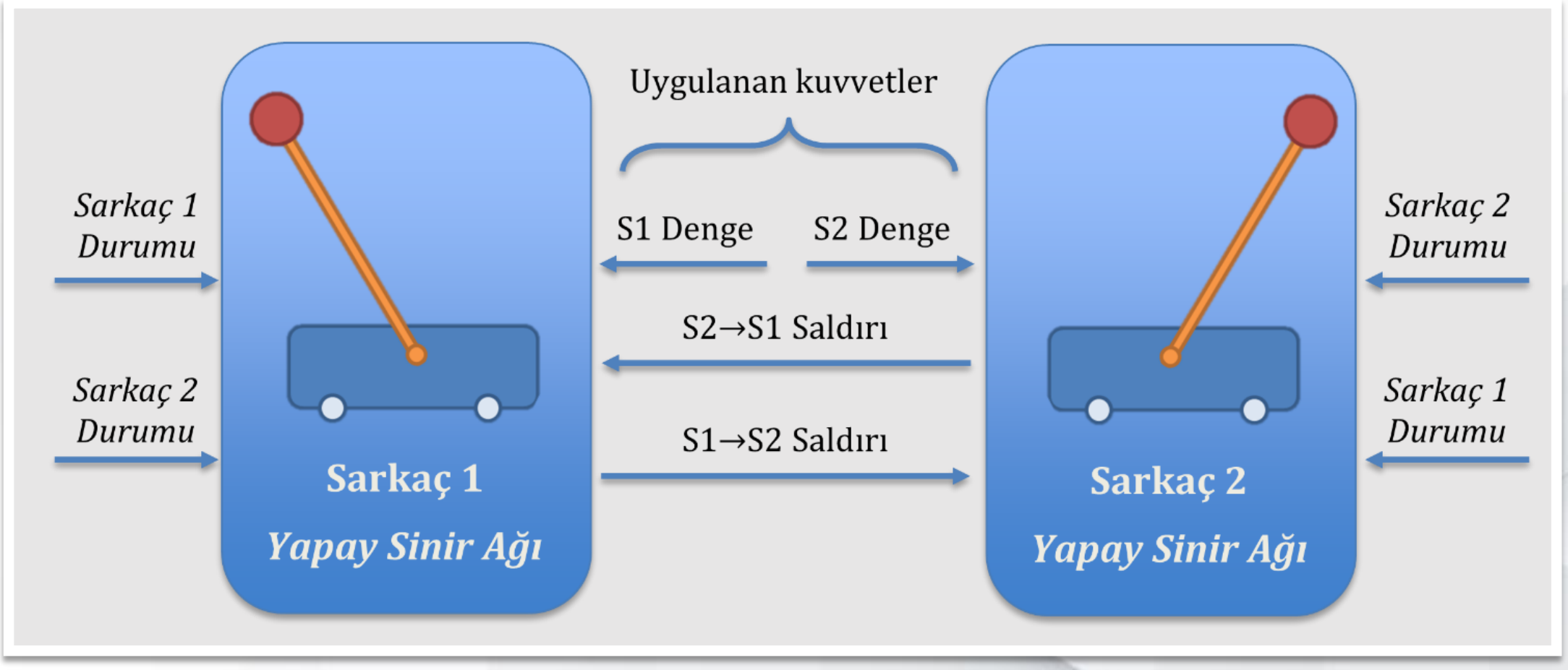
Sonraki aşamada bir adamın diğerini ezici bir üstünlükle bastırmasını engellemek kendini bir kavganın ortasında bulan adam, iki aracı da aynı anda kontrol etmeyi öğrendi. Kendi ve karşı durumunu, kendine uygulanan kuvveti gördü ***(büyük şans!)*** ve bir denge, bir saldırı kuvvetine karar verdi.



Ters sarkaç kontrolü için tasarlanan DQN YSA'nın temsili modeli

Geri's Game (1997)

Çift Sarkaç Modeli



Çift ters sarkacın aynı YSA'nın kontrolü altında birbiriyle etkileşimleri

DQN 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
Örneklem Boyutu	128
Hafıza Boyutu	100,000
Ara Katman Boyutları	512, 512, 256
Giriş / Çıkış Katman Boyutu	9 / 18
Model Güncelleme Sıklığı	Her adımda bir
Ödül Fonksiyonu	Aracın kendi ödülü – 0.3 × Karşı aracın ödülü
Standart Ödül Fonksiyonu	$-(0.1 \times s \times Q \times sT + 0.01 \times F_{denge}^2 + 0.005 \times F_{saldırı}^2)$

DQN Yazılımsal Yapısı

Yeni sistem önceki durumdan devam edebilme, ikili veya tekli eğitim ortam, dövüş ortamı ve poisson ile rastgele bozucu darbe uygulanması durumlarını açıp kapatabilme imkanıyla tasarlanmıştır. 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 eylem seçimi fonksiyonunda, giriş ve çıkış uzayı ana araç ve karşı araç için olmak üzere ortadan ikiye bölünecek şekilde yapılmıştır. 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.

Bulgular: Parametreler

Öğrenme oranı (α): 0.1, 0.01, 0.001 öğrenme oranlarını denedim ve eğitimin daha tutarlı olması amacıyla 0.001 değerinde karar kıldım.

İskonto faktörü (γ): 0.90 ve 0.95 oranlarını denedim, sarkaç dengesinde uzun vadeli hedefler daha önemli olduğu için 0.99 değerinde karar kıldım.

Keşif Oranı (ε) azalması: 0.70, 0.90, 0.99, 0.999 oranları çok hızlı azaldığı için tek sarkaç sisteminde 0.9999, çift sarkaç sisteminde de 0.99995 değerlerinde karar kıldım.

Örneklem uzayı (batch size): 32 ve 64 değerlerini denedikten sonra tek sarkaçta 64, çift sarkaçta 128 boyutunda karar kıldım.

Ödül fonksiyonu: sırasıyla yalnızca açıya, açı ve açısal hıza, tüm durumlara önem veren fonksiyonlar kullandıktan sonra en son aşamada tüm durumlara, denge ve saldırı kuvvetine ayrı ayrı katsayılarla önem veren ve erken bitirme cezası içeren karmaşık bir ödül fonksiyonu kullanarak adamın tüm hareketleri üzerinde tahakküm etmiş oldum. Kavga ortamında ise rakibin ödülünden de fayda sağlanan bir ortam oluşturdum.

