Düzce Üniversitesi Bilim ve Teknoloji Dergisi

Araştırma Makalesi

DA Makinesi Hız Kontrolünün Q-Öğrenme Tabanlı PID Kontrolör ile Gerçek-Zamanlı Uygulaması

Bekir Murat AYDIN [a,*], Burhan BARAKLI [a]

[a] *Elektrik Elektronik Mühendisliği Bölümü, Mühendislik Fakültesi, Sakarya Üniversitesi, Sakarya, Türkiye*
** Sorumlu yazarın e-posta adresi: murataydin@sakarya.edu.tr*
DOI: 10.29130/dubited.1111267

Öz

Çalışmamızda Q-öğrenme tabanlı adaptif PID kontrolörün gerçek zamanlı bir sistemdeki performansı incelenmiştir. Gerçek zamanlı sistem olarak DA makine hız kontrolü sistemi tercih edilmiştir. DA makine sisteminden gelen hata sinyali üzerinden sistemin durum bilgisi ve Q-öğrenme yöntemi için ödül sinyali hesaplanmaktadır. Durum bilgisi ve ödül sinyali yardımı ile PID katsayıları artırılıp azaltılarak optimal katsayılara ulaşılmaktadır. Her PID katsayısı için bir adet Q-tablosu tanımlanmıştır. Simülasyon çalışması ve gerçek zamanlı uygulama ile kontrolör performansı incelenmiştir. Pekiştirmeli öğrenme ile tasarlanan kontrolcünün klasik PID yapısı gibi başarılı olduğu tespit edilmiştir.

Anahtar Kelimeler: *Pekiştirmeli öğrenme, Adaptif PID, DA makinesi*

Real-Time Application of DC Machine Speed Control with Q-Learning Based PID Controller

Abstract

In this study, the Q-learning based adaptive PID controller's performance has been examined on a real-time system. DC machine speed control system selected as a real-time system. The system's state and reward signal are calculated by using the error signal of the DC machine speed control system. With the help of state and reward signals, the algorithm adjusts PID parameters to find the optimal solution. One Q-table is defined for each PID parameter. Controller performance was examined with a simulation study and real-time application. It has been determined that the controller designed with reinforcement learning is successful like the classical PID structure.

Keywords: *Reinforcement learning, Adaptive PID, DC machine*

Geliş: 29/04/2022, Düzeltme: 31/05/2022, Kabul: 05/06/2022

I. GİRİŞ

Makine öğrenmesi algoritmalarından birisi olan pekiştirmeli öğrenme, yapı itibariyle diğer makine öğrenmesi yöntemlerinden farklıdır. Literatürde kullanılan makine öğrenmesi yöntemleri gözlemcili ve gözlemcisiz olarak ikiye ayrılabilirken pekiştirmeli öğrenme bu iki sınıfa da ait değildir. Temel olarak canlıların öğrenme içgüdüsünü örnek alır. Çevresi ile etki-tepki ilişkisi içerisindeki bir canlı karmaşık davranışları çevresinden gelen tepkiler ile öğrenebilir ve bulunduğu durumlar ile seçeceği eylemleri ilişkilendirir [1].

Canlıların öğrenmesi alanında yapılan çalışmalar ve makine öğrenmesi alanındaki gelişmeler 1950'li yıllara dayanmaktadır. Minksy, geliştirdiği stokastik nöral analog pekiştirmeli bilgisayar (SNARC) ile bir farenin beyin modeli ile bir labirentten çıkma bulmacasının benzetim çalışmasını yapmıştır [2]. Bu çalışma ile pekiştirmeli öğrenmenin karmaşık problemlerin çözümünde kullanılabileceği gündeme gelmiştir ve birçok pekiştirmeli öğrenme yöntemi geliştirilmiştir. Watkins ve Dayan, Richard E. Bellman tarafından ortaya atılan dinamik programlama yöntemi [1], [3]–[5] ile zamansal fark öğrenmeyi bir araya getirerek Q-öğrenme algoritmasını geliştirmişlerdir [6]. Q-öğrenme algoritması az sayıda durum ve aksiyon içeren sistemlerde optimal noktaya başarılı bir şekilde yakınsayabilmektedir. Sistemin durum ve aksiyon sayısı arttıkça Q-öğrenme algoritmasının optimal noktaya yakınsaması zorlaşmaktadır. 2015 yılında yapılan bir çalışma ile derin öğrenme ve Q-öğrenme birleştirilerek Derin Q-öğrenme Ağı (DQN) algoritması geliştirilmiştir [7]. Makalede DQN kontrolcülerinin karmaşık sistemlerde oldukça başarılı sonuçlar verdiği görülmüştür ve DQN kontrolcülerin karmaşık sistemlerin kontrolünde kullanılmasının yolu açılmıştır. Karmaşık problemlerde pekiştirmeli öğrenmenin başarılı sonuçlar verdiği birçok çalışma yapılmıştır [7]–[12]. Guan ve Yamamoto yaptıkları çalışmada actor-critic yöntemini kullanarak; güçlü doğrusal olmayan durumlar içiren bir sistem için başarılı bir PID kontrolcü tasarlamıştır [13]. Yapılan diğer iki çalışmada farkı yüzeylerde farklı davranışı olan tekerlekli bir mobil robot için pekiştirmeli öğrenme tabanlı PID kontrolör tasarlanmıştır. Kabul edilebilir sınırlar içinde sistemi kontrol ettiği görülmüştür [14]–[16]. Başka bir çalışmada gemilerdeki dinamik pozisyonlama sisteminde üç-eksenli hareket için kullanılan üç adet PID kontrolcü için Q-öğrenme kullanılmıştır. İki eksende başarılı sonuçlar verirken üçüncü eksendeki harekette osilasyona sebep olmuştur fakat ödül fonksiyonu güncellenerek düzeltilebilir [17]. Diğer bir çalışmada karmaşık, doğrusal olmayan bir sistemde actor-critic yönteminin konvansiyonel PID'den başarılı sonuçlar verdiği görülmüştür [18]. Dikey kalkış sistemi için klasik PID yönteminden hem sinüsoidal hem sabit referans için başarılı olduğu görülmüştür [19]. Başka bir çalışmada hibrit mikro şebekelerde frekans dalgalanmalarını sönümlemek için kullanılan PID kontrolcü pekiştirmeli öğrenme yardımı ile tasarlanmış ve salp sürüsü algoritması ve klasik PID algoritmasına göre daha başarılı bir kontrolcü elde edilmiştir [20]. Yapılan bir çalışmada kayan-kipli kontrol ile beraber pekiştirmeli öğrenme yöntemi ile geliştirilen kontrolör beraber kullanılarak belirsiz parametreli bir sistemde başarılı sonuçlar veren bir kontrolcü elde edilmiştir [21]. Başka bir çalışmada da enterkonnekte bir şebekede yük frekansı kontrolöründe pekiştirmeli öğrenme tabanlı geliştirilen kontrolcünün üstünlüğü görülmüştür [22]. Reaktif güç kontrolü, trafik ışıklarının kontrolü, video oyunları için kontrol, adaptif PID tasarımı gibi çeşitli uygulamalarda pekiştirmeli öğrenme kullanılmıştır.

