Döner Ters Sarkaç Sisteminin Pekiştirmeli Öğrenme Algoritmaları ile Kontrolü

Nevrez İmamoğlu[1], *Aydın Eresen*[1], *Mehmet Önder Efe*[1]

[1]Elektrik ve Elektronik Mühendisliği Bölümü
TOBB Ekonomi ve Teknoloji Üniversitesi, Ankara
{nimamoglu, aeresen, onderefe}@etu.edu.tr

Özet

Yapay zekâ uygulamalarının en önemli uygulama alanlarından birisi otomatik kontroldür. Bu bildiride tek girişli ve çok çıkışlı bir yapıya sahip olan bir döner ters sarkaç sisteminin dinamik modeli kullanılarak pekiştirmeli öğrenme algoritmaları gerçeklenmiştir. Kontrol için Değer Yineleme[1], *Q*-Öğrenme[2], Sarsa, ve Sarsa (λ) isimli pekiştirmeli öğrenme[3] algoritmaları denenmiştir. Değer Yineleme algoritması modele ihtiyaç duyan bir yöntemdir, gerçeklenen diğer yöntemler ise dinamik modele ihtiyaç duyulmadan sistemin cevabına göre öğrenen yapılardır. Değer Yineleme algoritmasının eldeki model parametrelerine göre benzetim ortamında daha verimli sonuçlar verdiği gözlemlenmiştir.

Abstract

One of the most important Artificial Intelligence Applications is Automatic Control. In this paper, several Reinforcement Learning methods are realized by using a dynamic model of rotary inverted pendulum which is a Single Input Multiple Output system. Value iteration, *Q*-Learning, Sarsa, ve Sarsa (λ) Reinforcement Learning algorithms are applied for controlling of our system. Value Iteration is a technique, needing a dynamic model of system, on the other hand applied other techniques are using system responses, needn't dynamic model. Experimental results show that Value Iteration algorithm is more efficient than other techniques.

1. Giriş

Akıllı sistem kuramının temel öğrenme biçimlerinden ikisi öğreticili ve öğreticisiz öğrenme yöntemleridir. Her ne kadar öğreticili öğrenme yaygın olarak kullanılıyor olsa da bir çocuğun bisiklet sürmeyi öğrenmesi gibi doğal bazı süreçlerde öğrenme olgusu öğreticisiz öğrenme şeklindedir ve istenen değer için hâlihazırda bir bilgi yoktur. Geçmiş deneyimlere göre daha iyi veya daha kötü diye nitelendirilebilecek davranışların kusursuzlaştırılmasıyla istenen değerlerin elde edilmesi sürecinde ödül/ceza stratejisi uygulayan pekiştirmeli öğrenme yapıları pek çok problemin çözümüne doğal yöntemlerden esinlenen çözümler önerir. Bunlardan biri de ters sarkaç problemidir. Ters sarkaç sistemlerinde yaygın olarak üzerinde çalışılan sistem kremayer dişlisi üzerinde koşan ve ters sarkaç taşıyan bir arabadır[4] [1-3]. Maravall vd. [1] Bulanık kontrol ve Oransal-Türevsel (PD) denetim yöntemlerini birleştiren melez bir yapı ortaya koymuş ve altı alt bölge üzerinde PD kontrolörü kullanarak istenen kapalı çevrim başarımını elde etmiştir. Li vd. [2] Oransal-İntegral-Türevsel (PID) parametrelerini gerçek zamanlı olarak güncellemek yerine, bu parametreleri bir Yapay Sinir Ağının (YSA) gizli katmanındaki birer nöron olarak kabul edip eğitim sonucundaki değerleri PID kontrolörde kullanmışlardır. Yu vd. [3] altı basamaklı hareket stratejisi[5] ve kısmi geri beslemeli doğrusallaştırma yöntemleri ile ters sarkaçlı arabanın kontrolünü yapmıştır. Yapay zekâ algoritmalarının başarımlarının daha iyi irdelenebilmesi ve karşılaştırılabilmesi için bu yaklaşımları karmaşık mühendislik problemlerinde denenmesi de araştırıcıların sıkça başvurduğu bir yol olmuştur [4-7]. Wu ve Pugh [5], pekiştirmeli öğrenme tekniklerini stokastik kontrol problemleri üzerinde uygulamışlardır. Çalışmalarında öğrenen otomata tabanlı bir kontrolörü kesinsizlik içeren dinamik sistemler üzerinde denemişlerdir. Buskey vd. [6] gerçek zamanlı uyarlamalı kontrol sistemleri olarak tek katmanlı YSA, çok katmanlı YSA ve bulanık ilişkilendirmeli hafızalar yöntemlerini [6] karşılaştırmışlardır. Waslander vd. [7], çoklu aracı dönerkanat deney düzeneği üzerinde integral kayan kipli kontrolör ve model tabanlı pekiştirmeli öğrenmeye dayalı kontrolör tasarımlarını karşılaştırmalarını sunmuşlardır. Doğrusal olmayan bir yapıya sahip olan döner ters sarkaç sistemi de geri beslemeli kontrol sistemlerinde sıkça kullanılan bir denektaşı problemi olmuştur [8-9]. Khanesar vd. [8] döner ters sarkacın kayan kipli kontrolünün benzetimini yapmış, Sukontanakarn ve Parnichkun [9] ise PD ve doğrusal kuadratik denetleyici kullanarak döner ters sarkaç sistemi için kendinden doğrulmalı bir kontrol sistemi önermişlerdir. Bu çalışmada Şekil 1'de temsili resmi verilen sisteme ait dinamik model kullanılmıştır.