Bulgular: Eğitim Süreci

Tek Sarkaç Sistemi

Serbest denge: *adamın aldığı ödüller 2500 bölüm civarında sıfıra yakınsamıştır.*

Rastgele darbelere karşı denge: *adamın aldığı ödüller 3500 bölüm civarında sıfıra yakınsamıştır. Eğitim sıfırdan başlamış olup adamın kendisine etkiyen dış kuvvetleri bilmemesi eğitimi uzatan ve ödüller sıfıra yakınsadığı süreçte bile ödülün sık sık dalgalanmasını sağlayan önemli bir etmen olmuştur.*

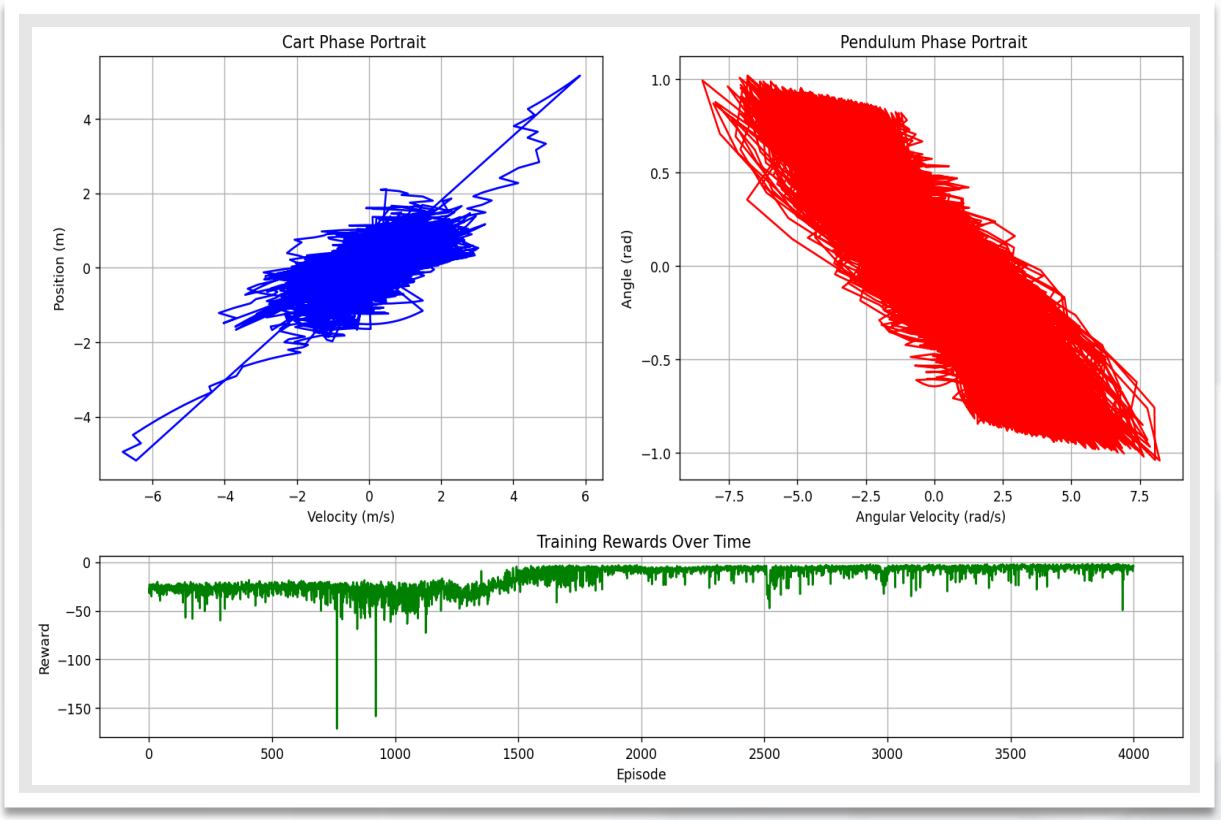
Çift Sarkaç Sistemi:

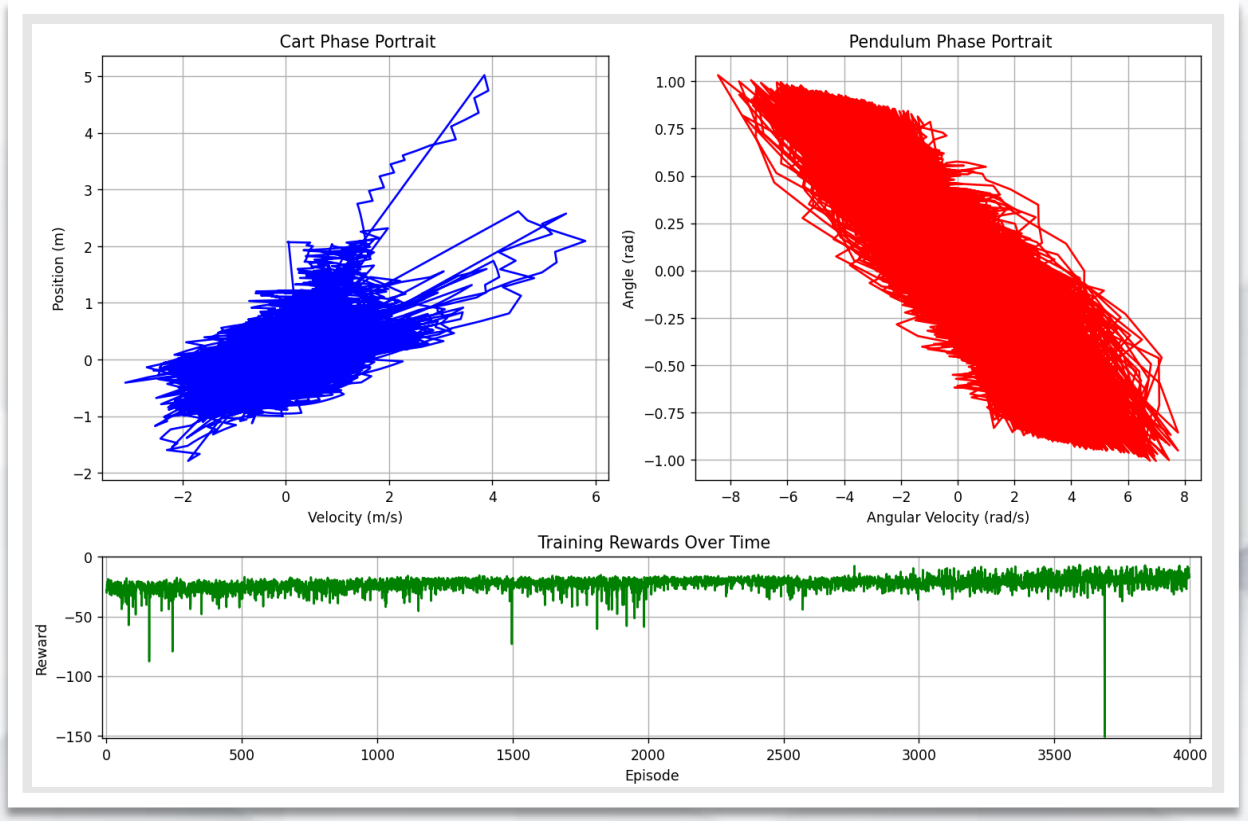
Serbest denge: *adamın aldığı ödüller 2600 bölüm civarında sıfıra yakınsamış ve maksimum adım sayısı sarkaç devrilmeden tamamlanmaya başlanmıştır.*