PID kontrolcüler uygulanabilirliği ve basitliği açısından endüstride en çok tercih edilen kontrolcülerden birisidir. PID kontrolcü üstüne yapılan çalışmalar endüstride hızla uygulamaya konması açısından akademi ve endüstri arasında önemli bir ortak çalışma alanıdır [23]. Uzun yıllardır optimal PID katsayılarının ayarlanması için yöntemler geliştirilmektedir. Pekiştirmeli öğrenme yöntemlerindeki gelişmeler PID kontrolcü tasarımı alanında da kullanılmaya başlamıştır. Fırçasız DC motorun hız

kontrolü için yapılan bir çalışmada DDPG-PID kontrolcünün standart PID kontrolcüden daha iyi sonuçlar verdiği gözlemlenmiştir [24]. PID parametre optimizasyonu için parçacık sürü algoritması ve genetik algoritma ile pekiştirmeli öğrenme algoritması karşılaştırılmış, pekiştirmeli öğrenmenin daha hızlı yakınsadığı gözlemlenmiştir [25]. Bu konuda yapılan diğer bir çalışmada Derin Q-öğrenme ve Advantage Actor-Critic Yöntemi ile tasarlanan kontrolcülerin özellikle sistem parametresinin değiştiği durumlarda daha başarılı olduğu görülmüştür [26].

Bu çalışmada Q-öğrenme yöntemi ile DA makinesi hız kontrolü gerçekleştirilmiştir. Benzetim çalışması ve gerçek-zamanlı çalışmaya ait sonuçlar elde edilmiş ve değerlendirilmiştir. Önerilen yöntem ile gerçekleştirilen kontrolcünün başarılı bir şekilde kontrol işlemini gerçekleştirdiği görülmüştür. Pekiştirmeli Öğrenme, Q-öğrenme, kontrol yöntemi ve kullanılan yardımcı yöntemler ile alakalı bilgiler Materyal ve Yöntem başlığı altında verilmiştir. Bulgular ve Tartışma bölümünde sistemin benzetim ortamı ve gerçek-zamanlı çalışmaya ait olan sonuçları verilmiştir. Q-öğrenme ile tasarlanan kontrolcünün eğitim sürecinde kullanılan sistemden farklı sistemler üzerindeki performansı incelenmiştir.

II. MATERYAL VE YÖNTEM

A. Pekiştirmeli Öğrenme

Pekiştirmeli öğrenme yöntemi Markov özelliğini sağlar. Yani ayrık zamanlı bir sistemde, sistemin $n+1$ anındaki durumu sadece n anındaki durumuna bağlıdır. Eğer sistem sonlu durum ve sonlu aksiyon uzayına sahipse ve Markov özelliğini sağlıyorsa uygulama Sonlu Markov Karar Verme Süreci diye adlandırılır [1]. Pekiştirmeli öğrenmede kontrolcü, diğer adıyla ajan, sistem ile üç temel işaret üzerinden iletişim kurar, bunlar; durum sinyali $s \in \mathcal{S}$, aksiyon sinyali $a \in \mathcal{A}(\mathcal{S})$ ve sistemden gelen skaler ödül sinyali $r \in \mathcal{R}(s,a)$'dir. Durum sinyali, sistemin bulunduğu hal hakkında ajana bilgi verir. Ödül sinyali, sistemin durumunun ne kadar iyi veya kötü olduğu hakkında geri bildirimdir. Aksiyon sinyali ise sisteme ajan tarafından uygulanarak sistemi bir sonraki duruma geçirir [1], [27]. Ajan, ortam ile etkileşime girerek durumlar ile eylemler arasında bir eşleştirme yapar. Bu eşleştirmeye politika, $\pi(a|s)$ denir. Ajan, deneyimler ile politika belirlemek için değer fonksiyonu, $\mathcal{V}_\pi(s)$ 'i kullanır. Değer fonksiyonu, ortamdan gelen durumun ne kadar iyi olduğunu belirtir ve toplam beklenen ödül miktarı, G_n'nin bir fonksiyondur. s durumu için seçilecek olan a aksiyonunun ne kadar iyi olduğunu ise aksiyon-değer fonksiyonu olan $Q(s\,,a)$ belirtir.