Bu çalışmada modele bağlı öğrenen Değer Yineleme algoritması ve modelden bağımsız çalışan *Q*-Öğrenme, Sarsa, Sarsa (λ) algoritmalarıyla belirli durumlarda en uygun hareketi öğrenmesi sağlanarak oluşturulan durum ve hareket tablosundan seçilecek kontrol sinyalleri ile kontrol yapılmaya çalışılmıştır. Bahsedilen yöntem pekiştirmeli öğrenme olarak literatürde sıkça kullanılmaktadır [5-7, 11-13].

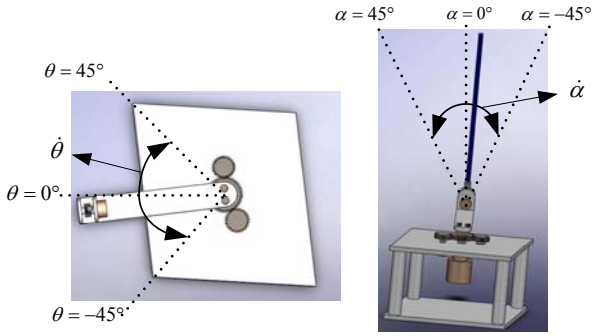
[1] İng. *Value-Iteration*
[2] İng. *Q-Learning*
[3] İng. *Reinforcement Learning*
[4] İng. *Cart Pole*
[5] İng. *Six step motion strategy*
[6] İng. *Fuzzy Associative Memories*

Şekil 1: Döner ters sarkaç sistemi

Bu çalışmada model tabanlı Değer Yineleme ve modelden bağımsız *Q*-Öğrenme, Sarsa, ve Sarsa (λ) pekiştirmeli öğrenme algoritmaları gerçeklenmiş benzetim ortamında elde edilen sonuçlar tartışılmıştır 2. bölümde döner ters sarkaç sisteminin dinamik modeli verilmiş, 3. bölümde pekiştirmeli öğrenme ve belirtilen algoritmalar özetlenmiştir. 4. bölümde benzetimler tartışılmış, son kısımda ise genel sonuçlar verilmiştir.

2. Döner Ters Sarkaç Sistemi

Döner ters sarkaç sistemi ve ilgili durum değişkenleri Şekil 2'de gösterilmektedir. Döner ters sarkaç sisteminin kontrolünde amaç α açısının sıfır derece olması ve θ açısının da sabit bir değere yakınsamasıdır.



Şekil 2: Sistemin açısal pozisyon durum değişkenleri

Tablo 1. Sistemin fiziki parametreleri

L	Sarkacın kütle merkezi uzaklığı	0.1675 m
m	Sarkaç kolunun kütlesi	0.1250 kg
r	Döner kol uzunluğu	0.2150 m
J_{eq}	Sarkacın ağırlık mekezinin eşdeğer ataleti	0.0036 kg/m^2

Sarkacın potansiyel enerjisi (V) ve kinetik enerjisi (T) sırasıyla (1) ve (2) denklemleriyle verilebilir, [14],

$$V = mgL\cos(\alpha) \tag{1}$$

$$T = \frac{1}{2}J_{eq}\dot{\theta}^2 + \frac{1}{2}m\left(r\dot{\theta} - L\dot{\alpha}\cos(\alpha)\right)^2 + \frac{1}{2}m\left(-L\dot{\alpha}\sin(\alpha)\right)^2 + \frac{2}{3}mL^2\dot{\alpha}^2 \tag{2}$$