Rastgele darbelere karşı denge: *adamın ε değeri sıfırlanarak eğitim serbest dengenin öğrenildiği durumdan kaldığı yerden devam ettirilmiş ve ardından alınan ödüller 1500 bölüm civarında sıfıra yakınsamıştır.*

Bu değerlerin tecrübe edilmesinin ve adamların sorunsuz bir şekilde uzun zamanlar dengede kaldığının anlaşılmasının ardından kavga modu çalıştırılmıştır.

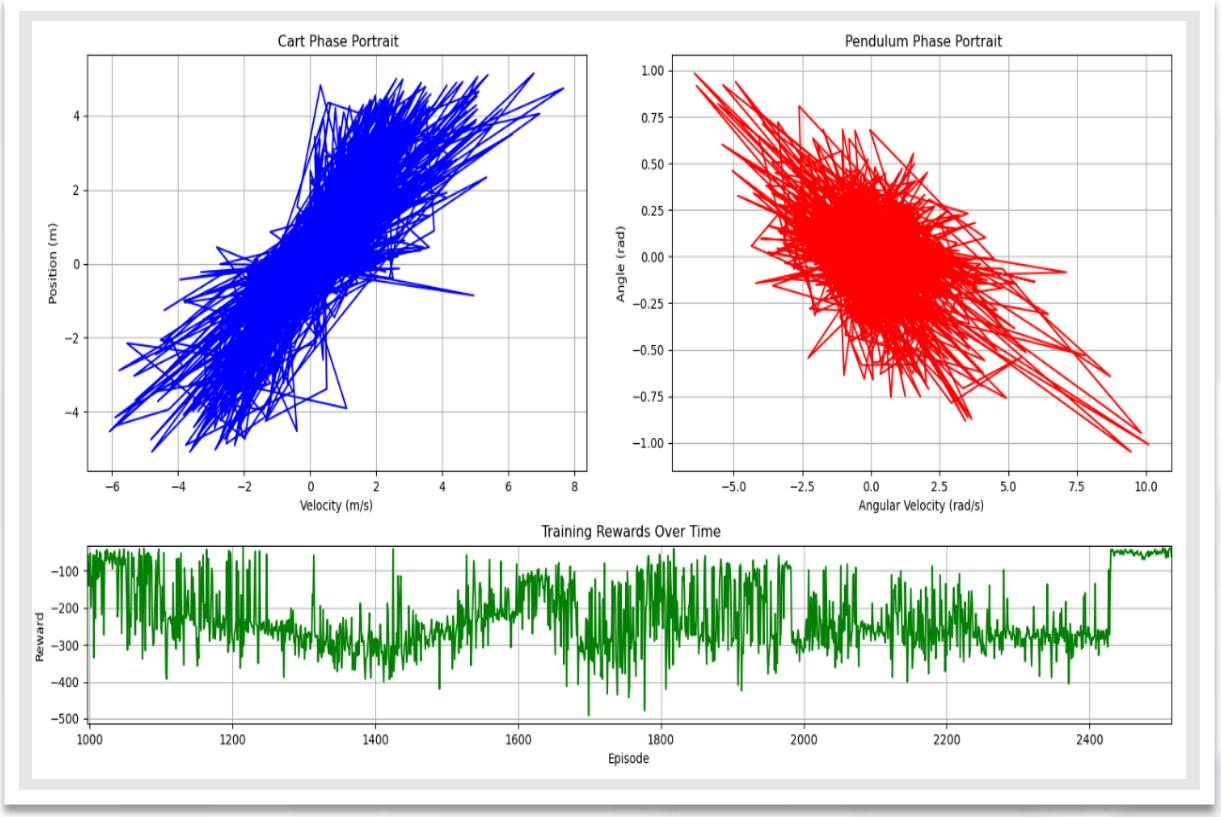
Tek Sarkaç Eğitim Süreci

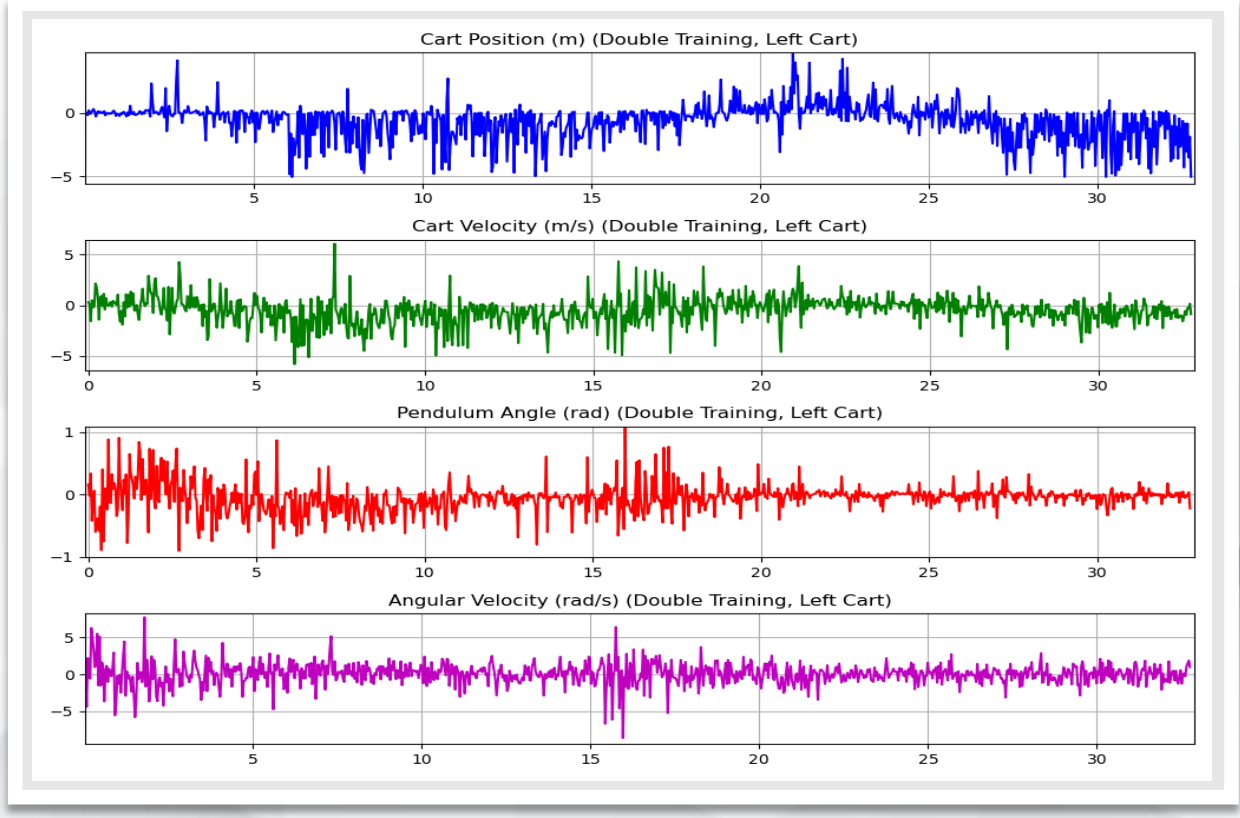




Ters sarkacın serbest (solda) ve bozucu etkiye maruz kalan (sağda) eğitim esnasında konum ve açı faz portreleri, ödül değişimi