Toplam ödül miktarı, G_n;

$$G_n = R_{n+1} + \gamma R_{n+2} + \gamma^2 R_{n+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{n+k+1} \tag{1}$$

Şeklinde ifade edilir. Eşt. (1)'de γ, $0 \leq \gamma \leq 1$ arasında seçilen zayıflatma faktörüdür. γ sıfıra ne kadar yakınsa ajan anlık ödüllere o kadar çok odaklanır. Çalışmamızda γ seçimi yapılırken ajanın toplam ödül miktarına daha çok odaklanması beklenmiştir. Bu nedenle $\gamma = 0.99$ olarak seçilmiştir.
Buna göre sistemin bulunduğu durumun değer fonksiyonu;

$$\mathcal{V}_\pi(s) = E_\pi[G_n \mid S_n = s] \tag{2}$$

Eşt. (2)'de E_π değeri beklenen değer anlamına gelmektedir. Eşt. (2)'de verilen denklem düzenlenerek Eşt. (3) ve Eşt. (4)'te verilen denklemler elde edilmektedir.

$$\mathcal{V}_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a)\,[G_n \mid S_n = s] \tag{3}$$

$$\mathcal{V}_\pi(s) = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{n+k+1} \mid S_n = s] \tag{4}$$

Eşt. (3)'te p, stokastik sistemlerde s durumundan s' durumuna geçiş olasılığıdır.
Eşt. (4)'teki ifade düzenlenirse Eşt. (5)'te verilen denklem elde edilir.

$$\mathcal{V}_\pi(s) = E_\pi[\, R_{n+1} \; + \gamma \sum_{k=0}^{\infty} \gamma^k R_{n+k+2} \mid S_n = s] \tag{5}$$

O halde değer fonksiyonu Eşt. (6)'da verilen denklem ile ifade edilebilir.

$$\mathcal{V}_\pi(s) = E_\pi[R_{n+1} + \gamma\, \mathcal{V}_\pi(s') \mid S_n = s] \tag{6}$$

Eşt. (6)'da verilen ifade ile benzer şekilde; aksiyon-değer fonksiyonu, Eşt. (7)'deki gibi ifade edilebilir.

$$Q(s,a) = E_\pi[R_{n+1} + \gamma Q(s', a') \mid S_n = s,\, A_n = a] \tag{7}$$

Pekiştirmeli öğrenme sürecinde ajanın amacı optimal politika, π_*'ı bulmaktır. Optimal politikanın bulunması için çeşitli yöntemler geliştirilmiştir. Modeline bütünüyle hâkim olunan bir sistemde dinamik programlama yöntemiyle optimal politika elde edilebilir. Modeli tam olarak bilinmeyen sistemlerde ise Monte-Carlo ve Zamansal-Fark yöntemleri ile model kestirimi yapılabilir. Optimal politikanın bulunması için sistemin tüm durumlarının ve aksiyonlarının deneyimlenmesi gerekir.
Optimal değer fonksiyonu ve optimal aksiyon-değer fonksiyonları Eşt. (8) ve Eşt. (9)'da verilmiştir.

$$\mathcal{V}_*(s) = \max_\pi \mathcal{V}_\pi(s) \tag{8}$$

$$Q_*(s,a) = \max_\pi Q_\pi(s,a) \tag{9}$$

Şeklindedir.

A. 1. Q-Öğrenme Algoritması

Q-öğrenme, modelden bağımsız bir zamansal-fark yöntemidir. Aynı zamanda bir çeşit dinamik programlama yöntemidir [28]. Q-öğrenme algoritmasında temel amaç, sistemin bütün durumları ve her bir durum için seçilebilecek olan aksiyonlara ait aksiyon-değer fonksiyonlarını içeren bir Q-tablosu oluşturmaktır. Optimal Q-tablosu oluştuktan sonra bu tablo yardımı ile her bir durum için toplam ödül miktarını en yüksek yapacak olan aksiyonların seçilmesini sağlamaktır. Deterministik bir sistem için aksiyon değer fonksiyonu Eşt. (10)'da verilmiştir.

$$Q(s,a) = [R_{n+1} + \gamma Q(s',a') \mid S_n = s,\ A_n = a] \tag{10}$$

Eğitim sürecinde ajan, sistemin bütün durumlarını ve her durum için seçebileceği aksiyonları deneyimlemesi gerekir. Daha önceden deneyimlemiş olduğu aksiyonları da tekrar tekrar deneyimleyerek aksiyon-değer fonksiyonunu optimal yapması gerekir. Bu sebeple eğitim süreci boyunca ajanın keşif-istifade dengesinin kurulması gereklidir. Keşif-istifade dengesi için literatürde genellikle adaptif ε -greedy algoritması kullanılır. Çalışmamızda da bu yöntem tercih edilmiştir.

$$\pi(a|s) = \begin{cases} Rastgele\ \ a \in \mathcal{A}(\mathcal{S}), & eğer\ \xi < \varepsilon \\ \underset{a \in \mathcal{A}(s)}{\operatorname{argmax}}\, Q(s,a), & diğer \end{cases} \tag{11}$$

Eşt. (11)'de adaptif ε -greedy yöntemi verilmiştir. ξ değeri, $0 \leq \xi \leq 1$ aralığında rastgele seçilen bir sayıdır. ε değeri ise Eşt. (12)'de verilen denklem ile 1'den başlayarak 0.005 değerine ulaşana kadar azalmaktadır.

$$\varepsilon = \varepsilon * (1 - \varepsilon_{decay}) \tag{12}$$

Her iterasyonda rastgele sayı üreteci yardımı ile bir ξ sayısı üretilmektedir. Aksiyon seçiminden önce Keşif mi, istifade mi yapılacağına Eşt. (12)'de verilen ifade ile ulaşılmaktadır.

