



**T.C.
ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**SÜRÜCÜ DAVRANIŞLARININ MODELLENMESİNE
BİLİŞSEL MİMARİ YAKLAŞIMI**

Gökhan YENİKAYA

**DOKTORA TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

BURSA 2006


T.C.
ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ


SÜRÜCÜ DAVRANIŞLARININ MODELLENMESİNE
BİLİŞSEL MİMARİ YAKLAŞIMI

Gökhan YENİKAYA


DOKTORA TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI


Bu Tez 26/12/2006 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.


Yrd. Doç. Dr. Halil YEŞİLÇİMEN
Danışman


Doç. Dr. Eldar MUSAYEV


Doç. Dr. İbrahim GÜNEY


Doç. Dr. Nejat YUMUŞAK


Yrd. Doç. Dr. Osman H.
KOÇAL

ÖZET

Geçen yüzyılın sonlarına doğru ulaşım teknolojilerindeki hızlı gelişmeler, can veya mal kaybına yol açan trafik kazaları ve zaman kaybına yol açan trafik sıkışıklıkları şeklinde sorunları da beraberinde getirmiştir. İstatistiksel olarak bakıldığı zaman bu sorunların büyük oranda insan kaynaklı hatalardan ortaya çıktığı görülmektedir. İnsandan kaynaklı sorunların çözümünde, otomobil kullanma esnasında sürücünün üzerinden yükü alacak sürücü yardımcı sistemlerinin oluşturulması fikri araştırmacılar tarafından büyük ilgi görmüş ve bu konuda yoğun çalışmalar gerçekleştirilmiştir.

Sürücü yardımcı sistemlerinin oluşturulmasında en önemli aşama, sürücü davranışının modellenmesi aşamasıdır. Bilişsel mimari insan davranışının modellenmesi için gelecek vaat eden en önemli seçeneklerden biri durumundadır, fakat araç sürme problemine uygulanmasıyla ilgili literatür oldukça kısıtlıdır. Bu tez çalışmasının bir katkısı, araç sürme davranışının modellenmesinde, bilişsel mimariyi temel alan bir yöntemin önerilmesidir. Önerilen mimarinin çekirdeğinde, deneyimler yoluyla sürücü davranışlarını öğrenmeye imkan sağlayan pekiştirmeli öğrenme yöntemi, pekiştirmeli öğrenme yöntemi olarak da Q-Öğrenme yöntemi kullanılmıştır. Literatür taramasında standart Q-Öğrenme tekniklerinin insan davranışlarının modellenmesinde kullanılmasının, insan davranışlarının sürekli olmasıyla durum uzayının büyüdüğü, içinde bulunduğu durumun dinamik olduğu ve insan davranışları için bir pekiştirme fonksiyonu oluşturmanın oldukça zor olduğu durumlarda problemlere sahip olduğu görülmüştür. Bu tez çalışmasının bir diğer katkısı, süreklilik ve büyük durum uzayı probleminin çözümü için alt görevlerin otomatik olarak belirlenmesi tekniğinin geliştirilmesi, dinamik ortamlarda çalışabilmenin sağlanması için bir hafıza yapısı eklenmesi ve pekiştirme fonksiyonunun gözlenen sürücüden otomatik olarak çıkarılmasının sağlanması, ve bu yöntemleri içeren geliştirilmiş Q-Öğrenme yöntemini kullanan bilişsel mimari temelli bir modelin oluşturulmasıdır.

Anahtar Kelimeler: Araç Sürücü Modelleme, Modelleme, Bilişsel Mimari, Pekiştirmeli Öğrenme, Q-Öğrenme, Seçenekler, Otomatik Alt Görev Belirleme

ABSTRACT

Near the end of the last century, the rapid developments in transportation technologies lead the traffic accidents which cause the loss of human or assets and increase the traffic congestions which cause the loss of time. Statistically, it is obvious that the most important source of these problems are human-centered. To solve these problems, researchers have focused on driver assistance systems which support human while driving and increasingly vigorous efforts have been put in this area.

The most important stage in developing driver assistance systems is modelling the driver's behaviour. Cognitive architecture is a very promising approach in modelling human behavior. But for modelling drivers' behavior based cognitive architecture, there exists very limited literature. In this thesis, for modelling drivers' behavior, a method based on cognitive architecture is proposed. In the core of the proposed method, reinforcement learning which allows learning with experience, is used. Q-Learning method as the reinforcement learning method is implemented in the architecture. Standard Q-Learning methods suffer from large state sets rising from continual operation, from the dynamic operating environment, and from providing a reinforcement function which seems as a hard process for human behavior. In this thesis, for solving the continual operation problem a novel sub-goal discovery algorithm is introduced, for tackling dynamic environments usage of history memory is implemented and a method for automatically extracting the reinforcement function is inserted. Using the modified Q-Learning structure, a model based on cognitive architecture is developed. The proposed method is tested experimentally and good results have been obtained.

Keywords: Drivers' Behavior Modelling, Cognitive Architecture, Reinforcement Learning, Q-Learning, Options, Automatically Subgoal Discovery, Reverse Reinforcement Learning,

İÇİNDEKİLER DİZİNİ

1. GİRİŞ	1
1.1. Sürücü Yardımcı Sistemleri	1
1.1.1. Araç Sürme Davranışı – Yapay Zeka	3
1.1.2. Araç Sürme Davranışı – Bilişsel Mimari	5
1.2. Tez İçeriği	6
2. KAYNAK ARAŞTIRMASI	7
2.1. Sürücü Yardımcı Sistemleri	7
2.2. Sürücü Davranışı Modelleme	9
2.3. Bilişsel Mimari	11
2.3.1. Bilişsel Mimaride Öğrenme	13
2.3.2. Q-Öğrenme	14
2.3.3. Alt Görevler ve Uzayın parçalanması	15
3. MATERYAL VE YÖNTEM	17
3.1. Bilişsel Mimari	17
3.1.1. Bilişsel Mimari Yapısı	17
3.1.2. Bilişsel Mimari Alt Sistemleri	21
3.1.2.1. Karar Verme Alt Sistemi	21
3.1.2.2. Karar Destek Alt Sistemi	23
3.1.3. Bilişsel Mimaride Öğrenme	24
3.1.3.1. Üstten Alta Öğrenme	25
3.1.3.2. Alttan Üste Öğrenme	26
3.2. Pekiştirmeli Öğrenme	32
3.2.1. Markov Özelliği	34
3.2.2. Markov Karar Verme Süreci	35
3.2.3. Değer Fonksiyonları	35
3.2.4. En Uygun Değer Fonksiyonu	37
3.2.5. Çözümleme Yöntemleri	39
3.2.5.1. Dinamik Programlama	39
3.2.5.2. Monte Carlo Yöntemi	39
3.2.5.3. Geçici Farklar Yöntemi ve Q-Öğrenme	40

3.2.6. Genelleştirme ve Fonksiyon Yakınlaştırıcı	42
3.2.6.1. Geri Yayılım Ağları	43
3.2.6.2. Geri Yayılım Ağları ve Q-Öğrenme	46
3.2.6.3. Öğreticili Q-Öğrenme	49
3.3. Problem Tanımı	51
3.4. Bilişsel Mimari ve Önerilen Çözüm Yolları	53
3.4.1. Markov Özelliğine Sahip Olmayan Durumlarda Pekiştirmeli Öğrenme	53
3.4.2. Süreklilik – Büyük Durum Uzayı, Durum Uzayının Parçalanması ve Seçenekler Yapısı	54
3.4.2.1. SMDP ve Q-Öğrenme	57
3.4.2.2. Alt Hedeflerin Kullanımı ve Gridworld Ortamı Deneyi	58
3.4.3. Seçenekler ve Isıracı Aralıklardan Alt Görev Belirleme Yöntemi	63
3.4.3.1. Alt Görev Sınırlarının Otomatik Olarak Belirlenmesi	64
3.4.3.2. Isıracı Durumların Belirlenmesi	64
3.4.3.3. Algoritma	65
3.4.3.4. Isıracılık ve Geçiş Durumları	66
3.4.4. Pekiştirme Fonksiyonu ve Çıraklık Dönemi Eğitimi	67
3.5. Eylem Seçimi – Faydalanma – Araştırma	69
3.5.1. İki Seviyenin Çıkışının Entegre Edilmesi	70
3.6. Model Kurulumu	72
3.6.1. Öğreticili Q-Öğrenme	74
3.6.2. Kendi Kendine Öğrenme	81
3.7. Deneysel Çalışma ve Sonuçlar	82
3.7.1. Deneysel Çalışma	82
3.7.2. Algısal Girişler	84
3.7.2.1. Algılayıcılar	84
3.7.2.2. Ayırıklaştırma	88
3.7.3. Pekiştirme Fonksiyonunun Elde Edilmesi	93

3.7.4. Alt Görevlerin Otomatik Belirlenmesi	98
3.7.5. Eylem Seçimi ve Güncelleme	102
3.7.5.1. Anlık Koşul	102
3.7.5.2. Alt Görevin Seçimi	103
3.7.5.3. Q-Değerlerinin Belirlenmesi	106
3.7.5.4. Alt Görev - Eylem Seçimi	107
3.7.5.5. Güncelleme	110
3.7.5.6. Sonlandırma	118
3.7.6. Deneysel Çalışma Sonuçları	119
3.7.6.1. Deney I	119
3.7.6.2. Deney II	122
3.7.6.3. Deney Sonuçlarının Değerlendirilmesi	129
4. ARAŞTIRMA SONUÇLARI ve TARTIŞMA	132
5. KAYNAKLAR	139
6. TEŞEKKÜR	147
7. ÖZGEÇMİŞ	148

SİMGELER DİZİNİ

s	Anlık durum
s'	Bir sonraki durum
r	Anlık pekiştirme
a	Eylem
C	Koşul
$PM_a(C)$	C koşulunda a eyleminin uygulandığı ve sonucun olumlu olduğu çakışma sayısı
$NM_a(C)$	C koşulunda a eyleminin uygulandığı ve sonucun olumsuz olduğu çakışma sayısı
γ	İndirim faktörü
$sta = sta + 1$	İstatistik, $PM_a(C)$ veya $NM_a(C)$
$IG(A, B)$	Aynı eyleme giden A kuralı ile B kuralı arasındaki bilgi kazanç ölçümü
$IG(C, all)$	C koşulu ile kayıtlı diğer tüm kurallar arasındaki bilgi kazanç ölçümü
$Q(s, a)$	s durumunda a eylemini uygulamanın Q değeri
p_{re}	Kuralın ne olasılıkla seçileceğini belirleyen olasılık değişkeni
d_r	Eylem kuralını tutabilmek için gereken en düşük girişle çakışma sıklığı
π	Davranış biçimi
R_t	t anındaki toplam pekiştirme
t	Zaman adımı
$P''_{ss'}$	Verilen bir s durumu ve a eylemi için her olası bir sonraki s' durumu
$R''_{ss'}$	verilen herhangi bir s durumu ve a eylemi için, bir sonraki s' durumu ile birlikte bir sonraki adımda ödülün beklenen değeri
$V^\pi(s)$	s durumunda bulunmanın değeri
$Q^\pi(s, a)$	π davranış biçimi doğrultusunda s durumunda a eylemini uygulamanın Q değeri