Çift Sarkacın Tek Eğitimi





Çift sarkacın müsabakası amacıyla tasarlanan sistemin tek sarkaçla eğitiminin faz portreleri (solda) ve bozucu etki altındaki sistemin durum grafiği (sağda)

Çift Sarkaç Kavga Modu

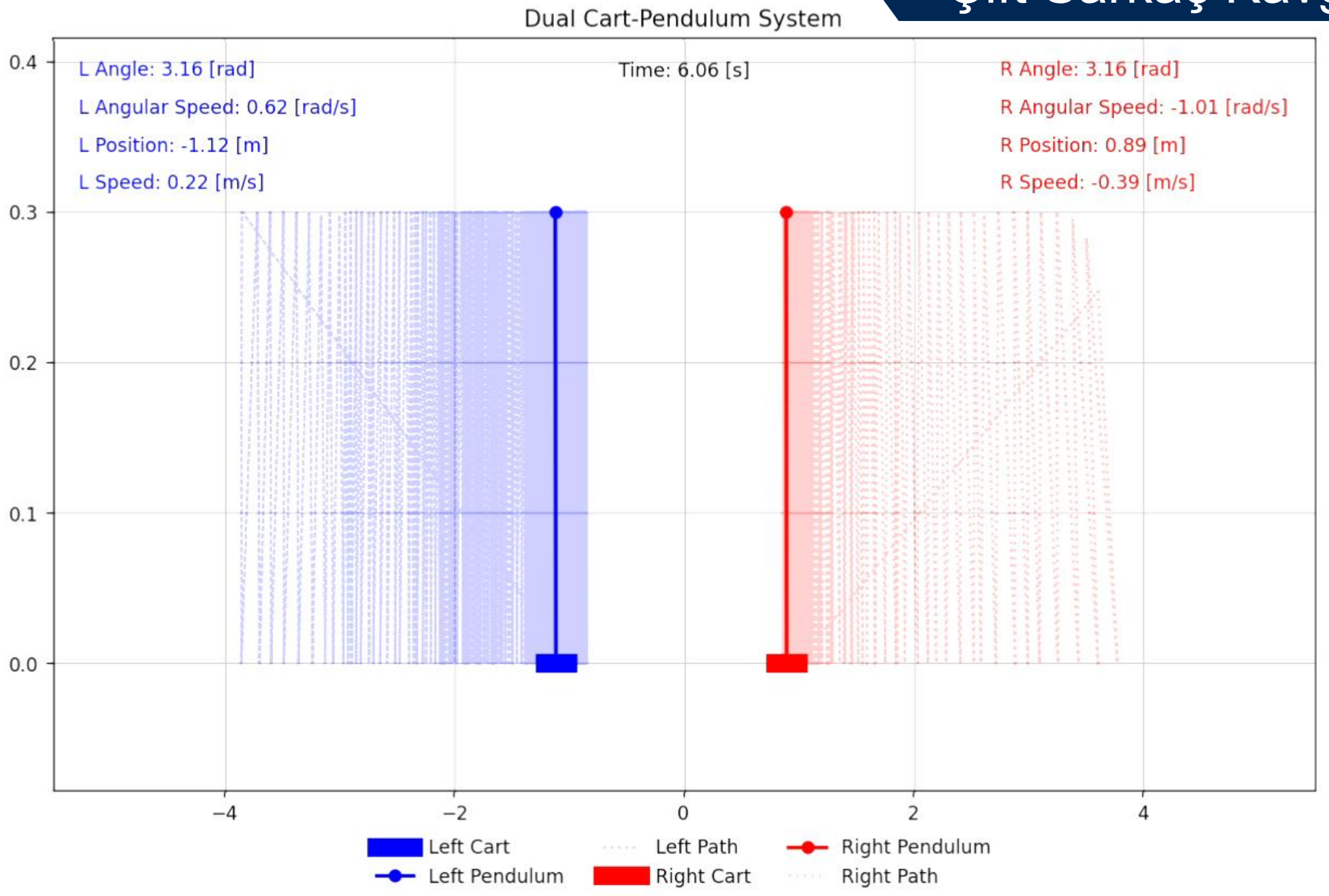
Denge kontrolünün başarıyla öğrenilmesinin ardından, iki araçlı sistem aynı model ile kontrol edilerek müsabaka başlatılmıştır. Uygulanan kuvvetin yıkıcı değil stratejik olması amacıyla saldırı kuvveti, denge kuvvetinin %15'iyle sınırlandırılmıştır.