Her iterasyonda kontrolcü aldığı ödüle göre $Q(s,a)$ değerini güncelleyerek optimal $Q_*(s,a)$ ve dolayısıyla optimal politikayı $\pi_*(a|s)$ elde edebilir. Ajan, $Q(s,a)$ değerlerini Eşt. (13)'de verilen ifade yardımı ile güncellemektedir.

$$Q(s,a) = Q(s,a) + \alpha[R + \gamma(Q(s',a') - Q(s,a))] \tag{13}$$

Eşt. (13)'te verilen α, öğrenme oranıdır. α ve γ değerleri birer hiper-parametredir. α değeri sıfıra ne kadar yakınsa öğrenme süreci o kadar uzar fakat eğitim daha doğru bir noktaya yakınsar. α değeri büyüdükçe eğitim süresi kısalır ama yakınsadığı nokta optimal olmayabilir. γ değeri büyüdükçe ajan toplam ödül miktarına daha çok odaklanmaktadır. Çalışmamızda $\alpha = 0.01,\ \gamma = 0.99$ olarak seçilmiştir. Simgesel dilde Q-öğrenme algoritması;

1: Q-Tablosu oluştur, ilk değerleri sıfır ata. $Q_i\big(\forall s \in \mathcal{S}, \forall a \in \mathcal{A}(\mathcal{S})\big) = 0$
2: Sistemi başlangıç durumuna getir
3: s durumu için bir a aksiyonu seç

4: Sisteme a aksiyonunu uygula
5: Sistemden *s'* ve *R(s,a)* bilgisini oku
6: *s'* 'ye göre bir *a'* seç
7: $Q(s,a) = Q(s,a) + \alpha[R(s,a) + \gamma(Q(s',a') - Q(s,a))]$
8: *s = s', a= a'*
9: goto 4

B. PID Kontrol

PID kontrol en yaygın kullanılan kontrol yöntemlerinden birisidir. Girişindeki sinyali oransal, integratör ve türevsel parametreleri ile çarparak çıkış sinyali üretir. Basitliği ve uygulanabilirliği sebebiyle en çok kullanılan kontrol yöntemlerinden birisidir.

$$U[k] = K_p e[k] + K_i \sum_{i=0}^{k} e[i]\,\Delta_k + K_d \frac{e[k] - e[k-1]}{\Delta_k} \tag{14}$$

Eşt. (14)'te verilmiş olan $e[k]$ hata sinyali ve Δ_k örnekleme zamanı olmak üzere; K_p, K_i ve K_d ise PID kontrolcü parametreleridir. PID kontrolcü sisteme iki adet kök ekler. Seçilen köklerin yerine göre sistem, istenilen davranışa yakın şekilde kontrol edilir.

C. Ayrık Gruplama

Sürekli zamanlı sistemlerde sistemden okunan sinyaller sürekli işaret formundadır. Q-öğrenme kontrolcüsü ayrık işaretleri işleyebilmektedir. Veri gruplama işlemi sürekli zaman sistemden alınan verileri kabul edilebilir hata sınırları içerisinde belirli gruplara veya kovalara ayırarak Q-öğrenme algoritmasının gerçek-zamanlı sistemlerde kullanılabilmesini sağlar.

$$x = \begin{cases} 0 & X_{kon} < X_{min} \\ N & X_{kon} > X_{max} \\ \dfrac{(X_{kon} - X_{min})}{X_{max} - X_{min}} * N & X_{min} \le X_{kon} \le X_{max} \end{cases} \tag{15}$$

Eşt. (15)'te verilmiş olan x, ayrık değişken. X_{kon} sürekli zaman sistemden alınan sinyal, X_{min} ve X_{max} sinyalin alabileceği maksimum ve minimum değerler, N ise Seçilen aralıkta kaç adet grup diğer adıyla kova bulunacağını ifade eder. N sayısı arttıkça kontrol algoritmasını çalıştıran işlemcinin yükü artmaktadır. X_{min}, X_{max} ve N değerleri kontrol edilecek olan sisteme uygun olarak seçilmelidir. Çalışmamızda $N = 100$ olarak seçilmiştir.

D. Ödül ve Aksiyon Fonksiyonu

PID kontrol sistemi için ödül fonksiyonu olarak hatanın mutlak değerini sıfıra yaklaştıracak bir fonksiyonu seçilmiştir. Ajanın örnekleme zamanı 10 milisaniye olmak üzere ödül fonksiyonu Eşt. (16)'da verilmiştir.

$$R(s,a \mid s = S_n = s,\ A_n = a) = -sign(|e_n| - |e_{n-1}|) \tag{16}$$

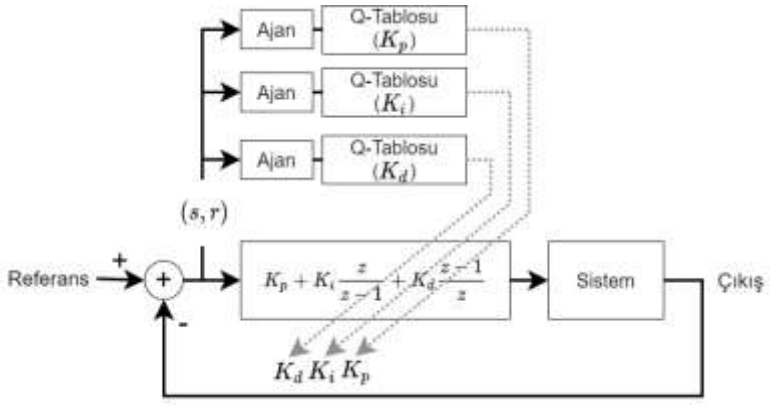
Eşt. (16)'da $|e_n|$, n anındaki hatanın mutlak değeri. $|e_{n-1}|$, $n-1$ anındaki hatanın mutlak değeridir. Hatanın n anında mutlak değeri, bir adım önceki değerinden daha küçükse olarak 1 puan, daha büyükse -1 puan ödül olmaktadır. Pekiştirmeli Öğrenme Kontrolcüsünün amacı toplam ödül miktarını maksimum yapacak olan Q-tablosunu belirlemektir. Aksiyonlar her ajan için PID parametrelerinin arttırılması, azaltılması veya sabit tutulması olmak üzere üç tanedir. K_p, [-0.05, 0, 0.05]; K_i, [-0.01, 0, 0.01]; K_d ise [-0.005, 0, 0.005] sayıları ile artıp azalmaktadır.