π^*	En uygun davranış biçimi
V^*	En uygun değer
Q^*	En uygun Q değeri
α	Öğrenme oranı
$\bar{\theta}$	Parametre fonksiyonu
$\Delta w_{i \rightarrow j}$	i düğümünden j düğümüne giden bağlantının ağ ağırlığı farkı
o_i	j düğümünün çıkışı
B_j	j düğümünün çıkış hatası
d_z	İstenen çıkış
ΔQ_a	Eylem gerçekleştirilmeden önce ve gerçekleştirildikten sonraki Q değeri farkı
β	Seçenekler için bitiş durumu
I	Seçenekler için başlangıç durumu
$\langle I, \pi, \beta \rangle$	Seçenekler için başlangıç durumu, davranış biçimi, bitiş durumu üçlüsü
$\mu(s_i, \cdot)$	s_i durumunda seçenek seçme olasılığı
O_s	Seçenek
h	Geçmiş bilgisi
$\varepsilon(o, h, t)$	t anında s durumunda başlatılan π davranış biçiminin olayı
$KL(P, Q)$	P ve Q olasılıklarının yönlendirilmiş Kullback-Leibler ölçümü
$SKL(P, Q)$	P ve Q olasılıklarının simetrik Kullback-Leibler ölçümü
ϕ	Özellik vektörü
μ	Durum beklentisi
w	Ağırlık vektörü
c_7	Sabit
c_8	Sabit
$Kazanç_j$	j kuralını uygulamanın kazancı
U_j^T	j kuralını uygulamanın faydası

μ_E	Gözlener sürücünün durum beklentisi
τ	Boltzmann Sıcaklığı

KISALTMALAR

MDP	Markov Karar Verme Süreci
SMDP	Yarı Markov Karar Verme Süreci

ŞEKİLLER DİZİNİ

Şekil 3.1.	Alt Seviye Üst Seviye Yapısı, Kurallar durum kümeleri ve düğümler	20
Şekil 3.2.	Bilişsel Mimari dört alt sistemi	21
Şekil 3.3.	Pekiştirmeli Öğrenme	33
Şekil 3.4.	Davranış Biçimi, V ve Q fonksiyonları	36
Şekil 3.5.	Geri Yayılım Ağı	44
Şekil 3.6.	Sigmoid Fonksiyonu	45
Şekil 3.7.	Hata Grafiği	45
Şekil 3.8.	Yarar ağı taslağı. Kalın çizgiler vektörleri, ince çizgiler skalar sinyalleri işaret eder.	46
Şekil 3.9.	Geri yayılım ağı - Q - öğrenme bileşimi	47
Şekil 3.10.	Q-değeri ve Geri Yayılım Ağları	47
Şekil 3.11.	Her eylem için ayrı bir ağ	49
Şekil 3.12.	Pencere – Q Mimari	54
Şekil 3.13.	Gridworld ortamı	58
Şekil 3.14.	Yöntemin uygulanmasıyla belirlenen yol	59
Şekil 3.15.	Çözümleme sonunda her hücrenin yani ortamın her durumunun oluşan değeri	59
Şekil 3.16.	Hedefe varmak için her denemede bilişsel elemanın kaç hücreden geçtiği, yani her denemede çıkıştan hedefe kaç adımda vardığı	60
Şekil 3.17.	Seçenekler yapısı çerçevesinde problem, iki pekiştirmeli öğrenme problemi şekline dönüştürülmüştür.	61
Şekil 3.18.	Yöntemin hedefe varmak için her denemede bilişsel elemanın kaç hücreden geçtiği	62
Şekil 3.19.	Dört ısrarcı üst seviye davranışına sahip bir durum dizisi.	64
Şekil 3.20.	Bilişsel Eleman Eylem Seçimi ve güncelleme	72
Şekil 3.21.	Öğrenme ve Güncelleme Blok Şeması	73
Şekil 3.22.	Önerilen mimarinin blok şeması	75
Şekil 3.23.	Önerilen mimari eylem seçimi ve güncelleme	75
Şekil 3.24.	Geri Yayılım ağlarıyla eylem seçme	78

Şekil 3.25.	Sistem Genel Yapısı	82
Şekil 3.26.	Algılayıcılardan gelen bilgilerin yazılıma aktarılması	83
Şekil 3.27.	Ön Kamera	85
Şekil 3.28.	Sol Ayna Kamerası	85
Şekil 3.29.	Arka Kamera	85
Şekil 3.30.	Ön Panel Kamerası	86
Şekil 3.31.	Tam Arkamızdaki Araç ile aramızdaki mesafe(pixel olarak&ters orantılı)	87
Şekil 3.32.	Sol Arkamızdaki araç ile aramızdaki mesafe(piksel olarak&ters orantılı)	87
Şekil 3.33.	Direksiyon Pozisyonu. Düz konum 74. piksel ve oynamalar piksel olarak belirtilmiştir(Doğru Orantılı)	87
Şekil 3.34.	Aracımızın hızı.(km/s olarak)	87
Şekil 3.35.	Her saniyedeki hız değişimleri (km/s olarak)	88
Şekil 3.36.	Her saniyedeki direksiyonun yönündeki değişim(piksel olarak)	88
Şekil 3.37.	Arkamızdaki aracın bize göre ayrıklaştırılmış uzaklığı.	88
Şekil 3.38.	Önümüzdeki aracın bize göre ayrıklaştırılmış uzaklığı. 0 – Çok yakın, ..., 8 – çok uzak	89
Şekil 3.39.	Sol şeritte ve arkamızdaki aracın bize göre ayrıklaştırılmış uzaklığı. 0 – Çok yakın, ..., 8 – çok uzak	89
Şekil 3.40.	Sol şeritte ve önümüzdeki aracın bize göre ayrıklaştırılmış uzaklığı. 0 – Çok yakın, ..., 8 – çok uzak	89
Şekil 3.41.	Sağ şeritte ve arkamızdaki aracın bize göre ayrıklaştırılmış uzaklığı. 0 – Çok yakın, ..., 8 – çok uzak	90
Şekil 3.42.	Sağ şeritte ve önümüzdeki aracın bize göre ayrıklaştırılmış uzaklığı. 0 – Çok yakın, ..., 8 – çok uzak	90
Şekil 3.43.	Ayrıklaştırılmış hız bilgisi. 0 – Çok yavaş, ..., 8 – çok hızlı	90
Şekil 3.44.	İçinde bulunduğumuz şerit. 0 – En Sol, 1 – Orta, 2 – En sağ	91
Şekil 3.45.	Gaz pedalına basmanın ayrıklaştırılmış şiddeti. 0 – Basma, ..., 4 – Kuvvetli Bas	91
Şekil 3.46.	Fren pedalına basmanın ayrıklaştırılmış şiddeti. 0 – Basma,	91

	..., 4 – Kuvvetli Bas	
Şekil 3.47.	Direksiyonu sağa doğru çevirmenin ayrıklaştırılmış şiddeti. 0 – Çevirme, ..., 3 –Çok Çevir	92
Şekil 3.48.	Direksiyonu sola doğru çevirmenin ayrıklaştırılmış şiddeti. 0 – Çevirme, ..., 3 –Çok Çevir	92
Şekil 3.49.	1047 kayıtlı durum dizisinin alt görev dizisi şeklinde ifadesi	102
Şekil 3.50.	Global geri yayılım ağı	103
Şekil 3.51.	Şerit değiştirme alt görev ağı	104
Şekil 3.52.	Maksimum hıza erişme alt görev ağı	105
Şekil 3.53.	Takip Mesafesine ulaşma alt görev ağı	105
Şekil 3.54	Eylem tavsiyesi ile gerçekleşen eylem arasındaki doğruluk oranı	120
Şekil 3.55.	Kademesiz inceleme	122
Şekil 3.56.	Alt görev seçimi doğruluk oranları	123
Şekil 3.57.	Eylem tavsiyesi ile gerçekleşen eylemin birebir çakıştığı durum için doğruluk oranları	124
Şekil 3.58.	Gerçekleşen eylemle ± 1 fark kümesi içinde bulunan eylem tavsiyelerinin doğru olarak kabul edildiği durumda doğruluk oranları	125
Şekil 3.59.	Kademesiz inceleme	126
Şekil 3.60.	Eylem tavsiyesi ile gerçekleşen eylemin birebir çakıştığı durum için doğruluk oranları	127
Şekil 3.61.	Gerçekleşen eylemle ± 1 fark kümesi içinde bulunan eylem tavsiyelerinin doğru olarak kabul edildiği durumda doğruluk oranları	128
Şekil 3.62.	Kademesiz inceleme	129

ÇİZELGELER DİZİNİ

Çizelge 3.1.	Dinamik Programlama İşletim Prosedürü	39
Çizelge 3.2.	Monte Carlo Yöntemi İşletim Prosedürü	40
Çizelge 3.3.	Geçici Farklar Yöntemi İşletim Prosedürü	41
Çizelge 3.4.	Q-Öğrenme İşletim Prosedürü	41
Çizelge 3.5.	Q-değerleri tablo yapısı	42
Çizelge 3.6.	Geri Yayılım Ağıyla Q-Öğrenme İşletim Prosedürü	48
Çizelge 3.7.	Geri Pekiştirmeli Öğrenme İşletim Prosedürü	69
Çizelge 3.8.	μ_E vektörü	94
Çizelge 3.9.	ω vektörü	97
Çizelge 3.10.	Her durum için ısrarcılık puanları matrisi	98
Çizelge 3.11.	Alt görev oluşumları	99
Çizelge 3.12.	Her bir alt görev için ağ çıkışları	108
Çizelge 3.13.	Şerit değiştirme alt görevi dizisi	109
Çizelge 3.14.	Üst Seviye Kural Deposu	112
Çizelge 3.15.	Üst Seviye Kural Deposu	114
Çizelge 3.16.	Durum değişkeni varyasyonları	115
Çizelge 3.17.	Eksiltilmiş durum değişkenli koşul olasılıkları	118

1. GİRİŞ

Son yıllarda teknolojiadaki gelişmeler ulaşım konusunda da önemli atılımlara yol açmış, teknolojiyle birlikte hızla artan trafik sorunlarının önüne geçebilmek için karayolu ulaşım ağındaki taşıtların, trafik içerisinde sürücülere yardımcı olabilmesi amacıyla aktif bir rol alması yaklaşımı gündeme gelmiştir. Bu yaklaşım otomobil sürücü yardımcı sistemlerinin ortaya çıkmasına yol açmıştır. Otomobil sürücü yardımcı sistemlerinin geliştirilmesi için sürücü davranışlarının modellenmesi en önemli aşamalardan birisidir. Bilişsel Mimari ise, insan davranışlarının modellenenebilmesi için gelecek vaat eden önemli seçeneklerden biridir, fakat bu mimarinin araç sürme olayına uygulanması oldukça yeni bir konudur ve bu konuyla ilgili literatür henüz oldukça kısıtlıdır.

1.1. Sürücü Yardımcı Sistemleri

Etkili ve güvenli bir karayolu ulaşım ağı, toplumun sosyal ve ekonomik sağlığı açısından hayati önem taşımaktadır. Trafik ile ilgili istatistiklere baktığımızda çok dramatik sonuçlar karşımıza çıkmaktadır. Dünyaya baktığımızda her yıl 1.2 milyondan fazla insan trafik kazalarında ölmekte ve bu ölümlerin %70'i gelişmekte olan ülkelerde olmaktadır (WHO 2004). Kentsel üretimin 1/3'ü sıkışık trafikte yolculuk gecikmesi sebebiyle kaybedilmektedir. Kentlerde pek çok düşük gelirli ailenin gelirlerinin %20'sinden fazlası ulaşım harcanmaktadır (Mandıracıoğlu ve ark. 1997). Bu tür istatistikler, doğal olarak, tüm dünyada karayolu ulaşım sistemlerinin iyileştirilmesi çalışmalarına hız vermiştir. Karayolu altyapısı ile ilgili çalışmalar hem çok pahalıdır hem de ciddi bir rahatlama sağlamaktan uzaktır. Üstelik karayolu altyapısının yapısından öte, sürücü kaynaklı problemler de trafik sıkışıklıklarının ve kazalarının oluşmasında önemli pay sahibidir. Sürücü hatalarının ana kaynağı olarak, sürücülerin araç kullanımı esnasında etkisi altında buldukları fiziksel ve zihinsel yükler gösterilebilir. Bu yüklerin etkisi altında sürücülerin dikkat yönetiminde (Boer ve ark. 1998) bir takım bozulmalar oluşmakta ve bu bozulmalar, sürücülerini hata yapmaya zorlamaktadır. Sürücülerin araç kullanma ile ilgili hata yapma olasılığını azaltabilmek amacıyla son yıllarda pek çok çalışma yapılmış, bunun sonucunda taşıt kullanma esnasında sürücülere yardımcı olabilecek sürücü

yardımcı sistemleri geliştirilmiştir. Bu yardımcı sistemlerin, belirli yükler altında sürücünün performansını olumlu yönde nasıl etkilediği, Takada ve Shimoyama'nın (2001) yaptığı deneysel çalışmalarda görülmektedir. Bu çalışmalarda, yaş ortalaması 39 olan 6 erkek sürücü üzerinde deneyler gerçekleştirilmiş, araç kullanırken gösterdikleri vücut tepkilerini, soluk alıp verme ve kalp atış sıklığını gözlemleyerek, sürücü yardımcı sistemlerinin varlığı durumunda sürücülerin çok daha rahat hareket ettiklerini deneysel olarak göstermişlerdir.

