MARKOV KARAR SÜRECİ İLE ASANSÖR KONTROLÜNÜN MODELLENMESİ

Cebrail Çiflikli[1], Emre Öner Tartan[2]

Kayseri Üniversitesi Meslek Yüksekokulu[1], Başkent Üniversitesi Teknik Bilimler Meslek Yüksekokulu[2]
cebrailc@erciyes.edu.tr[1], onertartan@gmail.com[2]

ÖZET

Bir Markov Karar Süreci, bir dinamik sistem kontrolünde, mevcut durumun yalnızca önceki duruma bağlı olarak modellenebildiği bir stokastik kontrol sürecidir. Markov Karar Süreçleri'nde amaç süreç boyunca elde edilecek ödülleri en yüksek seviyeye çıkarmaktır. Bu çalışmada asansör kontrol problemi Pekiştirmeli Öğrenme modelinde bir Markov Karar Süreci olarak ele alınmaktadır. Ödüller olarak kat çağrılarının bekleme zamanlarının negatif değerleri kullanılmaktadır. Bir başka ifadeyle bekleme zamanları ceza olarak yorumlanmakta ve dolayısıyla en yüksek ödül elde etme amacı, en düşük ceza elde etme olarak değerlendirilmektedir. Buna göre dinamik programlama ile elde edilen en iyi politika ile asansör kontrolü sağlanmaktadır. Bu yaklaşımda, klasik kat çağrılarının asansörlere atanması yaklaşımından farklı olarak asansör içinde bulunduğu duruma göre bir sonraki en iyi duruma geçişi sağlayacak en iyi hareket eylemine karar vermektedir. Elde edilen model tek asansörlü bir binada farklı kat sayıları için kontrol sürecine uyarlanmış ve simüle edilmiştir. Sonuçlar, asansör kontrolünün Markov Karar Süreci olarak modellenmesinin uygunluğunu göstermektedir. Bu yaklaşımın, bir Yapay Zeka yaklaşımı olan Pekiştirmeli Öğrenme'nin grup asansör sistemlerine uygulanmasında bir temel oluşturacağı öngörülmektedir.

1.GİRİŞ

"Optimal kontrol" terimi, 1950'li yıllarda, dinamik bir sistemin bir sistem ölçütünü en aza indirecek bir kontrolör tasarlama problemi olarak ele alınmıştır [1]. Bu soruna yaklaşımlardan biri 1950'lerin ortalarında Bellman tarafından, Hamilton ve Jacobi'nin 19. yüzyılda ortaya koydukları teori üzerine geliştirilmiştir [1]. Bu yaklaşım, Bellman denklemi olarak adlandırılan bir fonksiyonel denklemi tanımlamak için dinamik bir sistemin durumunu ve bir değer fonksiyonu veya "optimal getiri fonksiyonu" kavramlarını kullanır. Bu denklemi kullanarak optimal kontrol problemlerini çözmekte başvurulan yöntemler sınıfı dinamik programlama olarak bilinir hale gelmiştir [2]. Bellman ayrıca Markov Karar Süreci (MKS) olarak bilinen optimal kontrol probleminin ayrık stokastik versiyonunu tanıtmış, Ronald Howard MKS 'lerde optimal çözüm için politika yineleme yöntemini tasarlamıştır [3,4]. Sonuçta bu teorik altyapı, günümüzde yapay zeka alanındaki makine öğrenmesi yaklaşımlarından biri olan Pekiştirmeli Öğrenme'nin temelini oluşturmuştur [1]. Markov karar süreçleri, zaman planlaması, finans ve tıp gibi birçok farklı alandaki çeşitli pekiştirmeli öğrenme problemlerinde belirsizlik altında sıralı karar verme için matematiksel bir çerçeve sağlar [5]. Klasik tam toplamalı asansör kontrol sistemleri yolcuların gidecekleri kat bilgisini içermeyen dinamik bir sistemdir. Bu dinamik sistemin kontrolü Pekiştirmeli Öğrenme problemi olarak ele alınıp, bir MKS olarak modellenebilir.