Lagrangian ise aşağıdaki gibi yazılırsa ve sistem dinamiği α = 0 etrafında denklem doğrusallaştırılacak olursa durum uzayı denklemleri (4) denkleminde verilen biçimde elde edilir [14],

$$L = T - V = \frac{1}{2}J_{eq}\dot{\theta}^2 + \frac{2}{3}mL^2\dot{\alpha}^2 - mLr\cos(\alpha) + \frac{1}{2}mr^2\dot{\theta}^2 - mgL\cos(\alpha) \tag{3}$$

$$\begin{bmatrix}\dot{\theta}\\ \dot{\alpha}\\ \ddot{\theta}\\ \ddot{\alpha}\end{bmatrix} = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & \frac{bd}{E} & \frac{-cG}{E} & 0\\ 0 & \frac{ad}{E} & \frac{-bG}{E} & 0\end{bmatrix}\begin{bmatrix}\theta\\ \alpha\\ \dot{\theta}\\ \dot{\alpha}\end{bmatrix} + \begin{bmatrix}0\\ 0\\ c\frac{\eta_m\eta_g K_t K_g}{R_m E}\\ b\frac{\eta_m\eta_g K_t K_g}{R_m E}\end{bmatrix}V_m \tag{4}$$

Denklem (4)'te kullanılan η_m motor verimini, η_g dişli kutusu verimini, K_t motor-tork sabitini, K_g motor-dişli sabitini, K_m elektromotor kuvvet sabitini, R_m armatür direncini ifade etmektedir. Burada ilgili değişkenler aşağıdaki gibi tanımlanmakta, fiziki parametreler ise Tablo 1'de verilmektedir.

$$a = J_{eq} + mr^2 \tag{5}$$

$$b = mLr \tag{6}$$

$$c = \frac{4}{3}mL^2 \tag{7}$$

$$d = mgL \tag{8}$$

$$E = ac - b^2 \tag{9}$$

$$G = \frac{\eta_m\eta_g K_t K_m K_g^2 + B_{eq}R_m}{R_m} \tag{10}$$

Sayısal değerleri kullanılmasıyla (11) denkleminde verilen dorusal durum uzayı sistemi elde edilecektir, [14].

$$\begin{bmatrix}\dot{\theta}\\ \dot{\alpha}\\ \ddot{\theta}\\ \ddot{\alpha}\end{bmatrix} = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 39.32 & -14.52 & 0\\ 0 & 81.78 & -13.98 & 0\end{bmatrix}\begin{bmatrix}\theta\\ \alpha\\ \dot{\theta}\\ \dot{\alpha}\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 25.54\\ 24.59\end{bmatrix}V_m \tag{11}$$

3. Pekiştirmeli Öğrenme Algoritmaları

Pekiştirmeli öğrenme yaklaşımlarının temel felsefesi geçekleşen bir olgunun ödüllendirilmesi veya cezalandırılmasını bir mantık içerisinde düzenleyerek istenen bir sonucun ortaya çıkmasını sağlamaktır. Durum vektörünün olası değerleri ve kontrol sinyalinin olası değerleri için oluşturulan tabloda her bir kombinasyon için belirlenen ödül değerleri tablolanır, ve istenen davranışı ortaya çıkaracak

kontrol sinyalinin üretilmesi en yüksek ödül değerine sahip durumlar içerisinden seçilir [4, 15-16]. Bu öğrenme yöntemi hem modele bağlı hem de modelden bağımsız olarak gerçekleştirilebilmektedir. Bu bölümde çalışmamızda kullanılan pekiştirmeli öğrenme yöntemleri anlatılacaktır.

3.1. Değer Yineleme Algoritması

Sistem dinamiğinin ve model parametrelerinin bilindiği durumlarda model tabanlı öğrenme yöntemleri yaygınca kullanılmaktadır. Değer Yineleme algoritması da model tabanlı bir yöntem olarak bilinmektedir [4, 12-13]. Algoritmanın amacı olası tüm hareketler için belirli bir durumdaki en uygun hareketin sistem dinamiği de göz önünde bulundurularak belirlenmesidir. Alpaydın [12] tarafından gösterilen algoritma, buradaki uygulamaya yönelik olarak aşağıdaki gibi uyarlanmıştır.

$V(s)$ değerini keyfi olarak tanımla
Tekrarla
  Tüm $s \in S$ durumları için
    Tüm $a \in A$ hareketleri için
      $Q(s,a) \leftarrow E[r|s,a] + \gamma V(s')$
    $V(s) \leftarrow \text{maks}_a\, Q(s,a)$
$V(s)$ değişimi belirlenen eşik değerinden küçük olana kadar

Yukarıdaki algoritmada s değişkeni durum, S değişkeni önceden belirlenen durum uzayıdır. a hareket, A önceden belirlenen hareket uzayı, $Q(s,a)$ s durumu ve a hareketine karşılık gelen ödül değeridir. $V(s)$ ise $Q(s,a)$'nın en büyük olduğu değer olup $V(s')$ uygulanan hareketle gelinmesi beklenen durum için elde edilen en yüksek ödül değeridir ve γ da etkinlik değeridir.