Özellikle son on yıl içerisinde, otomobil firmalarının başını çektiği araştırmaların önemli bir bölümünü, otomobillerin de bizzat sürme olayının içerisinde aktif rol alması, sürücüyü yönlendirerek hata yapmasını engellemeye çalışması, ve onun da ötesinde ileri bir hedef olarak otomobillerin tam bir otonomluk kazanarak sürüş olayının tamamını üzerine devralabilmesi gibi konular kapsamaktadır. Bu konuda önemli bir yaklaşım, karayolları ağı ve araçların da içinde bulunduğu tüm yapıyı kapsayan Akıllı Taşıt Karayolu Sisteminin (IVHS – Intelligent Vehicle Highway System) (Fenton 1994) geliştirilmesidir. Bu yapı hem karayollarına hem de araç içlerine sistem entegrasyonunu gerektirmektedir. Bu yapının bir alt seviyesi, sürücü yardımcı sistemlerinin sadece araç içlerine yerleştirilerek, sürücüyü yönlendirebilecek veya uyarabilecek yapının kurulmasıdır. Bu çalışmalar yaygın olarak Otomatik araç kontrol sistemleri (AVCS – Automatic Vehicle Control Systems) (Shladover 1993), Gelişmiş sürücü sistemleri (ADS - Advanced Driver Systems) (Gallet ve ark. 2000) veya Arttırılmış Kontrol Sistemleri (ACS – Augmented Control Systems) (Pentland ve Liu 1995) başlıkları altında toplanmaktadır. Bu sistemler, otomobilin yanı sıra uçaklar (Schell ve Dickmanns, 1994), gemiler, denizaltılar, uzay araştırma araçları ve çeşitli robot platformlarını da kapsamaktadır (Trivedi 1989, Brooks 1991, Blackburn ve Nguyen 1994). Bu tez çalışmasında otomobil sürme problemine odaklanılmıştır ancak bu problem için öne sürülen teknikler diğer problemler içinde geçerliliğini korumaktadır.

Sürücü yardımcı sistemleri genel olarak şerit takibi, araç takibi, kavşaklarda ilerleme, gidiş yolunu planlama ve diğer araç kontrol görevlerini içermektedir. Bu sistemlerin getirdiği pek çok avantaj vardır. Güvenlik

açısından, otomatik cihazlarda insana has olan yorgunluk, dikkat dağılması gibi sorunlarla karşılaşmayacağı ve otomatik cihazlar insanlara göre daha keskin bir algılama kapasitesine ve daha hızlı cevap olanağına sahip olacağı için önemli bir avantaj sağlanmaktadır. Ayrıca bu tür desteğe sahip araçların yaygınlaşmasıyla trafik akışında düzelmeler sağlanacaktır.

Tam anlamıyla otonom sayılabilecek otomobillerin kullanımı şu an için uzak bir gelecek gibi görünmektedir, ancak sürücülerin otomobili kullanma kabiliyetini arttıracak sürücü yardımcı sistemlerinin üzerinde, araştırmacıların iyice odaklanmış olduğu ortadadır. Araç sürme olayı, diğer insan davranışlarına benzer şekilde karmaşık, belirsizlikle karakterizedir ve dolayısıyla, otomatik kontrol teknikleriyle ve statik yapay zeka yaklaşımlarıyla modellenememektedir. Bu tür yaklaşımlarda araç kontrolünün matematiksel bir ifadesi ortaya konulsa bile, bu yaklaşımları kullanarak insan davranış yapısını gerçek anlamda modellemenin sağlanamayacağı ortaya çıkmıştır (Forbes 2002).

Diğer bir yandan araç sürme davranışının oldukça sınırlı bir yapıda olmasından dolayı sürücü davranışlarının zihinsel kaynaklarını ortaya çıkarabilmek için aracı kullanırken sergilediği tipik davranışları ele almak yeterli olabilir. Örneğin sürücünün yanında oturduğumuzu düşünelim, sürücünün o an için ne yapmak üzere olduğunu kestirebiliyor muyuz?, 'Sağa mı dönecek?', yoksa 'Duracak mı?', veya 'Önündeki aracı mı sollayacak?'. Bu tür sorulara çoğu kez cevap verilebilir, bu da bizi, sürücünün o an için yaptığı hareketlere bakarak bir sonraki adımdaki hareketinin ne olacağını kestirebileceğimize, üstelik yüksek bir doğruluk oranıyla kestirebileceğimize inandırmaktadır.

Bu aşamada insan sürücülerin nasıl eğitildiği ve performansının nasıl değerlendirildiğine bakmak faydalı olacaktır. 'Aracı nasıl döndürüyorsun?', 'Düzgün bir şekilde duruyor musun?' gibi bazı sürme değerlendirme kriterleri, kontrol doğruluğunu ölçmeyi içermektedir. Diğer kriterler de güvenli takip mesafesinin korunmasından kaza ihtimalinin en aza indirilmesine kadar pek çok testi içerir. Sürücü hedefe doğru ilerlerken artı puan kazanırken, hatalı hareketleri de eksi puanla cezalandırılır. Sonuçta değerlendiren, eğitilen kişiye bir toplam puan verir ve bu puan eğitilenin toplam performansını gösterir. Çok

dikkatli bir eğitmen, eğitilenin performansını her an için değerlendirir ve çeşitli senaryolar açısından güvenlik, rahatlık, yetenek ve trafik kurallarına uygunluk gibi kriterlere göre derecelendirir. Toplam derece, sürücünün başka bir sürücünün kararlarını nasıl taklit ettiğine göre değil, içinde bulunduğu ortam ve ortamın etkileri altında nasıl kararlar verdiği göre verilmektedir.

1.1.1. Araç Sürme Davranışı – Yapay Zeka

Sıralayabileceğimiz birkaç faktör, araç sürme olayını yapay zeka disiplini açısından oldukça çekici bir çalışma alanı haline getirmektedir.

- Zamana bağımlı olayların, dizilerin ve ilişkilerin çözümlenebilmesi için geçici akıl yürütme gereklidir.
- Doğruluğun sürekli kontrolü gereklidir, çünkü türetilmiş gerçekler zaman içinde değişebilir.
- Sistem içinde oluşan herhangi rasgele bir duruma karşılık, bir cevabın mutlaka hazır olması gerektiğinden, beklenmeyen bir olayın etkili ve hızlı bir şekilde ele alınabilmesi için gerçek zamanlı algılama ve eyleme dönüştürme gereklidir.
- Eksik veya belirsiz veri ile akıl yürütme işlemi gerçekleştirilebilmelidir. Ortam, kestirilemeyen olayları, birbirinden bağımsız hareket eden araçlar ve rasgele şekilde hareket eden yayalarıyla oldukça zor bir ortamdır.
- Araçları ve sürme durumlarını da içeren ortam kombinasyonlarının tamamını düşünebilmek olası olmadığından, uyumlu davranış biçiminin geliştirilmesi zorunludur.

Bu sebepleri düşündüğümüz zaman araç sürme olayının 'akıllı' algoritmalar açısından oldukça zengin bir test ortamı olduğunu görebiliriz.

Modelleme açısından bir avantaj, sürme görevinin kendi içinde kolayca parçalanabilir olmasıdır. Bir sürücü zamanının büyük bir kısmını şerit takibi, araç takibi veya şerit değiştirme gibi alt görevleri gerçekleştirerek geçirmektedir. Aracı kontrol etmekse genel olarak iki kısma ayrılabilir: Yatay bir bileşen dönme açısını kontrol ederken, dikey bileşen aracın hızını, gaz ve fren pedalları

aracılığıyla ayarlamaktadır. Bazı kritik durumlarda, yatay ve dikey kontrol dinamikleri birbirinden bağımsız değildir. Örneğin, gaza basılı iken ve araç ivmelenirken, direksiyonu sert bir şekilde çevirmek pek önerilebilecek bir durum değildir. Bununla birlikte, geçerli dönme ve hızlanma-yavaşlama kontrol kararları, neredeyse her zaman bağımsızdır. Yüksek seviyelerde dikey ve yatay kontroller ilişkilidir. Eğer bir şerit değişikliği gerçekleştirilecekse, dikey bileşende yatay bileşende hesaba katılmalıdır.

Parçalanabilirliğinin aksine, araç sürme olayını yapay zeka uygulamaları açısından zor bir problem haline getiren bir takım yönler de mevcuttur. Birincisi, gerçek hayattaki sürme olayının sürekliliğinden dolayı, sonsuz sayıda ilgili durum oluşacaktır. Çok büyük ve sürekli durum uzayları şu andaki öğrenme teknikleri açısından bir sorun teşkil etmektedir (Sutton ve Barto 1998). Durum, kontrolcünün veya bilişsel elemanın bulunduğu ortamın değişkenlerinin anlık değerleri olarak ifade edilebilir. Sürme olayında durum, aracın hızı, yönü ve ortamda bulunan diğer nesnelerin pozisyonları, hızları vb. değişkenleri içerebilir. Yalnızca bir şerit boyunca mevcut olan sonsuz sayıda yatay pozisyonu düşünürsek, ne kadar küçük olursa olsun pozisyondaki en ufak bir değişiklik en uygun davranışı etkileyebilecektir. Dahası, aksiyonlar da sürekli, ve onları ayrıştırmak yeterli gelmeyebilir. Bir şeridi takip ederken, dönüş açılarının ayrıştırılması şeridin ortasında ilerlerken salınımlara yol açabilir.

Başka bir zor problem de araç sürmenin bir başlangıç durumuna dönmesizin sürekli olarak devam etmesidir. Birçok öğrenme algoritması, etkileşimin denemeler olarak tabir edilen benzer dizilere ayrılabilirdiği görevlere uygulanabilmektedir. Bir sürücü düz bir yolda uzun bir süre ilerlese bile, virajlarda veya kavşaklarda ilerleyebilme kabiliyetinde bir azalma olmamalıdır. Coğrafi gerçeklere göre düz ve virajlı yolların herhangi bir sıralaması olmadığı için, ikisinde de ustalık gereklidir. (Uzun düz bir yolda ne kadar ilerlenirse ilerlensin, hiçbir şekilde düz yollarda uzman bir sürücü haline gelebilmek için virajlı yollardaki araba kullanabilme yeteneğinden fedakarlık edilemez.)

1.1.2. Araç Sürme Davranışı – Bilişsel Mimari

Bir kere öğrenildikten sonra, araç kullanma esnasındaki andan ana hareketler, uygulandıkça otomatik hale gelir. Aslında, oldukça karmaşık bir trafik ortamındaki bu kadar karmaşık insan davranışlarının böylesine otomatikleştiği başka bir deney ortamı bulmak zordur. Çok karmaşık görünmekle birlikte nihai olarak araç kullanma işi genel olarak belirli bir sırayı takip etmektedir. Bununla birlikte klasik AI sistemlerindeki statik bakış açısıyla yola çıkmak, ileriki aşamalarda sorunlara yol açacaktır, çünkü insanın zamanla değişen doğasını göz önüne alırsak bir algısal girişin dönüştüğü eylem veya çıkarım zamana göre, şartlara göre, hatta insanın ruhsal durumuna göre sürekli değişmektedir. Şu ana kadar üzerinde çalışılan AI sistemleri ve bağlantısal sistemlerin büyük kısmı hep statik bakış açısını yansıtmaktadır. Bilişsel mimariler insanın değişken doğasını yansıtması açısından bir devrim niteliğinde düşünülebilir.