2. MARKOV ZİNCİRİ ve MARKOV KARAR SÜRECİ

Markov süreci veya Markov zinciri, her olayın olasılığının yalnızca önceki olayda elde edilen duruma bağlı olduğu olası olaylar dizisini tanımlayan stokastik bir modeldir[6]. 20. yüzyıl başlarında A. Markov'un sunduğu Markov Zinciri ekonomiden, kimyaya, dilbilimden termodinamiğe birçok bağımlı olasılık probleminde çözümünde başarı ile uygulanan bir tekniktir [6]. Markov süreçlerinin temel özelliği, belirli bir zaman dilimi içinde çeşitli durumlarda bulunmanın ve bir durumdan diğer duruma geçişin olasılıklarının göz önüne alınmasıdır. Markov süreçleri ileride ortaya çıkması olası durumların gerçekleşme olasılıklarının, geçmiş durumlara dayalı değil, yalnızca şu anki duruma dayalı olarak tahmin edilebildiği süreçlerdir. Sistemdeki durumlar arası geçişi modellemedeki bu hafızasızlık ya da geçmişe ihtiyaç duymama özelliğine Markov özelliği denir. St t anındaki durum olmak üzere, (1) ile verilen bir sonraki zaman adımına geçiş olasılığı denklemi bu özelliği ifade eder:

$$P[S_{t+1} \mid S_t] = P[S_{t+1} \mid S_1, \ldots, S_t] \tag{1}$$

Bir ayrık zamanlı Markov zincirinde, her zaman adımında, bu stokastik sürecin ilerleyeceği durumlara geçiş için bir olasılık dağılımı söz konusudur. Bu dağılım, her durumda bir ya da daha fazla eylem içinden yapılacak seçim ile farklı durumlara geçiş olasılıklarını versin. Seçilen eyleme göre geçilen durumda bir ödül kazanımının varsayıldığı ve süreç boyunca en büyük toplam ödülün elde edilmeye çalışıldığı bir Markov süreci Markov karar süreci olarak ifade edilir. Pekiştirmeli Öğrenme'de Markov karar sürecini açıklayabilmek için öncelikle sistemi modellemede kullanılacak terminolojiyi açıklamak gerekir.

3.TERMİNOLOJİ

3.1 Etmen

Pekiştirmeli Öğrenme'de ajan(agent) olarak da isimlendirilen etmen, eylemleri ile çevre ile etkileşime giren, çevrenin durumuna ilişkin bilgi elde eden ve bir amacı olan bir birimdir. Etmene örnek olarak amacı çöpleri temizlemek olan bir robot, Go oyununda amacı rakibi yenmek olan olan oyuncu verilebilir [7].

3.2 Çevre

Etmenin etkileşimde olduğu ortamdır. Etmen çevreyi değiştiremez ancak eylemleri ile durumu değiştirebilir. Örneğin asansör probleminde asansör sayısı, binadaki kat sayısı çevreye ilişkin unsurlardır ve değiştirilemez. Ancak asansör kontrolörü (etmen) aşağı gitme kararı ile mevcut durumu değiştirebilir.

3.3 Durum

Durum, problemde etmen ve çevrenin ele alınış biçimine bağlı olarak tanımlanan, etmenin içinde bulunduğu hali ifade eder. Örneğin, çöp temizleyen robot örneğinde durum robotun konumu ve şarj durumu ile ifade edilebilir. Etmenin elde edebildiği bilgi ve çevre tanımına göre farklı durum tanımları olabilir. Bir t zaman adımındaki durum S_t, mümkün durumlar kümesi S'in bir elemanıdır.

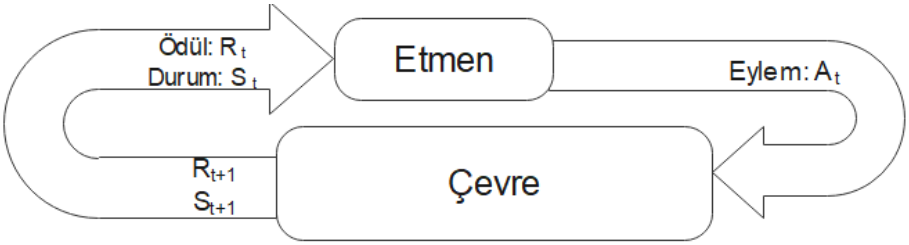
3.4 Eylem

Eylem, etmenin içinde bulunduğu durumda, çevre ile etkileşime neden olan yapabileceği seçimlerden her biridir. Bir S_t durumunda mümkün eylemler kümesi $A(S_t)$, t zaman adımında bu kümeden seçimi olan eylem A_t olarak temsil edilir. Go oyununda oyuncunun içinde bulunduğu durumda yapabileceği her hamle bir eylemdir [7].