Algoritmada da görüldüğü üzere, bir durumda A hareket uzayındaki bütün hareketler için ödül değeri hesaplanır ve en uygun harekete göre $V(s)$ değeri güncellenir. $V(s)$ değerindeki değişim miktarı belirlenen değere yakınsadığında, bütün durumlar için en uygun hareketler belirlenmiş olur.

3.2. *Q*-Öğrenme Algoritması

Sistem modelinin bilinmediği durumda, sisteme verilen hareket değerleri için elde edilen sonuçlara bakılarak ödül verilmektedir ve $Q(s,a)$ matrisi oluşturulmaktadır. *Q*-Öğrenme algoritması Alpaydın'ın da belirttiği üzere aşağıdaki gibidir [12].

Bütün $Q(s,a)$ değerlerini keyfi olarak tanımla
Tüm bölümler için
  s değerini tanımla
  Belirlenen döngü sayısı kadar ya da durum hedef duruma yakınsayana kadar
    ε açgözlü algoritmasını kullanarak Q matrisinden en uygun hareketi (a) seç
    a hareketini sisteme uygula, r ve s' değerlerini gözlemle
    $Q(s,a)$değerini güncelle
    $Q(s,a) \leftarrow Q(s,a) + \eta\,(r + \gamma\, \text{maks}_{a'}\, Q(s',a') - Q(s,a))$
    $s \leftarrow s'$

3.3. Sarsa Algoritması

Sarsa algoritması Q ödül matrisinden seçilen hareketi ve sonraki bir hareketi seçerek, o hareketin ödülünü de göz önünde bulundurarak Q ödül matrisinin güncellenmesini sağlar ve sözde kodu aşağıdaki gibidir [12-13].

Bütün $Q(s,a)$ değerlerini keyfi olarak tanımla
Tüm bölümler için
  s değerini tanımla
  ε açgözlü algoritmasını kullanarak Q matrisinden en uygun hareketi (a) seç
  Belirlenen döngü sayısı kadar ya da durum hedef duruma yakınsayana kadar
    a hareketini sisteme uygula, r ve s' değerlerini gözlemle
    ε açgözlü algoritmasını kullanarak Q matrisinden en uygun hareketi (a') seç
  $Q(s,a)$değerini güncelle:
  $Q(s,a) \leftarrow Q(s,a) + \eta\,(r + \gamma\, Q(s',a') - Q(s,a))$
  $s \leftarrow s',\ a \leftarrow a'$

Q-Öğrenme algoritmasından farklı olarak bütün olası sonraki hareketleri incelemez ancak Sarsa da Q-Öğrenme gibi dinamik modele ihtiyaç duymadan öğrenebilen bir algoritmadır.

3.4. Sarsa (λ) Algoritması

Sarsa (λ)'da Sarsa'dan farklı olarak ödül matrisinin güncellenmesi, güncellenen durumun en son ne zaman gelindiğine bakılarak yapılır [12]. Sarsa (λ) algoritmasının sözde kodu aşağıdaki gibidir [12],

Bütün $Q(s,a)$ değerlerini keyfi olarak tanımla
Bütün s,a değerleri için $e(s,a) \leftarrow 0$ olarak tanımla
Tüm bölümler için
  s değerini tanımla
  ε açgözlü algoritmasını kullanarak Q matrisinden en uygun hareketi (a) seç
  Belirlenen döngü sayısı kadar ya da durum hedef duruma yakınsayana kadar
    a hareketini sisteme uygula, r ve s' değerlerini gözlemle
    ε açgözlü algoritmasını kullanarak Q matrisinden en uygun hareketi (a') seç
    $\delta \leftarrow r + \gamma\, Q(s',a') - Q(s,a)$
    $e(s,a) \leftarrow 1$
    Bütün s,a değerleri için
      $Q(s,a) \leftarrow Q(s,a) + \eta\delta\, e(s,a)$
      $e(s,a) \leftarrow \gamma\, \lambda\, e(s,a)$
    $s \leftarrow s',\ a \leftarrow a'$

Algoritmada daha önce tanımlanan değişkenlere ek olarak $e(s,a)$ -seçilebilirlik belirtisi- s durumu ve a hareketinin en son ne zaman ziyaret edildiğinin belirten ifadedir.