Gerçek bir bilişsel mimari, tecrübesiyle öğrenmeye hazırdır, ve ikinci günde birinci günde olduğundan daha başarılıdır. Ne yaptığını ve niçin yaptığını açıklamaya hazırdır. Beklenmeyen durumlarla karşılaştığında, herhangi bir makineden daha başarılı bir şekilde mücadele edebilmelidir.

1.2. Tez İçeriği

Tez çalışmasının ikinci bölümünde araç sürücü yardımcı sistemleri, sürücü davranışlarının modellenmesi, bilişsel mimari ve pekiştirmeli öğrenme ile ilgili literatür özeti verilecektir. Ayrıca pekiştirmeli öğrenmenin varolan problemleri ve bu problemlerle ilgili çözümleri içeren yayınlar da ikinci bölümde yer alacaktır.

Üçüncü bölümde bilişsel mimari, pekiştirmeli öğrenme ve Q-öğrenmenin temelleri anlatılacak ve problem tanımı yapılacaktır. Daha sonra problemin çözümü için kullanılacak materyaller tanıtılacak ve araç sürücü davranışı için bilişsel mimari temelli bir yöntem önerilecektir. Önerilen model adım adım kurulacak ve deneysel çalışmanın nasıl gerçekleştirildiği anlatılacaktır. Üçüncü bölümün sonunda, kurulan modelle ilgili deneysel sonuçlar sunulmuştur, dördüncü bölümde ise elde edilen sonuçlarla ilgili tartışma ve yöntemin geleceği ile ilgili düşünceler yer almaktadır.

2. KAYNAK ARAŞTIRMASI

2.1. Sürücü Yardımcı Sistemleri

Günümüzde araçlarda yerleşik durumda çalışan ABS (Anti-skid Braking System), CC (Cruise Control) gibi pek çok kontrol tekniği mevcuttur. Örneğin, seyir kontrol (CC – Cruise Control) mekanizmasında yolun yapısı değişse bile aracın hızı her zaman sabit bir değerde tutulmaktadır. Fren uygulandığı zaman sistem, tekrar başlatılana kadar devreden çıkar. Sistem ayrıca sürücü tarafından belirlenen bir hıza erişmek için hızı arttırıp azaltabilme yeteneğine sahiptir. Son zamanlarda, araçlara daha fazla otonomluk sağlamakla ilgili çok sayıda teknik geliştirilmiştir. Otonom akıllı seyir kontrolü (AICC – Autonomous Intelligent Cruise Control) öndeki araçlarla arasındaki mesafeye göre hızını ayarlamaktadır. AICC tasarımları radar, lazer radar, infrared veya sonar algılama sistemleri gibi algılayıcıları kullanarak öndeki aracın bağlı hızını ve mesafesini belirler (NAHSC 1995). Bu özelliğin olumsuz bir tarafı, aracın önündeki araçtan başka ortamla ilgili hiçbir bilgiye sahip olmamasıdır.

Aracın tam otonom halde tasarlanması için yeterli olgunluğa henüz ulaşamamıştır (Fletcher ve ark. 2003). Tam otonomluktan çok, sürücü yardımcı sistemleri çalışmalarında odağın, özellikle ticari anlam taşıyan araştırma geliştirme çalışmalarında, daha küçük alt-sistemlerin oluşturulmasına doğru kaydığı gözlenmiştir. Bu alt-sistemler genel olarak şu alt başlıklar altında toplanabilir:

- Sürücü yorgunluğunun veya dikkat dağılmasının tespiti
- Yayaların veya kör noktaların kontrolü
- Şerit takibi esnasında sürücüye geri besleme veya uyarı sistemi
- Ortamdaki araçlarla aradaki mesafe kontrolü ve uyarı sistemi
- Yol bilgileri ile geri besleme, örneğin hız sınırları veya trafik işaretleri ile ilgili uyarı sistemi

Şerit ve yol takibiyle ilgili de pek çok çalışma mevcuttur (Lee 2002, An ve Harris 1996). Görüş sistemi, şeritlerin ve nesnelerin yakalanmasında

kullanılmaktadır. Şerit yakalama – şerit çizgilerinin modelinin ve kalınlığının doğru olduğu durum için – aracın şerit içindeki pozisyon ve yönünün tahmininde bulunmak için geliştirilmiştir (Taylor ve ark. 1999). Lee, aracın şerit üzerinde kalmasına yardımcı olacak bir sistem geliştirmiştir (Lee 2002). Önerilen sistem dinamik ve rasgele ortamlarda da başarı sağlamış ve şerit takibi konusunda güvenilir bir uyarı sistemini meydana getirmiştir.

Nesne yakalama algoritmaları ise, resimler üzerinde nesne ayırt etme teknikleri uygulanarak, görüş açısı içindeki diğer araçların boyutlarını, bağlı pozisyon ve hızlarını bularak raporlamayı amaçlamaktadır (Lützeler ve Dickmanns 1998). Resimler arasındaki ilinti bilgisi, hareketi belirlemede faydalıdır (Kruger ve ark. 1995) ve binokular stereopsis aracılığıyla üç boyutlu nesne yakalama, PATH StereoDrive projesinde (Malik ve ark. 1997) başarıyla kullanılmıştır. Edgar ve Haris (1996), kaza önleme amaçlı uyumlu bir sürücü modelini içeren akıllı sürücü uyarı sistemi için bir mimari geliştirmişlerdir. Bu çalışmada sürücü modelleme, yalnızca öndeki araca odaklanmıştır. Modelin girişi olarak gaz pedalı açılarının geçmiş değerleri, aracın hızı, öndeki araçla mesafe alınmış ve sistemin çıkışı olarak da o andaki gaz pedalı açısı seçilmiştir. Gerçek zamanlı sürüş verilerine yapay sinir ağları yöntemi uygulanmıştır.

Bilgisayarlı görüş sistemlerinin kullanımı gerçek dünya prototiplerinde önemli oranda gelecek vaat etmektedir (Bertozzi ve ark. 2000, Lee 2003). Münih Üniversitesinde geliştirilen VaMP otonom aracı Almanya'nın Münih şehrinden Danimarka'nın Odense şehrine 1600 km'lik yolun %95'ini kendi başına almıştır (Bertozzi ve ark. 2000). Bu çalışmadaki insan sürücünün katkısı, şerit değiştirme manevralarının başlatılması, hedef hızın ayarlanması, hedeflerin ve varış noktalarının dikkate alınması, ve tabii ki otomatik kontrolcünün hata yaparak yol dışına çıkmasını veya başka bir araca çarparak kaza yapmasını engellemektir.

Yalnızca trafik kazaları değil, trafik sıkışıklıklarına da çözüm getirmeyi amaçlayan bir takım çalışmalar mevcuttur. Berkeley üniversitesinde, Gelişmiş Karayolları için ortaklar (PATH – Partners for Advanced Transit and Highways) adlı bir grup, platooning adlı sistemi geliştirmişlerdir (Varaiya 1991), bu sistem

araçların yüksek hızlarda fakat çok düşük araçlar arası mesafeyle ilerleyebileceği şekilde tasarlanmıştır. Bu kontrol hem araç hem de yola ilave cihaz ve yazılım ilavesiyle gerçekleşmiştir. Trafik sıkışıklığı, araçların birbirine iyice yanaşarak yüksek hızlarda seyretmesi, hem de bunu yaparken güvenlikten taviz vermemesi sağlanarak giderilebilir. Platooning'de gerçekleştirilen çalışma dikkat çekicidir, çünkü araçlara çok küçük bir ilave yaparak verimin nasıl artırılabileceğini göstermektedir. Bununla birlikte, platoon'lar AICC ile aynı sorunları yaşamaktadır, çünkü belirli koşullarda çalışmak üzere tasarlanmıştır. Lygeros ve arkadaşları (1997), platooning problemini hiyerarşik hibrid bir kontrol problemi olarak yapılandırmış böylece belirli çalışma koşullarında güvenlik, konfor ve verimlilik gibi temel kriterleri garanti etmesini sağlamıştır. Hibrid sistem, sürekli ve ayrık dinamiklerin etkileşim halinde olduğu sistemdir. Hiyerarşide, alt seviye sistem performansının yerel yönleriyle uğraşırken, yüksek seviye, bilişsel elemanlar arasındaki kaynakları paylaşmak gibi daha global meselelerle uğraşmaktadır.

Buraya kadar değindiğimiz tüm çalışmalar sürüş ortamında algılamanın iç zorluklarına odaklanmıştır. Sürüş olayı kontrol-teorik tekniklerle gerçekleştirilmiştir. Sistem filtre edilmiş algılayıcı bilgisini alır ve önceden belirlenmiş bir seriyi izleyerek görevini tamamlar. Fakat, araç sürme olayını bütünsel olarak düşündüğümüzde matematiksel olarak ifade edilemeyecek kadar karmaşık, belirsizliğin yüksek olduğu ve zaman içinde değişen bir yapısı olduğunu görebiliriz. Bu araştırmacıların odağını, sürme olayını doğrudan matematiksel ifadeler ile açıklamak yerine, adaptif bir sürücü davranış modelinin oluşturulmasına doğru kaydırmış ve araştırmalar yapay zeka disiplini içinde geniş çapta yer bulmuştur.

2.2. Sürücü Davranışı Modelleme

Sürücü modellemeye yönelik araştırma-geliştirme çalışmaları uzun yıllardır devam etmektedir. Yapılan çalışmalarda araç sürme probleminin parçalarını modellemekten, problemin tamamını modellemeye kadar pek çok yöntem önerilmiş ve bir kısmı da prototip halinde geliştirilmiştir.

Shibabi ve Mourant (2001), sürücü davranışının modellenenilmesi için dört birimden oluşan bir model önermiştir. Bu dört birim: Algılama birimi, duygusal birim, karar verme birimi ve karar uygulama birimidir. Bu modelde algılama birimi, lokal ve global anlamda içinde bulunduğu durum hakkında bilgi toplamakta, duygu birimi duygusal faktörlerin diğer birimler üzerindeki etkisini belirlemekte, karar verme ünitesi ortamdan algıladığı bilgiler doğrultusunda bir karar vermekte, ve karar uygulama birimi ise alınan kararları eyleme dönüştürmektedir. Song, Delorme ve Vanderwerf (2000), bu modele benzer yine dört birimden oluşan farklı bir model önermiştir: Algılama birimi, Dikkat Yönetimi Birimi, Kaynak Yönetimi Birimi ve Uygulama Birimi.

Pentland ve Liu (1999) ve Shimoyama ve Liu (2000) ve Oliver ve Pentland'ın (2001) çok değişkenli gauss yöntemi, en yakın komşu yöntemi gibi klasik yöntemler kullanarak gerçekleştirdikleri çalışmalarda, klasik yöntemlerin taşıt kullanma görevlerini ayırt etmede başarılı olamadıklarını göstermişlerdir.

Mitroviç (1999) modelleme için çok katmanlı perseptron sinirsel ağ kullanmış, fakat eğitim kümesine dahil edilmeyen durumlarla karşılaştığında veya çok yakın olmayan bir geleceğin kestirilmesi gerektiğinde başarılı olmadığını deneylerle göstermiştir.

Pentland ve Liu (1999), insan davranışının alt seviye yapısının, Kalman Filtre gibi dinamik bir model ile ele alınmasını üst seviye davranışlarının ise Markov zinciriyle bir araya getirilmesi yaklaşımını, yani Markov Dinamik Modeliyle tanımlanması yaklaşımını ortaya koymuştur. Oliver ve Pentland (2001) ise bu modeli geliştirerek bağlantılı gizli Markov Modeli yöntemini önermiştir. Markov Modelleri, anlık durum bilgisinin, kendisinden önceki tüm durumlar hakkında yeterli bilgiyi sağlayacağı durumlar için geçerlidir. Fakat algısal girişim (Whitehead ve Ballard 1991) problemi, Markov modellerinin bu tür ortamlarda başarılı olamayacağını göstermektedir.