E. Gerçek Zamanlı Deney ve Benzetim Çalışması

Q-öğrenme tabanlı optimal PID kontrolcü tasarımı uygulamasında, PID kontrolcünün her parametresi için bir Q-öğrenme kontrolcüsü ve bir Q-tablosu oluşturulmuştur. Dolayısıyla sistemden alınan durum bilgisi ve aksiyon üç adet Q-öğrenme kontrolcüsüne gitmekte ve hepsi ayrı PID katsayısının ayarlamasını yapmaktadır. Sistem gerçek-zamanlı bir sistemdir ve sistemin durumu hata değeri ve hatanın değişim yönü bilgileri ile kestirilmektedir.

$$s = [f(e_{n)}, f(e_n - e_{n-1})] \tag{17}$$

Eşt. (17)'de $f(e_{n)}$, n anındaki hata sinyalinin ayrık gruplama yapılması ile elde edilen değere karşılık gelmektedir. $f(e_n - e_{n-1})$ fonksiyonu ise hatanın değişim yönünü ifade etmektedir.



Şekil 1. *Çalışma Diyagramı*

Şekil 1' de tasarlanan yapının çalışma diyagramı gösterilmiştir. Burada kullanılan DA makinesi hız kontrol sisteminin transfer fonksiyonu Eşt. (18)'de verilmiştir.

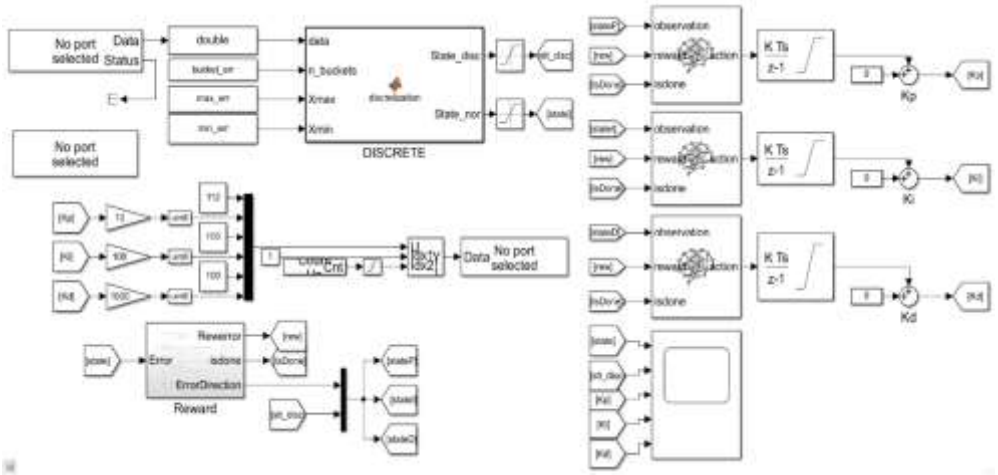
$$G(s) = \frac{K_T}{s^2(J_m L_a) + s(B_m L_a + J_m R_a) + (B_m R_a + K_T K_b)} \tag{18}$$

Çalışmamızda kontrol edilecek olan sistem olarak DC motor hız kontrolü sistemi seçilmiştir. Sistemin açık çevrim transfer fonksiyonu eşitlik **Hata! Başvuru kaynağı bulunamadı.**' de verilmiştir. Benzetim ç alışması için Matlab-Simulink ortamı tercih edilmiştir. Gerçek-zamanlı çalışma için bir adet DC motor deney düzeneği kullanılmıştır. DC motor deney düzeneğini sürmek için bir sürücü kart ve kontrol işlemlerini yerine getirecek olan ADUC841 geliştirme kartı kullanılmıştır. Q-öğrenme algoritması bilgisayar üzerinde çalıştırılmış ve gerçek-zamanlı olarak ADUC841 mikrodenetleyicisi ile seri

kanaldan haberleştirilmiştir. Benzetim çalışması ve gerçek zamanlı kontrol için Q-öğrenme algoritması ise Matlab ortamında Reinforcement Learning Toolbox kullanılarak gerçekleştirilmiştir.

(a) ***(b)*** ***(c)***

Şekil 2. (a) *Deney Düzeneği Kontrol Kartı,* ***(b)*** *DA Makinesi Deney Düzeneği,* ***(c)*** *Deney Düzeneği Sürücü Kartı.*