İlk 900 bölümde keşif oranı minimuma indikten sonra toplamda 1300 bölüm süren bir eğitim süreci tamamlanmıştır. Araçlar maksimum 200 adımdan oluşan bölümleri çoğunlukla yenişemeden ve yüksek puanlarla tamamlamaya başlamıştır.

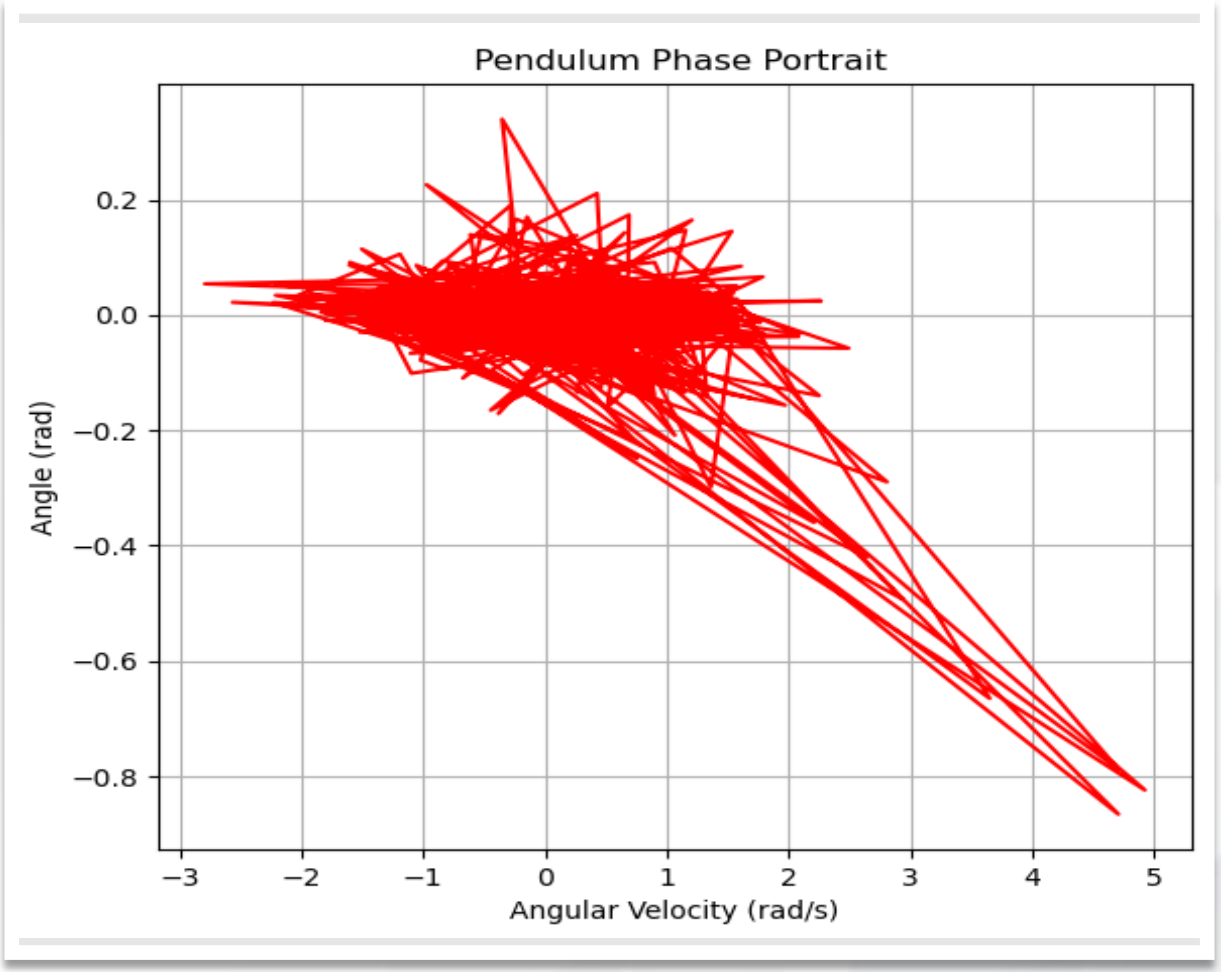
Son noktada bir test modu oluşturulmuş, 300 bölüm boyunca maksimum adım 1000 ve keşif değeri 0.00'a ayarlanarak adamın tamamen kendi tecrübelerini kullanarak uzun süren bir müsabaka oynaması sağlanmıştır. Adam bu müsabakaları ortalama 320 adım ve en uzun bölümde 700 adım oynayarak uzun bir seriyi başarıyla tamamlamıştır.