3.5 Politika

Politika, her zaman adımında, içinde bulunulan durum ile o durumda yapılması mümkün eylemler arasındaki olasılık eşleştirmeleridir. Bir başka ifadeyle içinde bulunulan durumda hangi olasılıkla hangi eylemin seçileceğini belirleyen mekanizmadır. Politika t zaman adımında π_t olarak temsil edilir. İzlenen politikaya göre t zaman anında bulunulan s durumunda ($S_t = s$) seçilen eylemin a olması ($A_t = a$) olasılığı $\pi_t (s|a)$ biçiminde ifade edilir.

Bir etmen ve çevre t = 0, 1, 2, 3… ayrık zaman adımları dizisinde etkileşim halindedir. Her t zaman adımında etmen çevrenin durumunun bir temsilini, $S_t \in S$, elde eder. Etmen içinde bulunduğu durumda mümkün olan eylemler kümesi $A(S_t)$ içinden izlediği politikaya (π_t) göre bir eylem $A_t \in A(S_t)$ seçer. Bu eylemin sonucunda, bir sonraki zaman adımında sayısal bir ödül $R_t \in R$ elde eder ve sonraki zaman adımındaki duruma S_{t+1} geçiş yapmış olur. Bu süreç Şekil 1'de gösterilmektedir. Pekiştirmeli Öğrenme'de ayrık MKS olarak modellenen süreçlerde, etmen deneme ve yanılma ile en iyi politikayı öğrenmeye çalışır. Özellikle birçok bilgisayar oyununda Pekiştirmeli Öğrenme'nin etkinliği gösterilmiştir [8].



Şekil 1. Çift katlı asansörlü sistemin genel görünümü

Özetle bir MKS;

- S : durumlar kümesi
- A: eylemler kümesi
- P: S x S x A → [0,1] sistem dinamikleri olarak da nitelendirilebilen geçiş olasılıklarını
- R: S x A x S → R ödül dağılımını

  içerir.

Bir etmenin s durumunda a eylemini yapınca s′ durumuna geçme olasılığı P(s′|s,a) olarak ifade edilir. Buna göre belirli bir durumdan her olası eylem için her olası duruma geçiş olasılıklarının toplamı 1 olacaktır.

$$\forall s \in S \forall a \in A \sum_{s' \in S} P(s' \mid s, a) = 1 \qquad (2)$$

Bir etmenin s durumunda a eylemini yapınca s′ durumuna geçişinde kazanacağı ödül R(s,a,s′) olarak ifade edilir. Hesaplamalarda genellikle, s durumunda a eylemi yapılınca beklenen ödül değeri R(s,a) ifadesi kullanılmaktadır.

$$R(s,a) = \sum_{s'} R(s,a,s') * P(s' \mid s,a) \tag{3}$$

Etmenin amacı süreç boyunca toplam ödül miktarını veren değer fonksiyonu değerini en büyük yapmaktır.

$$V = \sum_{i=1}^{\infty} r_i \tag{4}$$

Bazı problemlerde toplam ödül hesaplanmasında bir sonraki adımdaki ödül ile gelecekte elde edilecek ödül aynı ağırlıkta değerlendirilmeyebilir. Örneğin finansta kısa zamanda elde edilecek kâra, uzun vadede elde edilecek kâra göre daha fazla önem atfedilebilir. Bu durumda değer fonksiyonunda azaltımlı toplam ödül kullanılır:

$$\begin{aligned} V &= \sum_{i=1}^{\infty} \gamma^{i-1} r_i \\ V &= r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots + \gamma^{i-1} r + \ldots \\ V &= r_1 + \gamma(r_2 + \gamma(r_3 + \ldots)) \end{aligned} \tag{5}$$

4. MARKOV KARAR SÜRECİNDE DURUM TEMSİLİ

Bu kısımda tek asansör kontrolü için Markov Karar Süreci olarak modellenmesi ele alınmaktadır. Bir MKS; çevredeki mümkün durumlar ve bu durumlarda gerçekleştirilebilecek eylemler ve bu eylemlerin doğuracağı durum geçişlerinin olasılıkları ile tanımlanır.