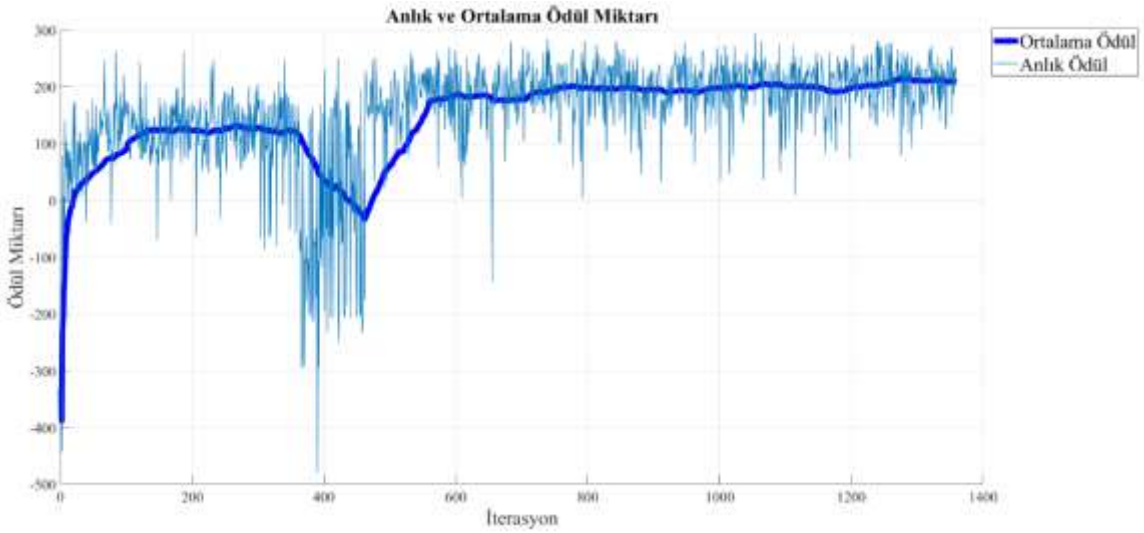


Şekil 3. *Gerçek Zamanlı Çalışma Matlab Simulink Ortamı.*

Şekil 2'de gerçek-zamanlı çalışma için kullanılmış olan düzeneğe ait görseller verilmiştir. Matlab-Simulink ortamında gerçek zamanlı çalışmanın benzetim çalışması yapılmış ve teorik sonuçlar benzetim ortamında doğrulanmıştır. Gerçek-zamanlı çalışmayı yapabilmek için Matlab-Simulink ortamında hazırlanan ve gerçek-zamanlı veri işleyen uygulamaya ait yapı Şekil 3'te verilmiştir.

III. BULGULAR VE TARTIŞMA

Yapılan benzetim çalışması ve gerçek-zamanlı deney sonuçları elde edilmiş ve Q-öğrenme algoritmasının kontrol işlemini gerçekleştirdiği gözlemlenmiştir. Eğitim süreci 1359 iterasyon sürmüştür. Eğitim sürecinin sonunda elde edilen Q-tabloları kullanılarak sistem çalıştırılmıştır. Gerçek-zamanlı sistem çalışırken Q-öğrenme algoritması 2 saniye boyunca çalıştırılmıştır.

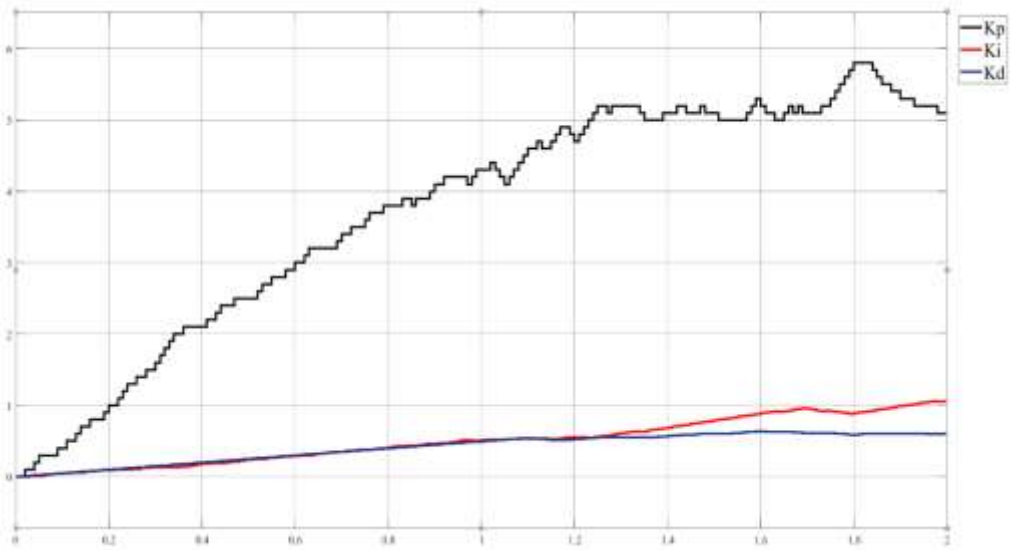


Şekil 4. *Eğitim Sürecindeki Ajanların Ortalama ve Anlık Ödül Miktarları.*

Ajanların eğitim süreci boyunca aldığı ortalama ve anlık ödül miktarları Şekil 4'te verilmiştir. Ajanlar eğitim sürecinin başında epsilon değerine göre keşif yapmaya yönelik aksiyonlar seçtiği için ortalama ödül miktarı düşüktür. İterasyon sayısı 100 civarlarındayken ortalama ödül miktarı 100 seviyelerine çıkmıştır 100. İterasyonda ajan %60 olasılıkla keşif yapmaya yönelik eylem seçmektedir. 350 iterasyona kadar bu seviyelerde kalan ortalama ödül miktarı 450 iterasyon civarlarında -30 seviyesine düşmüştür. 450. İterasyonda ajan %10 olasılık ile keşif yapmaktadır. Ardından yaklaşık 550. iterasyonda 180 civarlarına çıkmış ve eğitim sürecinin sonuna kadar 200 seviyelerine yükselmiştir. Eğitimin sonlarına doğru epsilon değerine göre ajan %0.1 olasılık ile keşif yapmaya yönelik aksiyon seçmektedir.