4. Benzetim Çalışmaları

Bu çalışmada önceki bölümde belirtilen pekiştirmeli öğrenme algoritmaları gerçeklenerek benzetim sonuçları elde edilmiştir. *Q*-Öğrenme ve Sarsa algoritmalarında Değer Yineleme ve Sarsa (λ) algoritmalarında kullanılandan daha geniş durum-hareket uzayları tanımlanmıştır. Değer Yineleme ve Sarsa (λ) algoritmaların gerçeklenmesi sırasında kullanılan durum ve hareket 1575 durum ve 211 hareket kullanılarak döner ters sarkacın kontrolü yapılmaya çalışılmıştır. Pekiştirmeli öğrenme algoritmalarının kullanılmasıyla elde edilecek

kontrol işaretleri ayrık olduğundan, durum ve hareket sayısının uzayı ne kadar kapsadığı önem taşımaktadır. Olay uzayının büyüklüğünün artması işlem zamanını arttıracağından uzayı küçük tutup gelebilecek ara değerler için uygulanacak kontrol sinyali doğrusal ara değerleme ile oluşturulmuştur. Böylelikle ayrık noktalarda tanımlanan karar bilgileri bu doğrusal fonksiyon yardımı ile uzayın her noktasında hesaplanabilir hale getirilmiştir. Yapılan uygulamalarda kullanılan durum ve hareketler aşağıdaki vektörlerin kartezyen çarpımı ile elde edilen ayrık uzay noktalarından oluşmaktadır.

$$\alpha = [\text{-10 -4 -1 -0.5 0 0.5 1 4 10}] \quad (12)$$

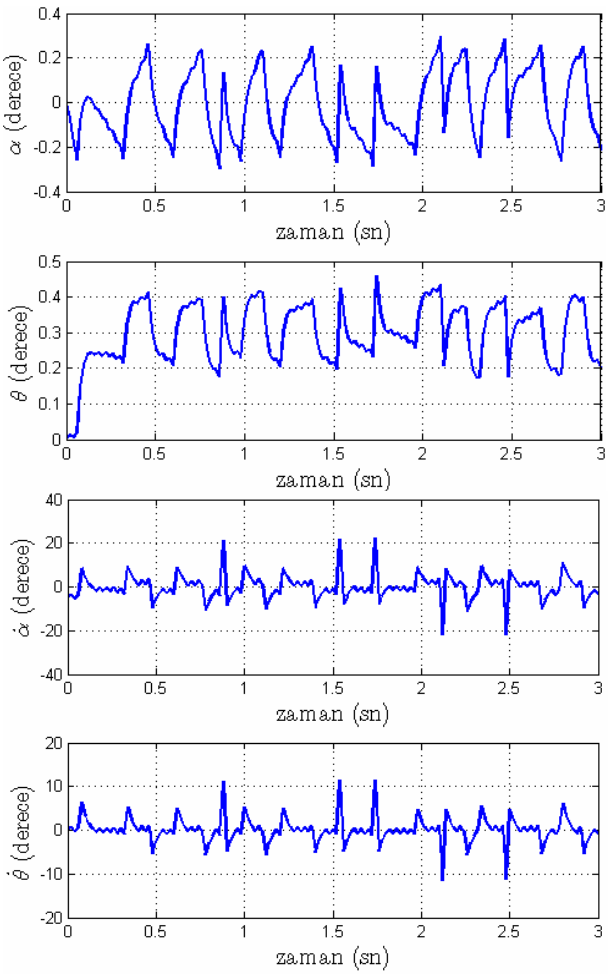
$$\dot{\alpha} = [\text{-675 -135 -30 0 30 135 675}] \quad (13)$$

$$\theta = [\text{-25 -10 0 10 25}] \quad (14)$$

$$\dot{\theta} = [\text{-30 -10 0 10 30}] \quad (15)$$

$$A = [\text{-6 -5 -4 -3 -2 } \mathbf{V} \text{ 2 3 4 5 6}] \quad (16)$$

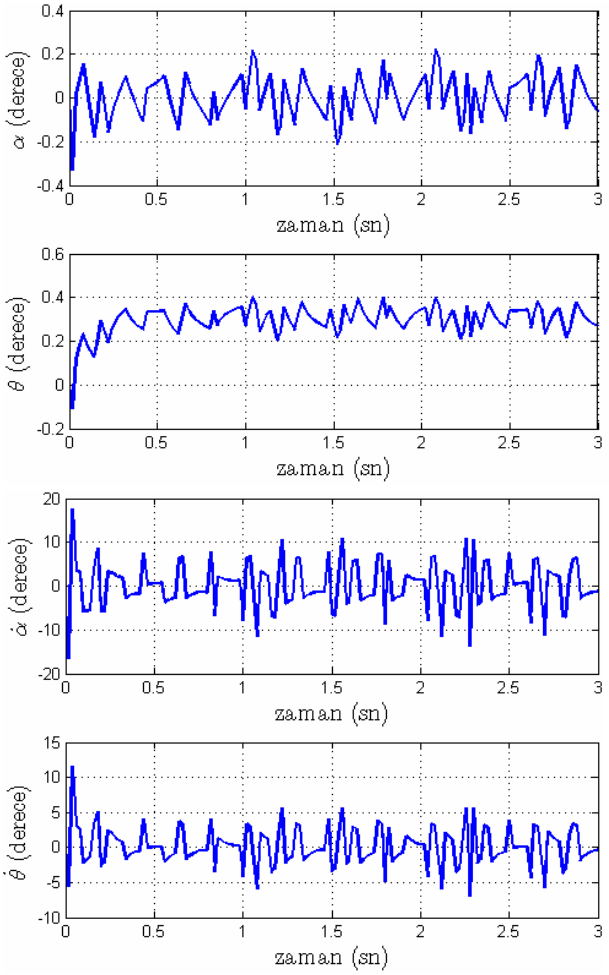
Burada **V**= [-1, -0.99, -0.98, ..., 0.98, 0.99 1] olarak tanımlanmıştır.