Asansör kontrol probleminde, binadaki kat sayısı, asansör sayısı, asansör kinematiği çevreye ilişkin unsurlardır. Bu çalışmada öncelik MKS'nin asansör kontrolüne uyarlanabilirliğini ve çalışmasını göstermek adına basit bir durum tanımı yapılmıştır. Bir durum; asansörün bulunduğu kat, asansörün hareket yönü, aşağı ve yukarı yönlü kat çağrılaı dizileri ile temsil edilmektedir. Buna göre bir s durumu için kabin katı s(kabin), kabin yönü s(yön), yukarı yönlü çağrılar s(yukarı) ve aşağı yönlü çağrılar s(aşağı) ile ifade edilmektedir. Dolayısıyla bir s_t durumu {s_t (kabin), s_t (yön), s_t (yukarı), s_t (aşağı)} dizisi ile temsil edilmektedir. Burada s_t (yukarı) ve s_t (aşağı) elemanlarını Çizelge 1'de gösterildiği gibi bit dizileridir. Bu iki dizinin birleşimine s_t(kç) olarak temsil edilebilir.

Çizelge 1'de 3 katlı bir binadaki durumlardan, asansörün 1.(giriş) katta iken olası durumlar gösterilmiştir. Burada kabin yönü; yukarı doğru ise 1, aşağı doğru ise 2 ve kabin duruyorsa 0 ile temsil edilmektedir. Kabin katı s[kat] 1 ile en üst kat KS arasında değerler alabilmektedir. Kat çağrıları KÇ ise s[yukarı] ve s[aşağı] olarak yukarı yönlü çağrılar ve aşağı yönlü çağrılar olarak temsil edilmektedir. Kat çağrılarında o katta bir çağrı varsa 1 yoksa 0 olarak temsil edilmektedir.

Çizelge 1. Markov Karar Süreci'nde durum temsili

Kabin katı	Kabin yönü	Çağrılar			
		KÇ			
S[kabin]	S[yön]	S[yukarı]		S[aşağı]	
		K1	K2	K2	K3
1	0	0	0	0	0
1	0	0	0	0	1
1	0	0	0	1	0
1	0	0	0	1	1
1	0	0	1	0	0
1	0	0	1	0	1
1	0	0	1	1	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	0	1
1	0	1	0	1	0
1	0	1	0	1	1
1	0	1	1	0	0
1	0	1	1	0	1
1	0	1	1	1	0
1	0	1	1	1	1
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

5- DURUM UZAYI, EYLEM UZAYI ve ÖDÜL

En üst katta yukarı yönlü çağrı, en alt katta ise aşağı yönlü çağrı olamayacaktır. Dolayısıyla Çizelge 1'de s(yukarı) altında en üst kat için bir sütun, s(aşağı) altında en alt kat için bir sütun yoktur. Sonuçta kat çağrıları dizisi s(kç)'nin alabileceği değer 2^{KS-2} olacaktır.

Yine ilk ve son katlarda kabin yönü sırasıyla aşağı ve yukarı olamaz. Dolayısıyla s(yön) sütunu, kabin son katta ise 1 değerini, kabin ilk katta ise 2 değerini alamayacaktır. Asansör yönünün 0(duruyor), 1 (yukarı) veya 2 (aşağı) değerlerinden her birini alabileceği (yukarı, aşağı ve duruyor olabileceği) kat sayısı KS-2'dir. Sadece 0 ya da 1 değerini alabileceği (giriş katı) ve sadece 0 ya da 2 değerini alabileceği (en üst kat) kat sayıları birerdir. Özetle asansör yönü KS-2 ara katta üç değerden (0,1,2) birini alabilir, 2 uç katta (en alt ve en üst katlarda) ise iki değerden (0,1 veya 0,2) birini alabilir.