Şekil 5*. Gerçek-Zamanlı Çalışma Cevabı.*

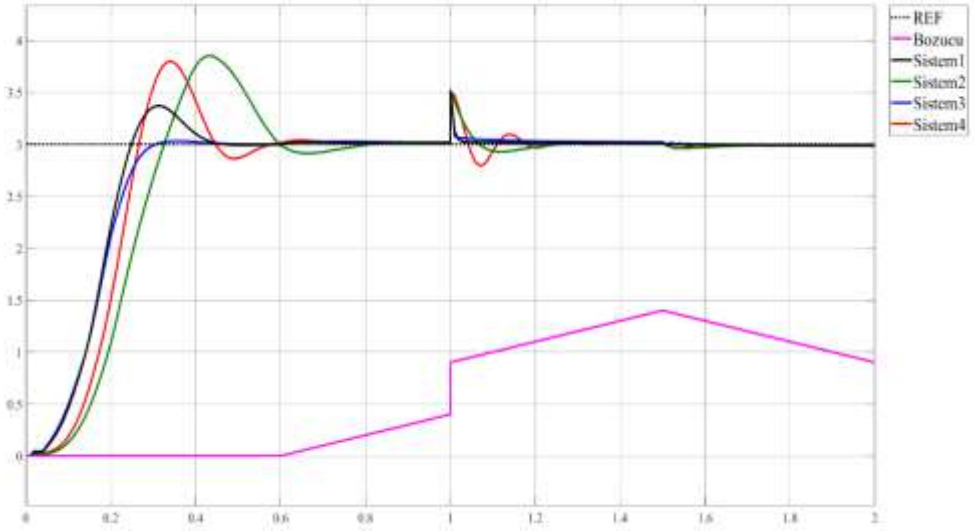
Şekil 5'te verilen sistem cevabı incelendiğinde çıkış; %7'lik bir aşımla yaklaşık 450 milisaniyede referansa yerleşmiştir.



Şekil 6. *PID Katsayılarının Değişimi.*

Şekil 6'da verilen K_p, K_i ve K_d parametrelerinin değişimi incelendiğinde, K_p katsayısı 5 değerine yaklaşırken; K_i katsayısı 1 değerine; K_d katsayısı ise 0.6 değerine yaklaşmıştır.

Eğitim sonrasında elde edilen Q-tabloları üzerinden sistem çalıştırılarak kontrolcünün performansı gözlemlenmiştir. Önerilen yöntem modelden bağımsız olduğundan dolayı kontrol edilen sistemin parametreleri değişse bile Q-öğrenme kontrolcüsü PID parametrelerini uygun değerlere ayarlayabilecektir.



Şekil 7*. Sistem Parametre Değişimlerinin Etkisi.*

Şekil 7'de görüldüğü üzere ajanlarımızın eğitiminin gerçekleştiği sistemin kazancı 0.98, zaman sabiti ise 120 milisaniyedir. Sistem1'de kazanç değişmezken zaman sabiti 60 milisaniye olarak seçilmiştir; Sistem1'in çıkışı %13'lük bir aşım ile 450 milisaniyede referansa ulaşmıştır. Sistem2'nin kazancı 1.2 ve zaman sabiti 250 milisaniyedir,%26'lük bir aşımla 800 milisaniyede referansa ulaşmıştır. Sistem3'ün kazancı 1.5, zaman sabiti ise 120 milisaniyedir, %2'den daha düşük bir aşım ile 300 milisaniyede referansa ulaşmıştır. Sistem4 ise kökleri $s = -8.41$ ve $s = -41.58$' de olan ikinci dereceden bir sistemdir, %25'lik bir aşımla 580 milisaniyede referansa ulaşmıştır. Benzetim çalışması gerçekleştirilirken kontrol işareti sınırlandırılmıştır.

Tablo 1. *Sistem Parametreleri.*

Sistemin Modeli	Sistem	Sistem1	Sistem2	Sistem3	Sistem4
Sistemin Transfer Fonksiyonu	$\frac{0.98}{0.12s+1}$	$\frac{0.98}{0.06s+1}$	$\frac{0.98}{0.25s+1}$	$\frac{1.5}{0.12s+1}$	$\frac{400}{s^2+50s+1}$

IV. SONUÇ

Q-öğrenme tabanlı PID kontrolcüsü seçilen DA makinesi hız kontrolü sistemi üzerinde eğitilmiştir. Yaklaşık 950 iterasyon sonucunda öğrenme işlemi yeterli düzeye ulaşmıştır. Hem gerçek-zamanda hem de benzetim ortamında sistemi beklendiği gibi kontrol ettiği görülmüştür. Modelden bağımsız bir yöntem olması sebebiyle modeldeki parametre değişimleri olsa dahi kontrol işlemi gerçekleştirilmiştir. Model tabanlı kontrolcülerle kıyaslandığında model parametrelerindeki değişimlere karşı daha çok toleranslıdır. Model parametrelerindeki küçük değişimlerin performans üzerinde etkisi çok az olduğu görülürken, parametrelerdeki büyük değişimlerin ana modelin performansına yakın bir performans gösterdiği görülmüştür. Sistemden alınan durum bilgisi ve ödül fonksiyonunun seçimi kontrolcü üzerinde doğrudan etkiye sahiptir. Sistemden daha çok veri alınabildiği uygulamalarda daha iyi sonuçlar elde edilebilir. Sistemden daha çok veri alındığı uygulamalarda nöral ağ kullanılması uygundur. Nöral ağ yaklaşımı ile tasarlanan Q-öğrenme ajanları başarılı sonuçlar verecektir.

V. KAYNAKLAR

[1] R. S. Sutton and A. G. Barto, "An introduction to reinforcement learning," *Decis. Theory Model. Appl. Artif. Intell. Concepts Solut.*, pp. 63–80, 2011.

[2] M. L. Minsky, "Theory Of Neural-Analog Reinforcement Systems and Its Application To The Brain-Model Problem," Princeton University, Princeton, 1954.