Şekil 3: Değer Yineleme algoritmasıyla elde edilen sonuçlar

Denklem (12)-(16)'da belirtilen her bir durum ve hareket için öğrenme aşamasında ödül değerleri belirlenir. Ödül değeri belirlenirken kullanılan ödül fonksiyonu α, θ ve türevlerinin hedef değerlerine olan uzaklıkları dikkate alınır. Sistem çalışırken, bulunduğu durumda yapacağı hareket için eğitim sonunda oluşan çizelgeden [1] kendine en yakın iki durumun işaret ettiği hareketlere bakarak oluşturulmuş doğrusal fonksiyonla son değer belirlenir. Uygulama sonuçları Şekil 3-6 ile verilmiştir.

Sistem dinamikleri ve model parametreleri doğru olarak elde edildiğinde Değer Yineleme algoritması oldukça tatminkâr sonuçlar vermektedir. Sarsa (λ) algoritmasına göre daha iyi sonuç vermesine rağmen, öğrenme işlemi de çok daha fazla zaman almaktadır. Değer Yineleme algoritmasının öğrenme süresi verilen durum-hareket uzayı için yaklaşık 150 dakika sürerken, Sarsa (λ) algoritması için yaklaşık olarak 15-25 dakikada sonuçlanmıştır. Öğrenme süresinin uzun olmasına rağmen bu algoritma, bütün durum ve hareketler için en uygun durum-hareket ikilisini tespit edebilmektedir. Değer Yineleme algoritması kullanılarak yapılan kontrolde elde edilen deney sonuçları Şekil 3'te verilmektedir. Şekilden de görüldüğü üzere sarkacın devrilme yönünde hızlı hareketler yaparak istenen dik pozisyon civarında bir hareket elde edilebilmektedir.