Sonuçta KS katlı bir binada ara katlar için olası durumların sayısı $3x(KS-2)x2^{KS-2}$ olmaktadır. En alt ve en üst katlardaki olası durumların toplam sayısı $2x2x2^{KS-2}$ olacaktır. Kat sayısı ile artan olası durum sayıları ile süreci modelleyebilmek için durumlar sınıflandırıp, her sınıf altında gruplandırma yapılmaktır. Böylelikle, her durum için ayrı eylem kümesi tanımlamak yerine eylem kümeleri bellekten de tasarruf ederek daha hızlı ve kolay biçimde kullanılabilir. Buna göre bir durum 4 sınıftaki toplam 10 durumdan birine ait olacaktır:

$$S = \{S_{boş} \cup S_{aşağı} \cup S_{yukarı} \cup S_{uç}\} \quad (6)$$

5.1 $S_{boş}$ Durum Sınıfı ve $A_{boş}$ Eylem Sınıfı

Durumları sınıflandırmada birinci sınıf, asansörün boş olması halindeki $S_{boş}$ sınıfına ait durum gruplarını içermektedir. Asansör boş iken bir durum 4 olası durum grubundan birine aittir:

$$S_{boş} = \begin{cases} S\_boş^{kç\_yok} = \left\{ s \in S : s[yön] = 0,\ s[yukarı][kabin] = 0,\ \begin{matrix} s[aşağı][kabin] = 0 \\ s[kabin] \neq 1, \\ s[kabin] \neq KS \end{matrix} \right\} \\ S\_boş^{kç\_aşağı} = \left\{ s \in S : s[yön] = 0,\ s[yukarı][kabin] = 0,\ s[aşağı)[kabin] = 1 \right\} \\ \qquad\qquad s[kabin] \neq KS \\ S\_boş^{kç\_yukarı} = \left\{ s \in S : s[yön] = 0,\ s[yukarı][kabin] = 1,\ s[aşağı][kabin] = 0 \right\} \\ \qquad\qquad s[kabin] \neq 1 \\ S\_boş^{kç\_yukarı\_aşağı} = \{ s \in S : s[yön] = 0,\ s[yukarı][kabin] = 1,\ s[aşağı][kabin] = 1 \} \end{cases} \quad (7)$$

Burada s[yukarı][kabin] yukarı çağrılar dizisinin kabin katındaki değerini, bir başka ifadeyle kabinin bulunduğu katta yukarı çağrı bulunup bulunmamasını temsil etmektedir. Bu durum gruplarına ait eylem kümeleri ise şöyledir:

$$A_{boş} = \begin{cases} A\_boş^{kç\_yok} = \{ yukarı\_hareket, aşağı\_hareket, bekle \} \\ A\_boş^{kç\_aşağı} = \{ yukarı\_hareket, aşağı\_hareket, aşağı\_yolcusunu\_al, \\ \qquad\qquad bekle \} \\ A\_boş^{kç\_yukarı} = \{ yukarı\_hareket, aşağı\_hareket, yukarı\_yolcusunu\_al, \\ \qquad\qquad bekle \} \\ A\_boş^{kç\_yukarı\_aşağı} = \left\{ \begin{matrix} yukarı\_hareket, aşağı\_hareket, \\ yukarı\_yolcusunu\_al, aşağı\_yolcusunu\_al, bekle \end{matrix} \right\} \end{cases} \quad (8)$$

5.2 $S_{yukarı}$ Durum Sınıfı ve $A_{yukarı}$ Eylem Sınıfı

Durumları sınıflandırmada ikinci sınıf $S_{yukarı}$, asansörün yukarı gidiyor olması halindeki durum gruplarını içermektedir:

$$S_{yukarı} = \begin{cases} yukarı^{kç\_yukarı\_yok} = \{ s \in S : s[yön] = 1 \quad ve \quad s[yukarı][kabin] = 0 \} \\ yukarı^{kç\_yukarı} = \{ s \in S : s[yön] = 1 \quad ve \quad s[yukarı][kabin] = 1 \} \end{cases} \quad (9)$$

Bu durum gruplarına ait eylem kümeleri ise şöyledir:

$$A = \begin{cases} A\_yukarı^{kç\_yukarı\_yok} = \{ yukarı\_hareket \} \\ A\_yukarı^{kç\_yukarı} = \{ yukarı\_hareket, yukarı\_yolcusunu\_al \} \end{cases} \quad (10)$$