ALVINN projesi öğreticili öğrenme yöntemini kullanarak sürüş algoritmalarını geliştirme üzerine odaklanmıştır (Pomerlau, 1993). Öğreticili öğrenmede, öğrenciye doğru cevaplarıyla birlikte giriş örneklerinden oluşan bir eğitim kümesi sağlanır. Öğrencinin amacı, bu örnekleri kullanarak genelleştirme

yapmak ve daha önce görmediği girişlere karşılık çıkışları tahmin etmektir: ALVINN bir sürücüyü üç dakika boyunca izler ve daha sonra aynı temel manevraları kendisi gerçekleştirir. Eğitim örnekleri bir resimler kümesiyle karşılık gelen tekerlek dönüş açısidir. Öğreticili öğrenme algoritması olarak tek gizli katmanlı bir geri-yayımlı ağı kullanmıştır (Bishop, 1995). Sistem değişken koşullar ve farklı yol yüzeyleri için oldukça iyi çalışmaktadır. Bununla birlikte, genel olarak ortamdaki diğer nesnelere birlikte hareket etmede veya şerit değiştirme gibi görevlerde başarısız olmaktadır.

Sürücü yardımcı sistemleri için otonom cihazların geliştirilmesinde insan sürücülerden öğrenme mantıklı ve içgüdüsel bir yöntemdir (Amit ve Mataric 2002). Buradaki asıl mesele hemen hemen tüm durumlarda çalışabilecek kontrolcüler oluşturmaktır. Temel durumlar için gerekli etkinliğe erişilebilirken, asıl sorun pek karşılaşılmayan ve tehlikeli durumların öğrenilmesinde ortaya çıkmaktadır. Örneğin, aracın yolda kayarak yola dik bir konuma gelmesi, insan sürücüyü gözlemlerken pek karşılaşılabilecek bir durum değildir. Otonom bir araç bu tür durumlarda dahi başarıyla işin içinden çıkabilmelidir.

İnsanı doğrudan taklit eden kontrol yöntemi ise birkaç sebepten dolayı problemlidir (Forbes 2002). İnsan sürücünün, sistem tarafından gözlenemeyen kendi hedef ve kaygıları mevcuttur. Sistem, insan sürücüyü yıllarca gözleyerek örnekler toplasa bile genelleştirme kötü olacak ve sağlam bir temele oturmayacaktır. Performansı en yükseğe taşımakla ilgili bir düşünce olmayacak, görev yalnızca daha önce gözlenmiş örnekleri kopyalamak olacaktır. Ek olarak, insana ait kararlar yalnızca onu uygulayana göre en uygundur. Eğer herhangi bir zamanda başka bir insan, hazırdaki insan sürücünün yerine geçerse sonuçlar idealden çok uzak olacaktır.

2.3. Bilişsel Mimari

Şimdiye kadar değindiğimiz çalışmalarda önerilen yapılar deneysel olarak uygunluğunu kanıtlamıştır, fakat daha büyük bir sistem içinde veya daha geniş bir yapıda kullanıldığı zaman istenileni vermekten uzaktır. Değinen önerilerin kendilerine göre getirdikleri bir takım avantajları ve dezavantajları vardır. Bütün bu sistemlerin getirdiği avantajları kullanabileceğimiz, dezavantajlı

duruma geldikleri zaman ise kullanmaktan vazgeçeceğimiz daha geniş bir yapı kurmak ideal gibi görünmektedir. Bilişsel mimari olarak adlandırılan mimari, aslında bu amaca hizmet etmek için kurulan yapılardan birisidir. Çok eski bir kavram olmamasına rağmen bilişsel mimari ile ilgili pek çok çalışma mevcuttur.

Araç kullanma problemi bilişsel mimari açısından yeni bir uygulama alanıdır ve bu problemle ilgili çalışmalara dair literatür oldukça kısıtlıdır. Bilişsel mimari kullanılarak sürücü modellemeyle ilgili gerçekleştirilen ilk çalışmalardan birisi, kavşağa giren araç sürücüsünün modellenmesidir ve Aasman tarafından önerilmiştir (Aasman 1995). Daha sonra Salvucci, Boer ve Liu (2002), sürücü davranışlarını modellemede, algılama ve bilişsel işlemlerini sıralı görev şeklinde düzenleyerek dikkat yönetimi ile birleştiren, ACT-R adını verdikleri, bilişsel mimarinin kullanılmasını önermiştir.

Bilişsel Mimari, algısal ve bilişsel işlemleri, iyi seçilmiş ve test edilmiş parametrelerle ve sınırlamalarla bütünleştirebilen bir yapıdır. Bilişsel Mimari, parametre ve sınırlandırmaları aracılığıyla, unutma, dikkat verme veya hata yapma gibi bilişsel birçok özelliği kullanabilmektedir. Yapı ayrıca sürücünün algısal ve kontrol yapısıyla ilgili gözlenmiş verilerle karşılaştırma yaparak, kestirimler üretebilmektedir. Bu özellikleriyle mimari, sürücü davranışlarının teorik olarak anlaşılabilmesini ve sürücü duygularının matematiksel olarak ifade edilebilmesine olanak sağlar.

Bilişsel Mimari kullanımının başlıca iki avantajı vardır: İlki, bilişsel modeller, alt seviye algılama ve kontrol işlemleriyle, üst seviye bilişsel mekanizmaları ve karar verme mekanizmalarını birbirine çok etkili bir şekilde bağlayabilmektedir (Salvucci 2003). Diğer bir avantajı ise bilişsel modellerin, psikofiziksel modelleri de içerebilmeleridir. Bugüne kadar yapılan çalışmalarda, bilişsel mimariler tarafından çözülebilen problemler şu şekilde özetlenebilir:

- Kişisel farklılıkların ayırt edilebilmesi
- En uygun performansın elde edilebilmesi
- Dikkat ve dikkat kapasitesi yönetimi
- Çalışan hafıza yükünün ayarlanabilmesi

- Davranışlardaki serilik ve paralelliğin göz önünde bulundurulabilmesi
- Çalışma yükünün düzenlenebilmesi

Bilişsel mimariler bu faktörleri tek başlarına değerlendirebildikleri gibi, bunlar arasındaki etkileşimi de modelleyebilmektedir.

Sun (2003), oldukça kapsamlı bir çalışma gerçekleştirmiş ve CLARION modelini ortaya koymuşlardır. CLARION mimarisi, insanın öğrenme ve karar verme doğasını modelleyebilmek için temel olarak iki bölüme ayrılmıştır: Karar Verme Alt-Sistemleri ve Karar Destek Alt-Sistemleri. Karar Verme Alt-Sistemleri deneyimleme (veya öğrenme) ve karar verme işlemlerinin gerçekleştiği alt sistemlerdir ve temel olarak ikiye ayrılır: Eylem Merkezli Alt-Sistem ve Eylem Merkezli Olmayan Alt-Sistem (Schacter 1990, Moscovitch ve Umilta 1991). Karar Destek Alt Sistemleri ise bilişsel elemanın deneyimleme (öğrenme) veya karar vermesi için gereken destek sistemlerini kontrol eder ve temel olarak iki kısımdan oluşur: Duygusal ve Meta-Bilişsel Alt-Sistemler (Sun 2003). Bu destek sistemleri, sistemin dahili parametrelerinin düzenlenmesinden sorumludurlar. Deneysel olarak gerçekleştirilen çalışmalarında, sanal bir mayın tarlasında beş ayrı algılayıcıdan bilgi alarak ilerleyen bir insan modellenmektedir. Deneğin amacı enerji ve süre bitmeden ve hiçbir mayına çarpmadan belirli bir hedefe ulaşmaktır. Beş adet algılayıcı mayınların deneğe göre yönlerini, uzaklıklarını, enerji durumunu, hedefin uzaklığını ve kalan zamanı göstermektedir. Modelle ilgili ayrıntılı bilgi ve diğer modellerle olan karşılaştırması (Sun, Merrill ve Peterson 1998, 2000, 2001)' de detaylı olarak bulunabilir.

2.3.1. Bilişsel Mimaride Öğrenme

Öğrenme, bilişsel elemanın gözlenen sürecünün deneyimlerini kullanarak her durum-eylem çiftine bir değer atama ve karar verme ise durum-eylem çiftlerinin değerlerini göz önünde bulundurarak bir seçim yapmasının sağlanmasıdır. Durum-eylem çifti değerlerini göz önünde bulundurarak bir eylem seçme ve uygulama, bilişsel elemanın davranış biçimini oluşturmaktadır. En uygun davranış biçimine yakınsanabilmesi için gereken, en uygun durum-eylem çifti değerlerine ulaşılmasıdır, burada durum-eylem çiftlerine doğru değerler atamak büyük önem taşımaktadır. Fakat bu değer atama nasıl

yapılacaktır? Pekiştirmeli öğrenme, deneyimlerle öğrenme yapısı üzerine kurulmuş ve durumlara veya durum-aksiyon çiftlerine değer ataması yapan bir yöntemdir (Sutton ve Barto 1998). Pekiştirmeli öğrenme, robot davranışlarının düzenlenmesi, robotlar arası etkileşimin sağlanabilmesi ve çoklu robot futbol turnuvaları gibi robotlar arası etkileşim gerektiği görevlerde yoğun bir şekilde kullanım alanı bulmuştur.

2.3.1.1. Q-Öğrenme

Q-Öğrenme (Watkins 1989) pekiştirmeli öğrenmenin bilişsel elemanın içinde bulunduğu ortamla ilgili (ortamdan gelen anlık bilgiler dışında) sistem dinamiklerine sahip olmadan da (örn. durum geçiş olasılıkları dağılımları gibi) çalışabildiği bir tekniktir. Burada her eylem-durum çiftine bir Q-değeri atanmakta ve bu değerler eylem seçiminde kullanılmaktadır. Her durum-değer çifti için bu güncelleme çevrimiçi olarak yapılmakta ve güncellenmiş bilgiler bir tablo şeklinde tutulmaktadır. Burada ortaya bir problem çıkmaktadır; özellik vektörümüzün boyutunun büyük olmasından dolayı ortaya çıkacak durumların, ve durum-eylem çiftlerinin sayısı oldukça fazladır. Böyle bir tabloyu güncellemek için harcanan zaman ve de depolamak için gereken hafıza bu işin gerçekleşmesini şu an için imkansız kılmaktadır. Yapılan literatür taramasında bu sorunu aşmak için temel olarak üç yaklaşım önerildiği görülmüştür. Birinci yaklaşım, kontrol stratejilerinin sınırlı bir uzayını düşünmek ve bu uzayı aktör-kritik taslağı (Barto ve ark. 1983, Baxter ve Bartlett 2001) içinde doğrudan aramaktır (Moriarty ve ark. 1999). İkinci yaklaşım değer fonksiyonları için düşük seviyeli fonksiyon yakınlaştırıcılarının kullanılması tekniğidir (Bertsekas ve Tsitsiklis 1995). Üçüncü yaklaşım ise kontrol görevinin daha basit çeşitli alt görevlere bölünerek uzayın küçültülmesi ve hepsinin paralel ve bağımsız olarak öğrenilmesi yöntemidir (Dayan ve Hinton 1993, Dietterich 2000, Sutton ve ark. 1999, Barto ve Mahadevan 2003). Bu alandaki çalışmalar genel olarak hiyerarşik pekiştirmeli öğrenme, hiyerarşik parçalama, seçenekler, makro-eylemler ve geçici soyutlama gibi başlıklar altında toplanmaktadır. Başlıklar farklı olsa da hepsinde amaç aynıdır, düşük seviyeli eylemleri daha yüksek seviyeli akıl yürütmelerle soyutlamak, veya başka bir deyişle ele alınan görevleri alt-görevlere veya alt-hedeflere bölmektir. Bu yaklaşım öğrenme problemini iki

şekilde basitleştirir. Birincisi, her alt-görev kendini ilgilendiren durumları göz önünde bulunduracağı için durum uzayının etkili boyutu küçültülmüş olur. İkincisi, alt görevlerin öğrenilmesi çok daha hızlanmış olacağı için, toplam öğrenme süresi dikkate değer şekilde düşürülmüş olacaktır.