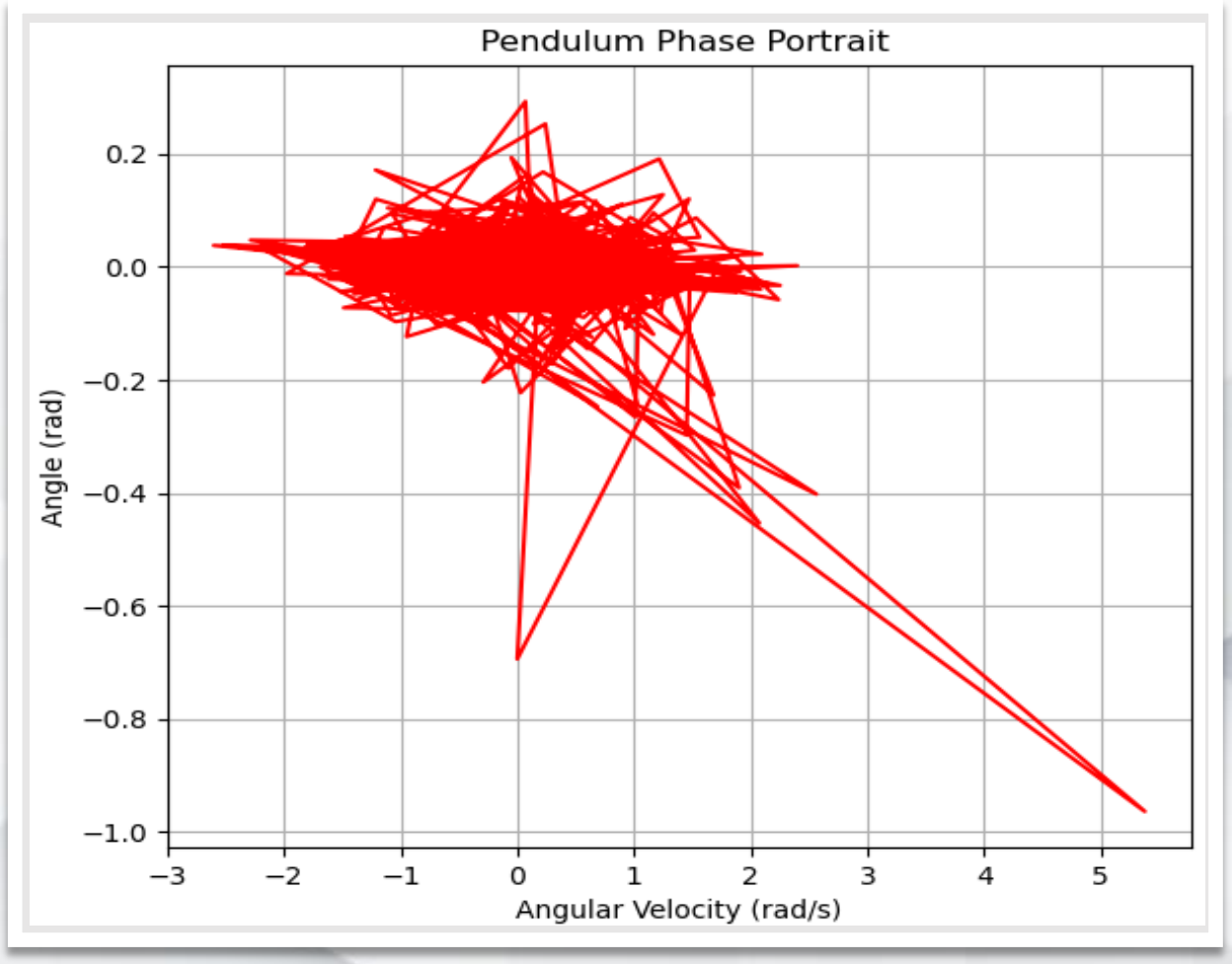
Bölümü erken kaybedip devrilen aracın ödülü erken bitirme cezasıyla ve karşının ödülü de odak aracın ödülünden çıkarıldığı için dramatik olarak yenen araçta 0, yenilen araçta -300 seviyesinde olmaktadır. Adımların tamamı hiçbir zaman oynanamadığı için berabere biten bir bölüm olmamıştır.