5.3 $S_{aşağı}$ Durum Sınıfı ve $A_{aşağı}$ Eylem Sınıfı

Durumları sınıflandırmada üçüncü sınıf $S_{aşağı}$, asansörün aşağı gidiyor olması halindeki durum gruplarını içermektedir:

$$S_{aşağı} = \begin{cases} aşağı^{kç\_aşağı\_yok} = \{s \in S : s[yön] = -1 \ \ ve \ \ \ s[aşağı] = 0\} \\ aşağı^{kç\_aşağı} = \{s \in S : s[yön] = -1 \ \ ve \ \ \ s[aşağı] = 1\} \end{cases} \quad (11)$$

Bu durum gruplarına ait eylem kümeleri ise şöyledir:

$$A = \begin{cases} A\_aşağı^{kç\_aşağı\_yok} = \{aşağı\_hareket\} \\ A\_aşağı^{kç\_aşağı} = \{aşağ\_hareket, aşağı\_yolcusunu\_al\} \end{cases} \quad (12)$$

5.4 $S_{uç}$ Durum Sınıfı ve $A_{uç}$ Eylem Sınıfı

Durumları sınıflandırmada dördüncü sınıf $S_{uç}$, asansörün uç katlarda olması halindeki durum gruplarını içermektedir:

$$S_{uç} = \begin{cases} S\_uç^{ilk\_kat} = \{s \in S : s[kabin] = 1\} \\ S\_uç^{son\_kat} = \{s \in S : s[kabin] = KS\} \end{cases} \quad (13)$$

Bu durum gruplarına ait eylem kümeleri ise şöyledir:

$$A_{uç} = \begin{cases} A\_uç^{ilk} = \{yukarı\_hareket, yukarı\_yolcusunu\_al\} \\ A\_uç^{son} = \{aşağı\_hareket, aşağı\_yolcusunu\_al\} \end{cases} \quad (14)$$

Belirli dönemlerde sonlanan (epizodik) süreçler, bir sonlandırıcı duruma veya maksimum zaman adımına kadar devam eder. Dolayısıyla n son zaman adımı olmak üzere, (4) ve (5) denklemleri ile verilen toplam ödül hesaplamasında i=1 zaman adımından n. zaman adımına veya sonlandırıcı(terminal) duruma geçilene kadar ödüller hesaba katılır. Daha sonra yeni bir MKS başlar. Asansör dağıtım probleminde bir dönemi sonlandıran durum, bekleyen tüm kat çağrılarına hizmet edilmesi, bir başka ifadeyle bekleyen kat çağrısının kalmaması olarak tanımlanabilir.

5.5 Ödül

Pekiştirmeli Öğrenmenin uygulamalarında değer fonksiyonundaki ödüller çok çeşitli biçimlerde tanımlanabilir. Bir ödül sinyali pozitif veya negatif olabilir. Bu etmenin bir eylemin iyi, kötü veya nötr olup olmadığını ölçmesine izin verir. Örneğin Super Mario oyununda sağa doğru ilerleme pozitif bir ödül iken oyuncunun çukura düşmesi negatif ödüldür [9]. Burada negatif ödül ceza olarak da nitelendirilebilir. Her halde amaç değer fonksiyonunu (toplam ödülü) maksimize etmektir. Sadece cezalardan (negatif ödüllerden) oluşan bir oyunda amaç en az cezayı almaktır. Asansör kontrol probleminde ise her zaman adımında bekleyen kat çağrılarının bu adımda beklediği toplam bekleme süresi ceza olarak tanımlanmaktadır. Bir başka ifadeyle bir durumdan bir başka duruma geçişte, negatif toplam bekleme süresi ödüldür. Bu durumda, KÇ bekleyen kat çağrısı sayısı olmak üzere daha önce (3) ile verilen ödül ifadesi şöyle tanımlanabilir:

$$R(s,a) = -KÇ.\Delta t \tag{15}$$

Bir Markov Karar Süreci'nde bir s durumunda atılabilecek en uygun adımı temsil eden optimal politika π* ile gösterilir.

$$\pi^*(s) = \arg\max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^*(s')] \tag{16}$$