2.3.1.2. Alt Görevler ve Uzayın parçalanması

İnsan karar verme mekanizması, alt-görevler arasından, veya farklı bir deyişle belirli zaman dilimine yayılmış olan eylemler dizisi arasından seçim yapmayı içermektedir. Örneğin araç sürme olayını ele alırsak, birkaç farklı alt görev vardır ve tüm araç sürme olayı, aslında, bu alt görevlerin arka arkaya sıralanmasından oluşmaktadır. Bu alt görevler durumunu korumak olabilir, şerit değiştirmek olabilir, veya araç sollamak olabilir. Burada, geçici soyutlama, pekiştirmeli öğrenme ve Markov Karar Verme Süreci (MDP - Markov Decision Process) bağlamında düşünülmektedir. MDP'ler normalde geçici soyutlamayı veya geçici genişletilmiş eylem kavramını içermezler. Tek ayrık zaman adımında incelenebilirler: t anında gerçekleştirilmiş bir eylem yalnızca $t+1$ anındaki durumu ve pekiştirmeyi etkilemektedir. Sonuç olarak, MDP yöntemleri, yüksek seviye geçici soyutlamaların getirdiği basitlik ve verimlilik avantajlarını kullanabilecek durumda değildirler. Bundan dolayı Sutton ve arkadaşları (1999) genişletilmiş ve değişken uzunluktaki durum geçişleri olan, MDP dahilinde eylem dizilerini temel alan "Seçenekler Yapısı"ni önermiştir. Bir seçenek, seçeneğin başlatılabileceği bir başlangıç durumuna, bir dahili davranış biçimine ve bir sonlandırıcı duruma sahip olan durum kümesiyle tanımlıdır. Eğer başlangıç ve bitiş durumları belirtilirse, geleneksel pekiştirmeli öğrenme yöntemleriyle seçeneğin dahili davranış biçimi öğrenilebilir. Burada seçenekleri, alt-görevler olarak düşünmek doğaldır, ve seçeneğin başlangıç durumu alt-görevin başlangıcı ve seçeneğin bitiş durumu alt-görevin tamamlanması olarak düşünülebilir.

Seçenekler yapısı, deneysel çalışmalarla da etkinliğini ve güvenilirliğini kanıtlamıştır, fakat bu yapıda açıkta kalan noktalardan birisi, başlangıç ve bitiş durumlarının otomatik olarak belirlenebilmesi problemidir. Bu değerler, seçeneğin geçiş durumları olarak da adlandırılmaktadır. Pekiştirmeli öğrenici,

bu geiş durumlarını otomatik olarak bulabilmelidir. Bu konuyla ilgili pek ok arařtırma yapılmıř ve bu geiş durumlarının otomatik olarak belirlenebilmesi iin eřitli yntemler nerilmiřtir.

Yaygın yaklařımlardan birisi, alt grevlerin durum uzayı baėlamında tanımlanmasıdır. ğrenici, ulařmaya deėer stratejik nem tařıdığına inanılan nemli durumları belirler. ğrenici, bu anahtar durumlara eriřmek iin alt davranıř biimleri belirler. Bu durumları bulmak iin gerekleřtirilen yaklařımlardan birisi tipik olmayan pekiřtirme deėerlerine sahip durumların aranmasıdır (Digney 1998). Bu yaklařımda Digney, alt-hedefler gibi pekiřtirme deėerinin yksek olduėu durumları veya sık ziyaret edilen durumları seer. Bu yaklařım pekiřtirmenin gecikmeli olarak oluřuėu durumlar iin faydalı olmayacaktır. McGovern ve Barto (2001), karřılařılma sıklıklarına gre durum seen bir yntem nermiřlerdir. Yntem alt-hedefleri otomatik olarak belirleyebilmek iin eřitlilik yoėunluėu yntemini kullanmaktadır. Bu yntemin temel noktası, gemiřte sıka ziyaret edilen durumların ğrencinin en uygun davranıř biiminin bir kısmı olduėunu varsaymasıdır. Kretschmar ve arkadařları (2003), bu yntemle ilgili birkaç olumsuz grř bildirmiř, ve sıklık ve mesafe metriėini kullanan FD (Frequency-Distance) yntemini nermiřtir. Sıklık temelli zmler iin en byk sorun, ğrencinin 'nemli' ve 'sıradan' durumlar arasındaki ayırımı yapabilmesi iin tm uzayı gereksiz řekilde dolařmak zorunda kalmasıdır, bylece alt-grevler ğrenme srecinin ok ileri ařamalarında ortaya ıkmakta bylece kullanılabilirliėini yitirmektedir. Mannor (2004), tek bir durum yerine, bařlangı ve bitiř kmeleri, veya ara durum kmeleri kavramını nermiřtir, bu kavram Mannor'a gre ğrenciyi daha saėlam ve gvenilir bir ğrenme srecine doėru gtrecektir. Her alt-grevi ve alt davranıř biimini, kullanıcının bir durum kmesinden diėerine kaymasına izin verecek řekilde tanımlamıřlardır.

3. MATERYAL VE YÖNTEM

3.1. Bilişsel Mimari

Çoğu tanımlama bilişi, bilme ve kavramayla ilişkilendirir. Bilişsel sistem hakkındaki genel görüş, bilen ve bildiğine göre davranan bir sistem olduğudur. Çeşitli algılayıcılar vasıtasıyla çevreden algısal bilgi toplar, daha sonra elde ettiği bu algısal bilgiyi kullanarak, mantıksal tümdengelimler ve çıkarımlarla, bu bilginin çok ötesinde önemli dolaylı bilgiye ulaşır. Örneğin, yağmur başlamış ve aracın hızı olması gerektiğinden fazla ise bilişsel eleman bu bilgiden faydalanarak aracın kayarak kaza yapabileceği sonucuna ulaşabilir.

3.1.1. Bilişsel Mimari Yapısı

Bilişsel sistemin çekirdeğinde, paralel olarak çalışan ve birbirleriyle etkileşim halinde olan üç temel işlem vardır (Sun 2003). Reaktif işlemler göreceli olarak hızlı işlemler tipidir. Bunlar otomatik olarak gerçekleşen işlemlerdir, bilinçli olarak yapılmazlar, refleksler bu tür davranışların en ilkel örneğidir, bununla birlikte otomatik olarak da gerçekleştirilen pek çok davranış vardır. Bunlar doğuştan veya reflekslere dayalı olmak zorunda değildir, öğrendiklerimizi de içerebilir. Örneğin, hiç kimse otomobil kullanmayı bilir halde doğmaz, fakat bir kere öğrenildiği zaman otomobil kullanma esnasındaki andan ana hareketler, uygulandıkça otomatik hale gelir. Bunlar bilinçli davranışlar değildir, bilinçaltından yönetilirler, yani açık ifade sel bilgi kullanmazlar, bu yüzden doğrudan erişilebilir değildir. Bilinçli işlemler, 'düşünmek' kelimesiyle ifade edilen davranışları içerir. Belirli bir şartın oluşması durumunda, o şarta uygun kararlar vermek, bilinçli davranışlara girmektedir. Herhangi bir tipte bilinçli işlemle üretilen bilgi, diğerleri tarafından da kullanılabilir. Burada bilgi açık ifade sel bilgidir, doğrudan erişilebilir ve değiştirilebilir. Üçüncü tip 'akıl yürüten' tip olarak adlandırılır, aslında bu, bilişsel sistemlerin nihai kullanılabilirliğinde oldukça önemli bir rol oynar. Bir çıkmaza girildiği zaman, o çıkmazdan kurtulabilmek için akıl yürütme işleminin gerçekleştirilmesi 'akıl yürüten' bir davranıştır.

Bu üç ayrı işlem tipi, birbirleriyle sürekli etkileşen iki ayrı bilgi tipini temsil eden iki ayrı seviyeyle ifade edilebilir. Bu iki ayrı seviye, insan bilişinin

doğasını oluşturur. Alt seviye, yani örtülü veya dolaylı bilgiyi içeren seviye, bilinçaltında yerleşik olan ve doğrudan erişilemeyen bilgiyi kapsar. Bu bilgi erişilememekle birlikte, insan eylemlerinin tamamında kullanılır. Doğrudan değiştirilemez, ancak kullanılmanın sıklığı doğrultusunda kendiliğinden değişebilir, hatta silinebilir yani unutulabilir. Üst seviye ise bilinçlilik halinin temsilidir, bilinçlilik açık veya ifade edilebilir bilgiyi değerlendirir (Kirsh 1990, Coward ve Sun, 2004). Bilgiler açık olması dolayısıyla doğrudan erişilebilirdir. Biliş modelini geliştirmede iki tip bilgi de gerekli olduğu için sistemin, hem açık hem de örtülü bilgiyi düzgün şekilde kullanan iki seviyeli mimari şeklinde ele alınması doğru bir yaklaşım olarak görülebilir. . Dreyfus ve Dreyfus (1987), Smolensky(1988), Hadley (1995) ve Sun (2002) yaptıkları yayınlarda bu iki seviyeli mimariye destek vermişlerdir.

Reaktif işlemler, doğrudan erişilebilir bilgiyi kullanmadığı için alt seviye davranışları olarak nitelendirilebilir. Bilinçli ve akıl yürüten işlemler ise, açık ve ifade edilebilir bilgiyi kullandığı için, üst seviye davranış olarak nitelendirilebilir. Alt-üst seviyelendirmesinin ötesinde bir de her seviyede bilişin fonksiyonlarını yöneten çok sayıda birim vardır ve bu birimler ayrık alt sistemlerin parçalarıdır. Alt sistemlerin birimsel parçaları ise boyut/değer çiftleri, kurallar ve durum kümeleridir (Slusarz ve Sun 2001).

Alt seviye alt sistemlerinin ana elemanları boyut/değer çiftleridir. Yapıları (boyut, değer) şeklinde ifade edilir. Boyut bir nesneye bakış açısını, değer ise sıfatı veya şiddeti temsil etmektedir. Örneğin, (sol-arka-araç, uzak) şeklinde bir boyut/değer çifti, bize göre sol şeritte ve daha arkada bulunan aracın bizden uzakta olduğunu tarif eder.

Üst seviye alt sistemleri ise durum kümelerini ve kuralları kullanır. Durum kümeleri, boyut/değer çiftlerini bir gruba bağlamaktan sorumludur, böylece belirli bir durumu tarif ederler. Durum kümeleri tanımladıkları durumlar açısından tekildir, yani aynı durumu tanımlayan birden fazla durum kümesi yoktur. Bir durum kümesi örneği verecek olursak, "Sollamaya uygun: (öndeki_araç, yakın), (arkadaki_araç, uzak), (sol_öndeki_araç, uzak), (sol_arkadaki_araç, uzak), (sol_öndeki_araç, uzak)" dizisi 'Sollamaya Uygun' etiketli

durum kümesinin ifade edilme şeklini meydana getirir. Bu örnekte, durum kümesi, çevre şartlarının sollamaya uygun olduğunu ifade eden bir durum kümesidir ve kümede 4 adet boyut/değer çifti kullanılmıştır.