[3] D. P. Bertsekas, *Dynamic Programming and Stochastic Control*. New York-London: Academic Press, 1976.

[4] R. Bellman, "The Theory of Dynamic Programming," *Bull. Am. Math. Soc.*, vol. 60, no. 6, pp. 503–515, 1954,.

[5] R. Bellman, "Dynamic programming and stochastic control processes," *Inf. Control*, vol. 1, no. 3, pp. 228–239, Sep. 1958.

[6] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn. 1992 83*, vol. 8, no. 3, pp. 279–

292, May 1992.

[7] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[8] Q. Shi, H. K. Lam, B. Xiao, and S. H. Tsai, “Adaptive PID controller based on Q-learning algorithm,” *CAAI Trans. Intell. Technol.*, vol. 3, no. 4, pp. 235–244, 2018.

[9] F. L. Lewis and D. Vrabie, “Adaptive dynamic programming for feedback control,” *Proc. 2009 7th Asian Control Conf. ASCC 2009*, pp. 1402–1409, 2009.

[10] B. P. Amiruddin and R. E. A. Kadir, “Ball and beam control using adaptive pid based on q-learning,” *Int. Conf. Electr. Eng. Comput. Sci. Informatics*, vol. 2020-Octob, no. October, pp. 203–208, 2020.

[11] M. Ali, A. Mujeeb, H. Ullah, and S. Zeb, “Reactive Power Optimization Using Feed Forward Neural Deep Reinforcement Learning Method : (Deep Reinforcement Learning DQN algorithm),” *2020 Asia Energy Electr. Eng. Symp. AEEES 2020*, pp. 497–501, May 2020.

[12] T. Tan, F. Bao, Y. Deng, A. Jin, Q. Dai, and J. Wang, “Cooperative deep reinforcement learning for large-scale traffic grid signal control,” *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2687–2700, Jun. 2020.

[13] Z. Guan and T. Yamamoto, “Design of a reinforcement learning PID controller,” *IEEJ Trans. Electr. Electron. Eng.*, 2021.

[14] I. Carlucho, M. De Paula, S. A. Villar, and G. G. Acosta, “Incremental Q-learning strategy for adaptive PID control of mobile robots,” *Expert Syst. Appl.*, vol. 80, pp. 183–199, Sep. 2017.

[15] I. Carlucho, M. De Paula, and G. G. Acosta, “An adaptive deep reinforcement learning approach for MIMO PID control of mobile robots,” *ISA Trans.*, vol. 102, pp. 280–294, Jul. 2020.

[16] M. Gheisarnejad and M. H. Khooban, “An Intelligent Non-Integer PID Controller-Based Deep Reinforcement Learning: Implementation and Experimental Results,” *IEEE Trans. Ind. Electron.*, vol. 68, no. 4, pp. 3609–3618, Apr. 2021.

[17] D. Lee, S. J. Lee, and S. C. Yim, “Reinforcement learning-based adaptive PID controller for DPS,” *Ocean Eng.*, vol. 216, Nov. 2020.

[18] X. song WANG, Y. hu CHENG, and W. SUN, “A Proposal of Adaptive PID Controller Based on Reinforcement Learning,” *J. China Univ. Min. Technol.*, vol. 17, no. 1, pp. 40–44, Mar. 2007.

[19] M. Ağralı, M. U. Soydemir, A. Gökçen, and S. Şahin, “Deep Reinforcement Learning Based Controller Design for Model of The Vertical Take-off and Landing System,” *Eur. J. Sci. Technol. Spec. Issue*, vol. 26, no. 26, pp. 358–363, 2021.

[20] A. Younesi and H. Shayeghi, “Q-Learning Based Supervisory PID Controller for Damping Frequency Oscillations in a Hybrid Mini/Micro-Grid,” *Iran. J. Electr. Electron. Eng.*, vol. 15, no. 1, pp. 126–141, Mar. 2019.

[21] C. Mu, K. Wang, S. Ma, Z. Chong, and Z. Ni, “Adaptive composite frequency control of power systems using reinforcement learning,” *CAAI Trans. Intell. Technol.*, May 2022.

[22] J. Khalid, M. A. M. Ramli, M. S. Khan, and T. Hidayat, “Efficient Load Frequency Control of Renewable Integrated Power System: A Twin Delayed DDPG-Based Deep Reinforcement

Learning Approach," *IEEE Access*, vol. 10, pp. 51561–51574, 2022.

[23] J. C. Hung and J. D. Hewlett, "PID control," *Control Mechatronics*, vol. 9, pp. 10.1-10.8, 2016.

[24] P. Lu, W. Huang, and J. Xiao, "Speed tracking of Brushless DC motor based on deep reinforcement learning and PID," *2021 7th Int. Conf. Cond. Monit. Mach. Non-Stationary Oper. C. 2021*, pp. 130–134, Jun. 2021.

[25] X. Y. Shang, T. Y. Ji, M. S. Li, P. Z. Wu, and Q. H. Wu, "Parameter optimization of PID controllers by reinforcement learning," *2013 5th Comput. Sci. Electron. Eng. Conf. CEEC 2013 - Conf. Proc.*, pp. 77–81, 2013.

[26] R. Mukhopadhyay, S. Bandyopadhyay, A. Sutradhar, and P. Chattopadhyay, "Performance Analysis of Deep Q Networks and Advantage Actor Critic Algorithms in Designing Reinforcement Learning-based Self-tuning PID Controllers," *2019 IEEE Bombay Sect. Signat. Conf. IBSSC 2019*, vol. 2019Januar, pp. 1–6, 2019.

[27] W. Yu and A. Perrusquía, *Human-Robot Interaction Control Using Reinforcement Learning*. 2021.

[28] C. J. C. H. Watkins, "Learning from delayed rewards. PhD thesis," *PhD thesis*. 1989.