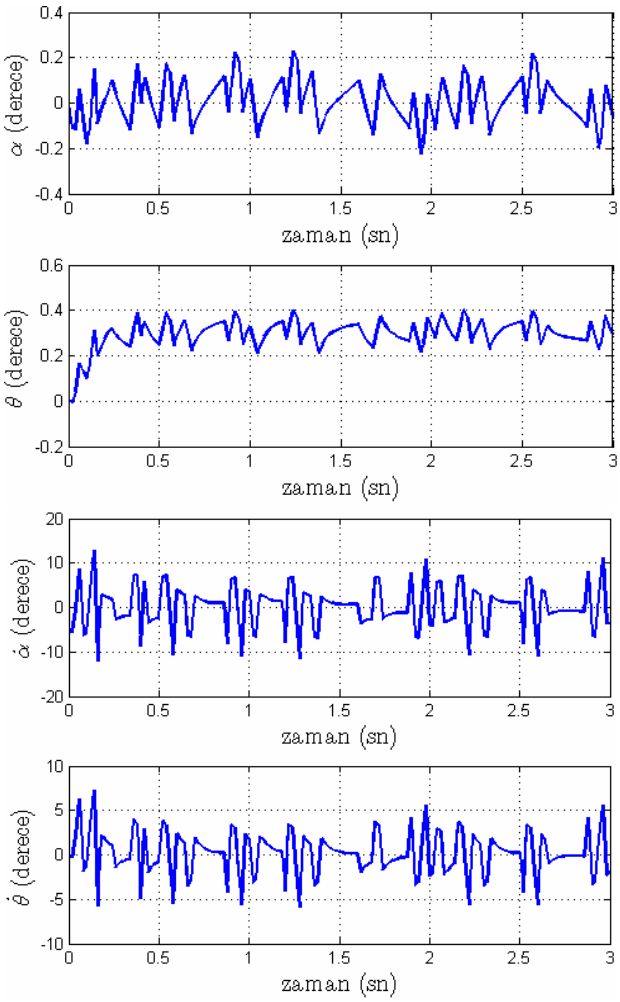


Şekil 4: *Q*-Öğrenme algoritmasıyla elde edilen sonuçlar

[1] İng. *Lookup Table*

Q-Öğrenme ve Sarsa algoritmaları, uygulanan diğer algoritmalar için kullanılan aynı durum-hareket uzayı kullanılarak eğitildiğinde elde edilen sonuçlar başarılı bulunmamıştır. Bunun nedeni durum-hareket uzayının uygun bir şekilde tasvir edilememesinden ya da ödül fonksiyonunun hedef ile ilişkisinin yeterli düzeyde olmamasındandır. Bu nedenle *Q*-Öğrenme ve Sarsa algoritmaları uygulanırken durum-hareket uzayı genişletilmiş ve diğer iki yönteme göre daha iyi bir şekilde hedef noktası etrafında başarım sağlamışlardır, fakat durum-hareket ikilisi sayısının artışı öğrenme süresini büyük ölçüde etkilemiştir. 15-25 dakika arasında olan öğrenme süresi yaklaşık olarak 200 dakikaya ulaşmıştır. Sonuçlar Şekil 4-5'te gösterilmektedir.

Q-Öğrenme, Sarsa ve Sarsa (λ) algoritmalarının üçünde de Alpaydın [12]'ın belirttiği yapıdan farklı olarak, bu uygulamada başlangıçta *Q* matrisini rastgele tanımlamak yerine model kullanılarak bu değerler hesaplanıp tanımlama yapılmıştır. Bu işlem sayesinde sistemin performansının büyük ölçüde arttığı görülmüştür. Buna göre elde dinamik modele dair bilgi olmasa bile bu değerleri rastgele atamak yerine tahmini değerler atamakla başarılı bir kontrol işlevinin daha iyi öğrenilmesine zemin hazırlayacaktır. Bu sayede daha az sayıda durum-hareket ikilisi kullanarak sistemin başarıyla kontrol edilmesi de sağlanabilecektir.



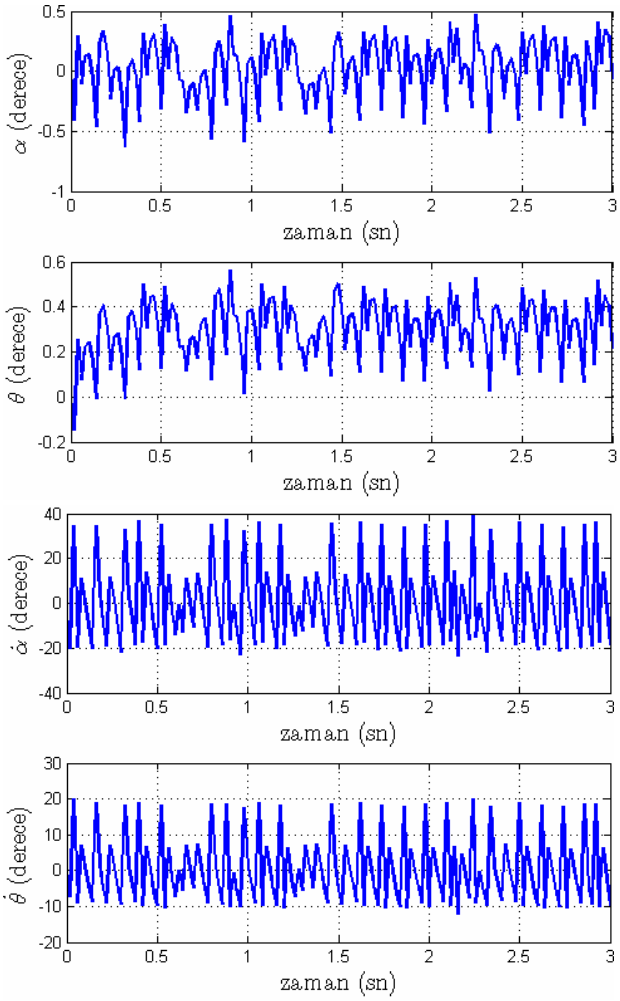
Şekil 5: Sarsa algoritmasıyla elde edilen sonuçlar

Sarsa (*λ*) algoritmasında, Sarsa'dan farklı olarak seçilebilirlik belirtisi dikkate alındığından *Q* matrisinin güncellenmesi aynı durum-hareket uzayı ile eğitilen *Q*-Öğrenme ve Sarsa'ya göre daha iyi sonuçlar verebilmektedir. Şekil 6'da görüldüğü üzere Sarsa (λ)'da α hedeflenen değere ulaşmıştır ve o değer etrafında bulunması sağlanmıştır.