Üst seviyenin alt sistemlerinin diğer temel elemanı kurallardır. Kurallar buldukları alt sisteme göre değişir. Eylem-merkezli modüllerde eylem kuralları adını alırlar. Eylem merkezli olmayan modüllerde ise ilişki kuralları olarak adlandırılırlar. Bir kural örneği vermek gerekirse,

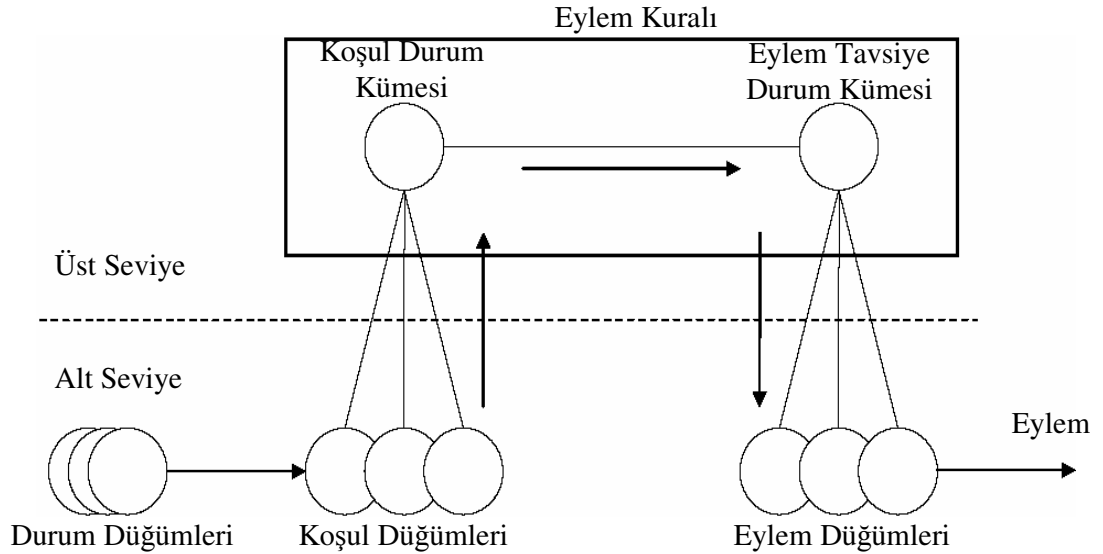
IF [öndeki_araç_çok_yakın] → THEN [Yavaşla]

Kuralın sol taraftaki kısmı koşulu, sağ taraftaki kısmı eylemi gösterir, koşul gerçekleştiği zaman sağ tarafta bulunan eylem tetiklenir. Kurallar, koşul olarak boyut/değer çiftlerine sahiptir. Eğer anlık durumla, kural koşulunun boyut değer çifti çakışırsa, kuralın çıkışı ve dolayısıyla önerilen eylem tetiklenir. İlişki kuralları da koşul olarak boyut/değer çiftlerine sahiptir fakat çıkışları hafıza durum kümesi yapısındadır. Kuralların her iki durumunda da, koşulun boyut/değer durum kümesi bir veya daha fazla durum kümesi ile ilişkilendirilir. Üst seviye açık bilgi ve alt seviye örtülü bilgi birbirleriyle boyut/değer çiftleri, durum kümeleri ve kurallar aracılığıyla bağlanır.

Bu iki seviyedeki bilgi tipinin matematiksel ifadesi konusunda da yoğun çalışmalar mevcuttur. Rumelhart (1986), Reber(1989), Seger (1994), Cleeramans ve arkadaşları (1998) ve Sun (1992, 1993, 1999) çalışmalarında, alt seviyenin alt-sembolik ve dağıtık yapısının bir sinirsel ağ tipi olan geri yayılım ağı kullanılarak ifade edilebileceğini göstermişlerdir. Bir sinirsel ağ, birbirleri arasında işaret akışına izin veren yollarla bağlanmış düğümler grubudur. Bu düğümler veya yollar kendi başlarına bir anlam taşımazlar. Bütün olarak bir ağ, kendi başına anlam taşımayan fakat grup halinde bir anlamı olan elemanların oluşturduğu bir sistemdir. Temel olarak bilginin işlenmesine izin verir, fakat işlemede kullanılan elemanlara erişime izin vermez. Geri yayılım ağı ise, doğru giriş-çıkış eşleştirmelerinin sağlandığı durum için, düğümleri arasındaki bağlantıları en uygun çıkışı verebilecek şekilde düzenleyebilen sinirsel ağ tipidir. Bir geri yayılım ağının birimleri, sembolik ve tek başına anlam taşımazlar, yani genel olarak mantıksal bir etiketle ilişkilendirilmemiş hesaplamaları

gerçekleştirebilecek yapıdadırlar, bu aslında alt seviye veya örtülü bilgi olarak adlandırılan yapının da genel özelliğini temsil etmektedir. Sun'a göre (1999) alt seviyedeki boyut/değer çiftlerini bir geri yayılım ağıyla temsil etmek, en uygun seçeneklerden birisidir. Her ayrık boyut ve her ayrık değer geri yayılım ağının bir düğümüdür ve örtülü bilginin bir parçasını temsil eder. Boyut/değer çiftini oluşturabilmek için, boyut düğümü ile değer düğümü birbirine bağlanır. Ayrık bir boyut düğümü veya bir değer düğümü tek başına anlamsızdır fakat durum kümeleri aracılığı ile diğer düğümlerle bağlandığı zaman açık bilginin birer elemanı haline gelir.

Üst seviye ise en uygun şekilde sembolik yolla temsil (Clark ve Karnikoff-Smith 1993, Sun 1994, Sun 1995). Sun'a göre sembolik sunum, "her birinin tercüme edilebilir ve net kavramlara sahip olması (mantıksal etiket)" anlamına gelmektedir. Böylece örtülü bilginin tersine açık bilgi, doğrudan erişilebilecek ve kullanılabilir önemli bilgi parçacıkları içerir (Smolensky 1988).



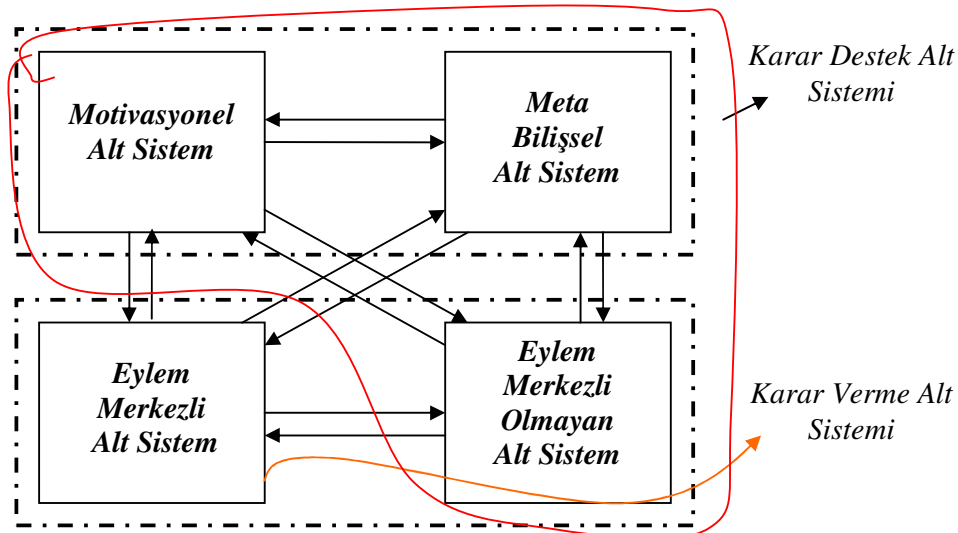
Şekil 3.1. Alt Seviye Üst Seviye Yapısı, Kurallar durum kümeleri ve düğümler

Durum kümeleri açık bilgiyi temsil eder, üst seviyededir fakat aynı zamanda alt seviyedeki ilgili boyut/değer çiftlerine de bağlıdırlar, başka bir deyişle üst seviyedeki durum kümesi düğümleri ile alt seviyedeki boyut/değer düğümleri birbirlerine bağlıdır. Durum kümeleri, alt seviyede boyut/değer

çiftleriyle ifade edilmiş dizilerin mantıksal bir etiketi olarak görev yapar. Kurallar durum kümelerini içerir ve üst seviyede bulunurlar, bununla birlikte, durum kümeleri boyut/değer çiftleriyle ilişkili olduğundan, kurallar da böylece alt seviyeye bağlanmış olur. Şekil 3.1'de üst seviye kural ve durum kümelerinin alt seviye boyut/değer çiftleriyle nasıl bağlandığı görülebilir. Bu sayede, iki seviyede etkileşimli olarak ayrı ayrı temsil edilebilir.

3.1.2. Bilişsel Mimari Alt Sistemleri

Genel olarak, bilişsel mimari üzerine yapılan çalışmalarda dört alt sistem üzerinde yaygın bir şekilde fikir birliğine varılmıştır (Sun 2003, Shibabi ve Mourant 2001, Salvucci ve ark. 2002), bunlar eylem merkezli, eylem merkezli olmayan, duygusal ve meta-bilişsel alt sistemlerdir. Bunlardan eylem merkezli alt sistem karar verme sistemi olarak nitelendirilebilirken, diğer alt sistemler eylem seçimine bir destek olarak düşünülebilecek etkinin oluşturulduğu karar destek sistemi başlığı altında incelenebilir. Bu alt sistemler kendi içinde bağımsız çalışmakla birlikte, her zaman adımında aynı giriş bilgisiyle çalışırlar ve çalışma esnasında birbirleriyle sürekli etkileşim halindedirler. Şekil 3.2'de bu dört alt sistemin şematik yapısı görülebilir.



Şekil 3.2. Bilişsel Mimari dört alt sistemi

3.1.2.1. Karar Verme Alt Sistemi

Eylem merkezli alt sistem, sistemin fiziksel ve zihinsel eylemlerini kontrol eder. Ortamla etkileşim temel görevidir, amacı veriyi alıp en iyi eylemin

seçilmesiyle ilgili tavsiyeler üretmektir. Sonuç tavsiyesi, her biri bir eylemi yöneten uygun fonksiyonlarla gerçekleştirilir.

Sistem tarafından herhangi bir eyleme geçilmeden, ortamın anlık durumu göz önünde bulundurulmalıdır. Sistemin girişi yani anlık durum, bir ortamda o an için neyin ne durumda olduğunu açıklayan, bir dizi boyut/değer çiftidir. Durum, sistemin o an içinde bulunduğu ortamdan algıladığı bilgileri tutar. Bu algılanan ham bilgiler (uzaklık, hız, vb.) boyut/değer çiftlerine dönüştürülür. Bu çiftler eylemlerin seçilip uygulanması için eylem merkezli alt sisteme gönderilir. Eylem merkezli alt sistem dahilinde durum hem alt hem de üst seviyede işlenir. Her iki seviyedeki birimler, verilen bilgi doğrultusunda eylem tavsiyeleri üretir.

Alt seviyedeki eylem merkezli alt sistem modüler olarak tasarlanmalıdır. Bunun anlamı, her biri bir eyleme, göreve veya uyarıcıya özel çok sayıda geri yayılım ağının oluşturulabileceğidir (Sun 1999). Bu ağlar belirli bir düşünce gerektirmeksizin, insan refleksleri gibi verilen bir uyarı (giriş) doğrultusunda bir tepki (çıkış) üretir. Öğrenme süreci eklenerek, bu ağlar değiştirilebilir ve böylece sistemin tepkileri değişir. Bununla birlikte, bu ağların bazıları içgüdü'nün reflekse dayalı davranışlarını yansıtabilir.

Eylem-merkezli alt sistemin üst seviyesi de birçok birimden oluşur. Bununla birlikte, üst seviyenin girişi alt seviyeye aynıdır. Durum, üst seviyedeki kural koşullarıyla karşılaştırılır. Her kural, aynı alt seviyedeki gibi, tetiklendiği zaman bir eylem tavsiyesi üretir. En iyi kuralı seçme yöntemi bir fayda ölçümü ile yönetilir. Fayda ölçümü, her bir eylem tavsiyesinin ne kadar yapıcı olacağını değerlendiren matematiksel bir fonksiyondur ve her birine bir "skor" atar. En yüksek skorlu eylem o an için en faydalı eylemdir. En iyi tavsiye seçildikten sonra, durum kümesi bağlı bulunduğu alt seviye eylem düğümlerini aktive eder.