Burada T(s,a,s') s durumunda a eylemi yapıldığında s' durumuna geçiş olasılığını, R(s,a,s') ise bu geçiş sonucunda elde edilecek ödülü, V* ise yeni s' durumun benzer şekilde elde dilmiş değer fonksiyonunu göstermektedir. γise azaltma parametresi daha sonraki adımlardaki ödüllerin etkisinin daha yakın zamandaki ödüllere göre daha az olması ve ödül toplamının sonsuza ıraksamaması için, özellikle sonlu olmayan süreçlerde kullanılan ve 0.9 gibi 1'e yakın seçilen bir parametredir. Ele alındığı biçimiyle asansör probleminde, bir eylem sonucunda bir durumdan geçilecek durum deterministiktir (olasılıklar 1'dir). Dolayısıyla optimal politika sadeleştirilerek ifade edilebilir:

$$\pi^*(s) = \arg\max_a \sum_{s'} [-KÇ.\Delta t + \gamma V^*(s')] \tag{17}$$

Dinamik programlama yaklaşımıyla iteratif biçimde optimal politikaya yakınsanması ile her durumda seçilmesi gereken optimal eylemler bulunabilir. Böylelikle eğer-ise biçimindeki kurallara dayanan bir yaklaşım yerine bekleme zamanını azaltmayı amaçlayan bir kontrol yöntemi elde edilebilir ve simüle edilebilir [10].

5. SONUÇ

MKS sunduğu soyut ve esnek matematiksel çerçeve birçok farklı alanda, Markov özelliği taşıyan süreçler içeren çeşitli problemlerde uygulanabilmektedir. Bu çalışmada, tek asansörün olduğu tam toplamalı asansör kontrol sisteminde ele alınmış ve asansör kontrol problemi kat çağrılarının bekleme zamanını en aza indirmek amaçlanarak MKS olarak modellenmiştir. Bu kapsamda durum-eylem kümesi ve ödül tanımlamaları yapılmıştır. Durum ve eylem kümeleri, her durumda seçilebilecek her eylemle geçilebilecek sonraki durumu içermektedir. Gelecek çalışmada her durumda, değer fonksiyonunu maksimize edecek optimal eylemleri veren optimal politikanın, dinamik programlama ile bulunması önerilmektedir. Bu çalışmanın kural tabanlı asansör kontrol yöntemlerine alternatif olarak, kendi kendine öğrenen, adaptif Pekiştirmeli Öğrenme tabanlı kontrol yöntemleri geliştirilmesine zemin hazırlayacağı öngörülmektedir.

KAYNAKLAR

[1] **Sutton, R. S., Barto, A. G.** 2015. Introduction to Reinforcement Learning. MIT Press/Bradford Books, Cambridge, MA, s. 17.

[2] **Bellman, R. E.** 1957. Dynamic Programming. Princeton University Press, Princeton.

[3] **Bellman, R. E.** 1957. A Markov decision process. Journal of Mathematical Mechanics, 6: s. 679–684.

[4] **Howard, R.** 1960. Dynamic Programming and Markov Processes. MIT Press, Cambridge, MA.

[5] **Steimle, L. N. , Kaufman, D. L. Denton, B. T.**. 2021. Multi-model Markov decision processes, IISE Transactions, 53:10, s. 1124-1139.

[6] **Gagniuc, P. A.** 2017. Markov Chains: From Theory to Implementation and Experimentation. USA, NJ: John Wiley & Sons. s. 1–235.

[7] **Silver, D., Huang, A., Maddison, C. vd.** 2016. Mastering the game of Go with deep neural networks and tree search. Nature 529, s. 484–489.

[8] **Shao, K., Tang, Z., Zhu, Y., Li, N., Zhao, D.** 2019. A Survey of Deep Reinforcement Learning in Video Games. https://arxiv.org/pdf/1912.10944.pdf

[9] **Karakovskiy, S., Togelius, J.** 2012. The mario ai benchmark and competitions, Computational Intelligence and AI in Games, IEEE Transactions on, cilt. 4, sayı. 1, s 55–67.

[10] **Çiflikli C., Tartan E. Ö. 2018.** Grup Asansör Sistemlerinin Simülasyonu. Mühendis ve Makina Dergisi, 59(693), s 1-18.