5. Sonuçlar

Değer Yineleme yöntemi dinamik modele dair ön bilgi mevcut ise çok iyi sonuçlar verebilmektedir, Her ne kadar dinamik modelde kesinsizlikler olabilse de nominal modelin bilinmesi bu algoritma için yeterli ön bilgi oluşturmakta ve oldukça iyi sonuçlar alınabilmektedir.

Modelden bağımsız olarak çalışan öğrenme algoritmalarından denklem (12)-(16) ile verilen durum ve hareket uzaylarıyla kabul edilebilir bir hatayla hedef noktasına ulaşan Sarsa (*λ*) algoritması olmuştur, ayrıca modelden bağımsız olmasına rağmen Değer Yinelemeye göre çok az bir hata ile hedef noktası çevresinde stabilizasyon sağlamıştır. *Q*-Öğrenme ve Sarsa algoritmaları durum-hareket uzayının kapsamı genişletildiğinde istenen hedefe ulaşmıştır, fakat Sarsa (*λ*)'ya göre öğrenme süresi çok artmıştır. Sonuçlara göre, Değer Yineleme ve Sarsa (*λ*) seçilen durum ve uzaylara uygun bir ödül fonksiyonuyla daha toleranslı olarak öğrenebilmektedir.



Şekil 6: Sarsa (λ) algoritmasıyla elde edilen sonuçlar

Elde edilen bulgulara göre durum-hareket uzayı ve ödül fonksiyonu uygun biçimde ayarlandığında başarılı bir kontrol yapılabilmektedir. Bu çalışmada elde edilen sonuçlar pekiştirmeli öğrenme algoritmalarının dinamik modeli bilinen ya da bilinmeyen sistemlerde başarıyla uygulanabileceğini ortaya koymuştur.

6. Teşekkür

Bu çalışma TUBİTAK 1001 Programı (Kontrat No 107E137) tarafından desteklenmiştir. Yazarlar TOBB ETÜ İHA Laboratuarını (http://donerkanat.etu.edu.tr) takdim etmekten memnuniyet duyarlar.

7. Kaynakça

[1] D. Maravall, C. Zhou ve J. Alonso, "Hybrid Fuzzy Control of the Inverted Pendulum via Vertical Forces", *International Journal of Intelligent Systems*, Cilt: 20, s:195-211, 2005.

[2] S. Li, C. Huo, Y. Liu, "Inverted Pendulum System Control by Using Modified PID Neural Network", The 3rd Int. Conf. on Innovative Computing Information and Control, s:426-429, 2008.

[3] H. Yu, Y. Liu ve T. Yang, "Tracking Control of A Pendulum-driven Cart-pole Underactuated System", *IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics*, s:2425-2430, 2007.

[4] S. Russell ve Peter Norvig, *Artificial Intelligence – A Modern Approach*, Prentice Hall, 2003.

[5] Q.H. Wu ve A.C. Pugh, "Reinforcement Learning Control of Unknown Dynamic Systems", *IEE Proceedings on Control Theory and Applications*, Cilt: 140, No: 5, s:313-322, 1993.

[6] G. Buskey, J. Roberts ve G. Wyeth, "Online Learning of Autonomous Helicopter Control", Proceedings of the Australasian Conference on Robotics and Automation, s:19-24, 2002.

[7] S.L. Waslander, G.M. Hoffmann, J.S. Jang, ve C.J. Tomlin, "Multi-Agent Quadrotor Testbed Control Design: Integral Sliding Mode vs. Reinforcement Learning", 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, August 2-6, Edmonton, Alberta, Kanada, s.468-473, 2005.

[8] M.A. Khanesar, M. Teshnehlab ve M.A. Shoorehdeli, "Sliding Mode Control of Rotary Inverted Pendulum", 2007 Mediterranean Conference on Control and Automation, s: 1-6, 2007.

[9] V. Sukontanakarn ve M. Parnichkun, "Real-Time Optimal Control for Rotary Inverted Pendulum", American Journal of Applied Sciences, Cilt: 6, No:6, s: 1106-1115, 2009.

[10] L.P. Kaelblimg, M.L. Littman ve A.W. Moore, "Reinforcement Learning: A Survey", *Journal of Artificial Research*, Cilt: 4, s:237-285, 1996.

[11] J.A. Bagnell, ve J.G. Scheneider, "Autonomous Helicopter Control using Reinforcement Learning Policy Search Methods", IEEE Proceedings International Conference on Robotics and Automation, Cilt: 2, s:1615-1620, 2001.

[12] E. Alpaydın, *Introduction to Machine Learning*, The MIT Press, 2004.

[13] R.S. Sutton ve A.G. Barto, *Reinforcement Learning – An Introduction*, The MIT Pres, 1998.

[14] Quanser SRV02-Series, Rotary Experiment # 7, Rotary Inverted Pendulum Student Handout, Revision: 01.