Çift Sarkaç Kavga Modu



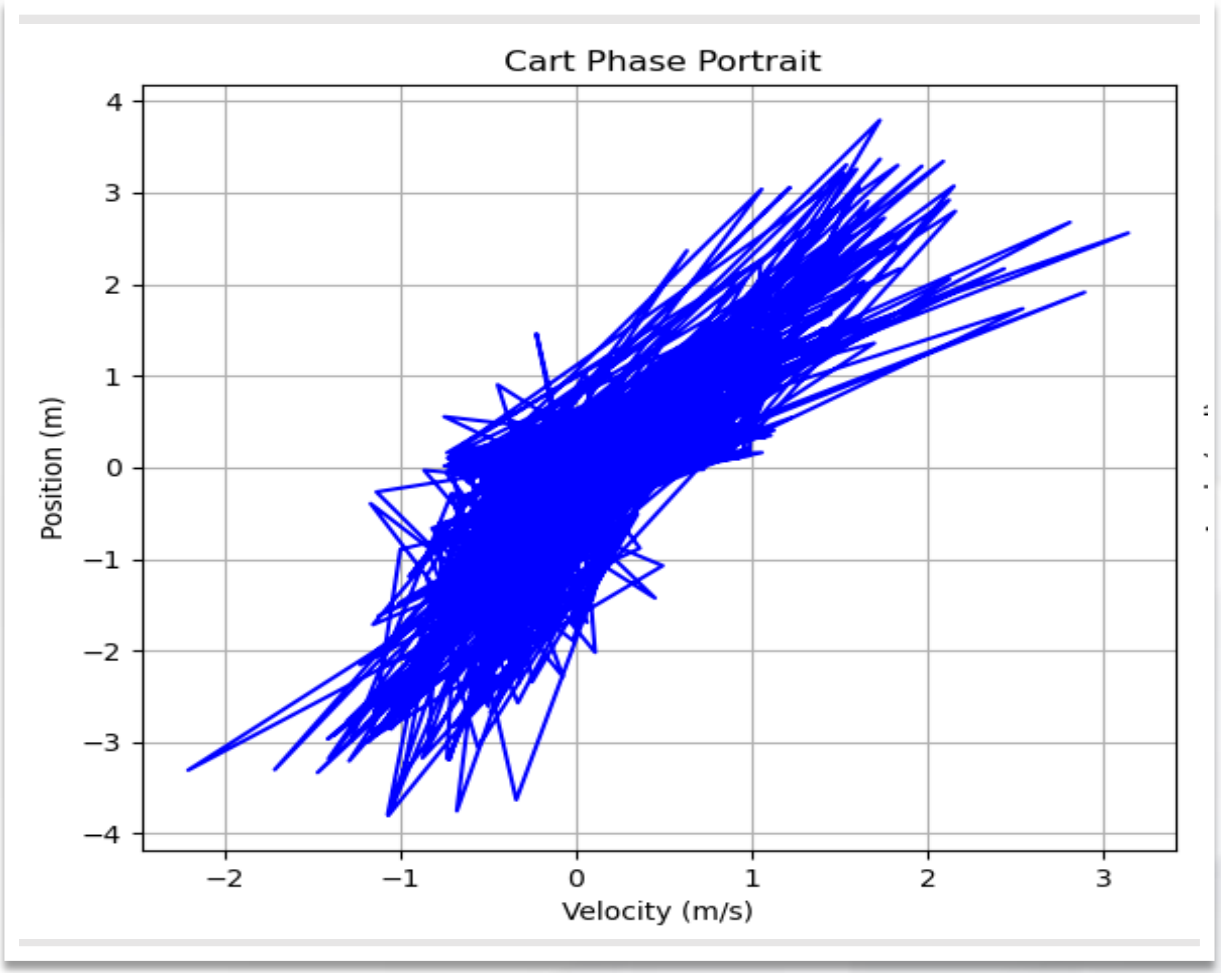
Çift Sarkaç Kavga Modu

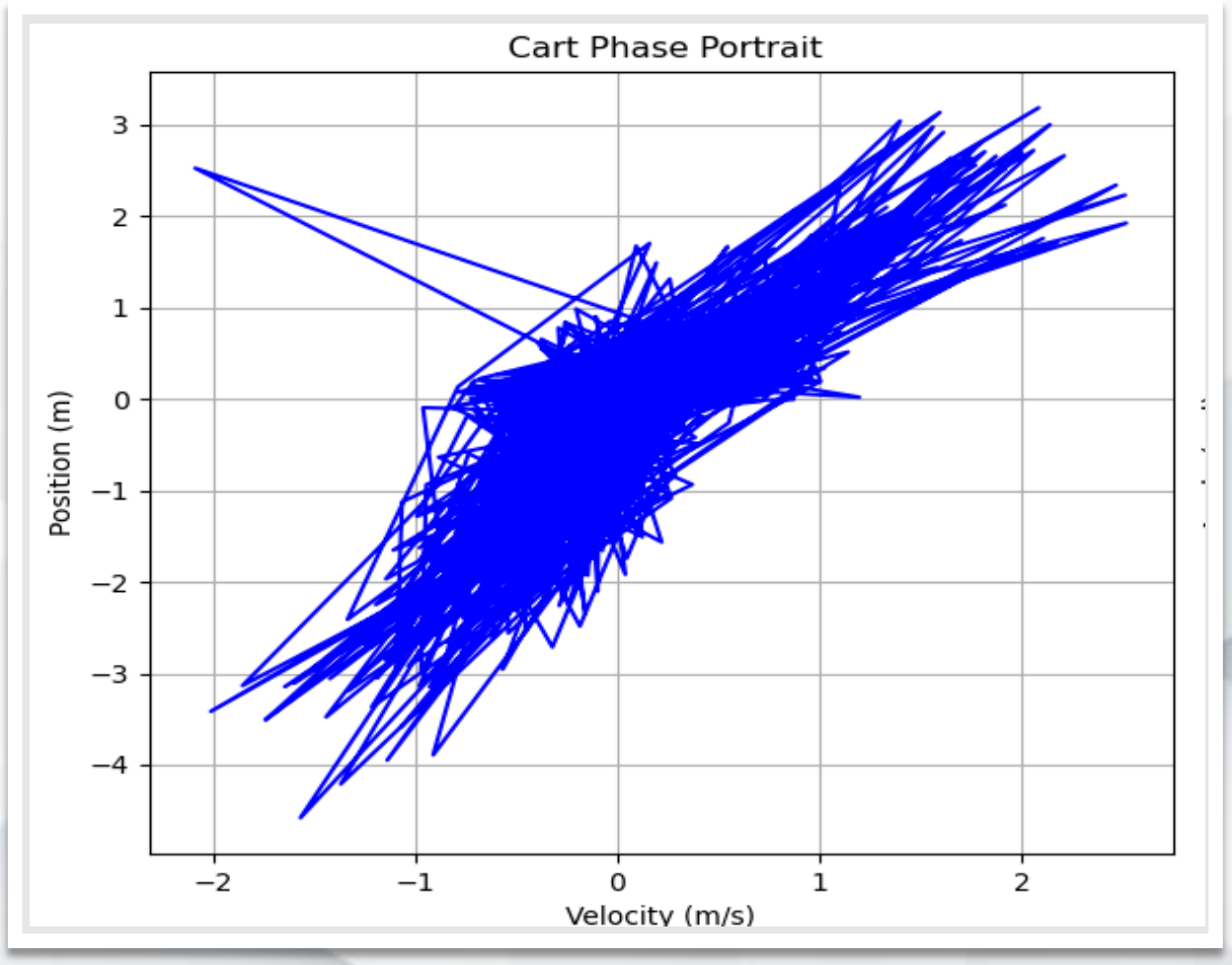




Çift sarkaç müsabakasında açısal konum faz portreleri sol ve sağ araç

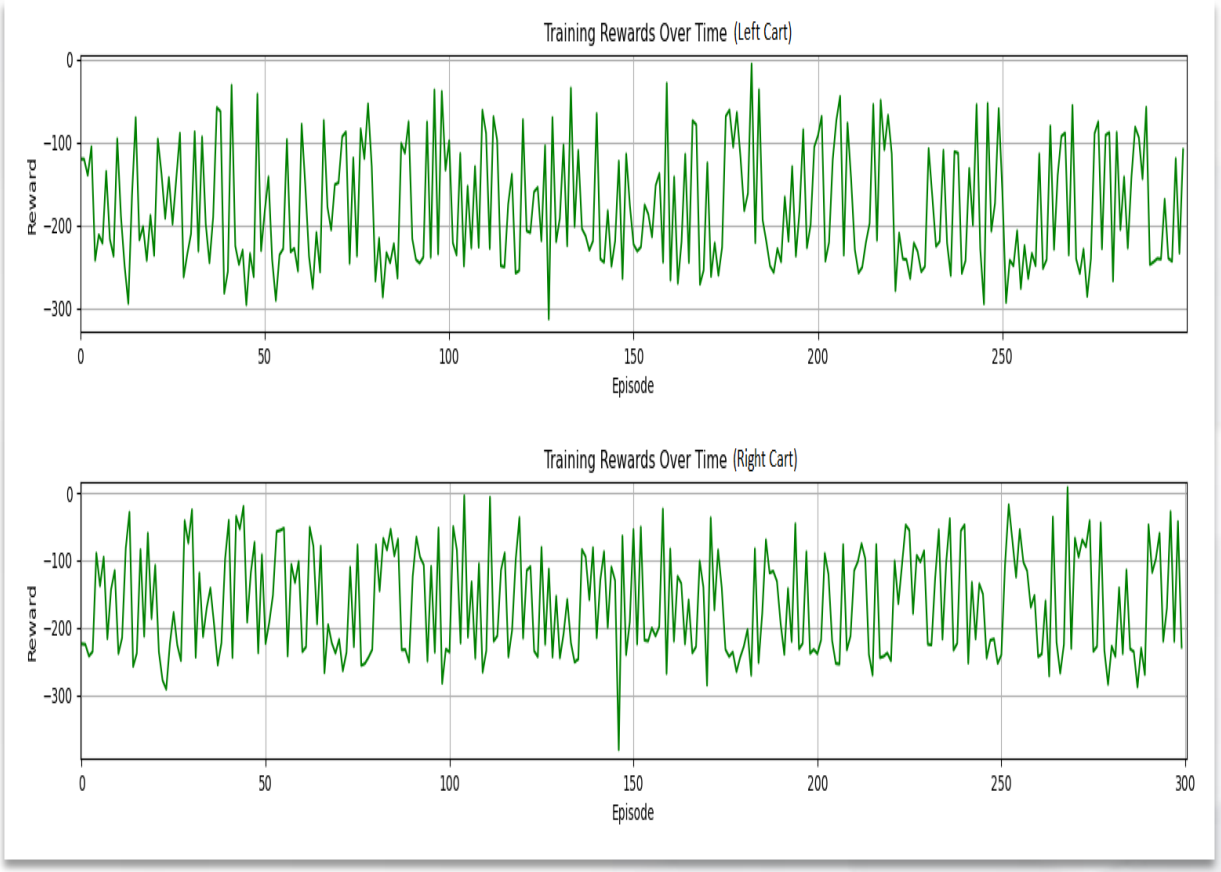
Çift Sarkaç Kavga Modu

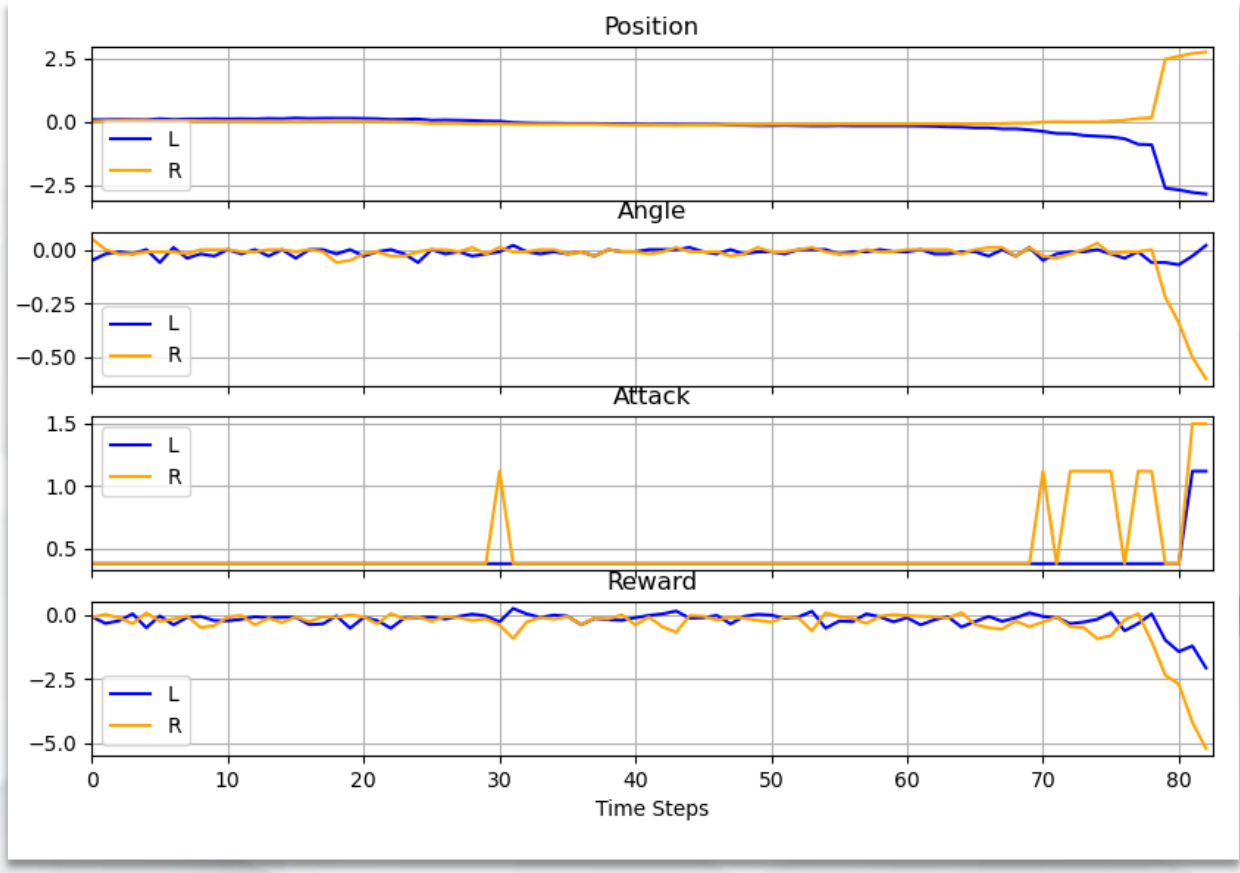




Çift sarkaç müsabakasında konum faz portreleri sol ve sağ araç

Çift Sarkaç Kavga Modu





Çift sarkaç müsabakasında ödüllerin karşılıklı değişimi ve en uzun süren 700 adımlık bölümde araçların karşılıklı durumları

Kaynakça

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

SORULARINIZ VAR MIYDI?

Teşekkürler

30.01.2025

Muhammet Işık

İTÜ

MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANILARAK BİR DÖVÜŞEN ROBOTUN EĞİTİLMESİ

İZLEDİĞİNİZ İÇİN TEŞEKKÜRLER!

Teşekkürler

30.01.2025

Muhammet Işık