Şekil 3.1'de, eylem seçme işleminde yer alan parçaların birbirleriyle ilişkisi görülebilir. Durum düğümleri (boyut/değer çiftleri), alt seviyede kuralın koşul düğümleri (boyut/değer çiftleri) ile karşılaştırılır. Eğer çakışırsa, koşul düğümleri üst seviyede bağlı oldukları durum kümesini harekete geçirir. Bu koşul durum kümesi de bağlı bulunduğu sonuç (eylem tavsiye) düğümünü

harekete geçirir. Koşul durum kümesi ve sonuç durum kümesi birlikte kuralı oluşturmaktadır. Sonuç durum kümesi harekete geçtiği zaman ise, alt seviyede bağlı bulunduğu eylem düğümlerini tetikler. Son olarak da, bu eylem düğümleri ilişkili oldukları temel seviye fonksiyonlarını harekete geçirir.

3.1.2.2. Karar Destek Alt Sistemi

Eylem merkezli olmayan alt sistem, genel bilgi olarak düşünülebilecek ne varsa içerir. "Semantik hafıza" olarak bilinen, dünyayla ilgili genel bilgi kavramsal sembolik yapıdadır. Bu yapı fikirleri, nesnelere ve gerçekleri içeren bilgi tipidir. Bu alt sistemin ürettiği sonuçlar eyleme dönüşmediği için bu alt sisteme eylem merkezli olmayan alt sistem adı verilmiştir. Burada çıkarımlar ve akıl yürütme işlemleri gerçekleştirilir.

Duygusal alt sistem bilişsel elemanın eylemlerinin altında yatan sebepleri sağlayan bilişin bir bölümüdür (Weiner 1992). Hedefler, görev temelli seviyede eylemleri yönlendirmede kullanılırken, eğilimler toplamda eylemleri belirler. Belirli hedeflerin nasıl aktif hedef seçildiğinin sebepleri bu alt sistemdedir. Eğilim bir elemanın ihtiyaçlarıyla bağlantılıdır. Bazı eğilimler diğerlerinden daha önemlidir, böylece hiyerarşik bir manada düzenlenebilirler (Toates 1986). Hiyerarşik olarak üst seviyede bir eğilim, o an için düşük seviyedekilere göre daha önemlidir. Bunun bir örneği, trafik kazasından korunma ile bir yere hızla ulaşmadır. Eğer kaza yapma durumu söz konusu ise bir bilişsel eleman için bir yere hızla ulaşmak o kadar önemli olmayacaktır.

Bilişsel alt sistem, hedefleri seçerken bir ihtiyacın erişilebilirliğini de göz önünde bulundurur. Bunun anlamı, eğilimin ihtiyaç duyduğu nesne, eğilimin karşılanabilmesi için hazır olmalıdır. Belirli bir ihtiyacı karşılamaya yönelik bir hedef belirlemek, eğer ihtiyacı karşılayan nesne ortalarda yoksa başarısız olacaktır. Öncelikle nesneyi bulabilmek için farklı hedefler tanımlanmalı, nesne bulunduktan sonra devam edilmelidir. Tyrell (1993), eğilimin ihtiyaçlarını şu şekilde belirlemiştir:

- Orantılı aktivasyon : İhtiyaç ne kadar çoksa, onu karşılamak için o kadar yüksek bir istek oluşur.

- Fırsatçılık : Çabuk karşılanabilecek ihtiyaçlara diğerlerinden daha fazla önem verilir.
- Eylemlerin bitişikliği : Başka bir eğilime geçmeden önce, bilişsel eleman bir eğilimi karşılayan işlemleri bitirmelidir.
- Israrcılık : Başka bir eğilime geçmeden önce, bilişsel eleman ısrarla bir eğilimi belirli bir noktaya kadar devam ettirmelidir.
- Gerektiği zaman kesme : Bazı durumlarda bir ihtiyacı karşılama durdurulmalı ve bir başka ihtiyaca geçilmelidir.
- Tercihlerin birleşimi : Tüm eğilimlerin ihtiyaçları hedefler belirlenirken düşünülmelidir, böylece bir kerede birden fazla eğilim karşılanabilir.

Meta-bilişsel alt sistem, bilişsel mimarinin temel denetleyicisidir (Metcalf ve Shimamura 1994). Diğer tüm alt sistemlerin hareketlerini düzenler. Örneğin duygusal alt sistemin eğilimlerine göre hedef yapısına yeni hedefler ekler. Eğilimleri ve güçlerini inceleyerek, o anlık odağın hangisi olacağına karar verir.

3.1.3. Bilişsel Mimaride Öğrenme

Bir sistemin değişen bir ortamda varlığını sürdürebilmesi için, öğrenme kabiliyetine sahip olması zorunludur. Öğrenme süreci, yeni bir davranış üretebilmek için üst seviye kurallarının ve alt seviye davranışlarının değiştirilebilmesini içerir. Bilişsel mimaride öğrenme, iki şekilde oluşur. Birincisi yeni gerçeklerin ve kuralların sisteme yerleştirildiği ayrık öğrenmedir. Örneğin, birisinin sürücüyü izlemesi gereken yolun doğrudan karşıya geçmek olduğunu söylemesi ile, sürücü bu bilgiyi yeni bir gerçeğe dayalı bilgi olarak kaydeder. İkincisi var olan gerçeklerin veya kuralların zaman içinde yapılan pratikler sonucu güçlendirilmesi veya zayıflatılmasıdır. Örneğin belirli bir kural veya gerçek çok sık kullanılıyorsa, kuvveti düzgün bir şekilde artar ve ileriki kullanımlar için erişimi daha kolay olur, diğer yandan çok az kullanılması veya hiç kullanılmamasıyla birlikte yeniden öğrenilmesi gereken bir noktaya kadar düşebilir. Ayrık ve sürekli öğrenmenin bir uyum içinde çalışmasıyla, bilişsel mimariler, acemiden uzmana kadar çok geniş bir uyumlu davranış yelpazesinde

kullanılabilir. Bununla birlikte, eşzamanlı çalışan farklı mekanizmalarla kabul edilebilir bir öğrenme üretimi bazen zor olabilir, ve bu tür mekanizmaların sürücülüğün karmaşık alanında nasıl etkileşeceğini belirlemek, yoğun bir çalışma gerektirmektedir.

Bilişsel mimarinin dört alt sisteminde de öğrenme işlemleri paralel ve birbirlerinden bağımsız olarak gerçekleştirilmektedir. Bunun ötesinde, alt seviye ve üst seviyede öğrenme de bilgi tiplerinin farklılığından dolayı farklı yöntemlerle yürütülmektedir. Üst ve alt seviyelendirme dahilinde, bilişsel elemanın öğrenme yöntemi iki temel yapıda incelenir: Öğrenme alt seviyede başlayabilir ve alt seviyeden üst seviyeye doğru bilgi aktarımı gerçekleştirilir, veya öğrenme üst seviyede başlar üst seviyeden alt seviyeye doğru bilgi aktarımı gerçekleştirilir.

3.1.3.1. Üstten Alta Öğrenme

Üstten alta öğrenme yöntemi genel olarak bireylerin öncelikle genel, açık bilgiyi öğrendiklerini ve daha sonra pratiklerle, bu tür bilgiyi belirli kullanılabilir örtülü tecrübeye çevirdiğini (Anderson ve Lebiere 1998) kabul eder. Anderson (1983) aritmetikten geometrik teorem ispatına kadar olan yelpazedeki tecrübe öğrenme çeşitlerinden elde edilen, veriye dayalı bir kavramayı önermiştir. Anderson için tecrübe gelişiminin başlangıç aşaması açık bilginin toplanmasıyla karakterize edilir. (Yani görevin nasıl gerçekleştirileceğinin açık bilgisi). Bu aşama sırasında öğrenci, görevi gerçekleştirebilmesi için bu açık bilgiyi beklemelidir. Pratikte birlikte, açık bilgiyi kullanmadan, tecrübenin değişik açılarının oluşturulmasına izin veren belirli yöntemler kümesi gelişir. Tecrübe yöntemselleştirildiği zaman, açık bilgiye neredeyse hiç başvurulmadan ve çoğunlukla aynı anda belirli detayların bilincinde olunmamasına rağmen görev gerçekleştirilebilir. Tecrübeye dayalı öğrenme üzerine açık/örtülü farkını benimseyen çoğu çalışma, üstten-alta yaklaşımını göz önünde bulundurur, yani öğrenciler öncelikle bir ortamdaki açık bilgiyi elde eder ve sonra pratik doğrultusunda, bu bilgiyi örtülü biçime dönüştürür ve bu da tecrübe performansına öncülük eder. Bu sürece yöntemselleştirme adı verilmektedir. Anderson'un (1983) önerisinde yöntemselleştirme, açık ifadesel bilginin komutlardan üretim kurallarına dönüştürülmesiyle elde edilir, burada bu

dönüştürme, pratiğin bir sonucudur. Anderson'un bir diğer önerisinde (1993) yöntemleştirilmenin, örneklerin açıkça hafızada tutulmasını ve bunların görevi gerçekleştirme esnasında durum kümeleri oluşturma vasıtasıyla yapılmasını önermiştir.

Temel fizik kuralları gibi hiç değişmeyen ve değişmeyecek tipte bilgiler bilişsel elemana harici (örneğin komutlar veya kitaplarla) olarak da sağlanabilir (Berry ve Broadbent 1984). Bunlara sabit kurallar adı verilmektedir. Harici verilen sabit kurallar asimilasyon yöntemi ile alt seviyeye geçirilir. Asimilasyon, eğitici öğrenme şeklinde gerçekleştirilebilir.

Taklit öğrenmede öğrenci, başka bir bilişsel elemanın durum ve eylemlerini gözlemler (Behnke ve Bennewitz 2005). Taklit eden bilişsel eleman, gözlenen durum ve eylemlere birebir uyan eylem kuralını oluşturur.

Bağımsız kural öğrenmede, tamamen rasgele veya özel bir yöntemle, değişik tiplerde kurallar üst seviyede bağımsız olarak oluşturulur. Daha sonra bu kurallar, bir başarı ölçütü kullanılarak deneyim yoluyla test edilir. Eğer kuralın başarı ölçütü belirli bir değer altındaysa kural silinir (Sun 2003). Başarı ölçütüne bölüm 3.1.3.2'de alttan üste öğrenme başlığı altında detaylı şekilde değinilecektir.

Üstten alta bilgi aktarımında, üstte oluşturulan bilgi, alt-seviye ağlarını yapılandırmada kullanılır. Bunun amacı sürekli tekrarlanan kuralları alarak, bu kuralları alt seviyede geri yayılım ağları ile temsil etmektir. Bu bilgi alt seviyede temsil edilerek, kural neredeyse otomatikleştirilir ve daha verimli hale getirilir (Anderson 1993, Dreyfus ve Dreyfus 1987).

3.1.3.2. Altan Üste Öğrenme

Bireylere yapacakları işlerle ilgili yeterli bilgi sağlanmadıysa, öğrenme alt seviyeden başlatılır.

Alt Seviyenin Eğitilmesi

Alt seviyede öğrenme birkaç yoldan gerçekleştirilebilir. Öğrenme ayarlarında, doğru giriş/çıkışın sağlandığı durumlarda, her ağ için doğrudan geri yayılım (öğreticili öğrenme algoritması) ağının kullanılması bu yöntemlerden

biridir. Bu tür öğreticili öğrenme yöntemi, her giriş için bir tek doğru çıkışın önceden belirlenmesini gerektirmektedir.

Öğrenme düzeninde, eğer harici olarak giriş/çıkış eşleştirmeleri sağlanmadıysa, pekiştirmeli öğrenme bu tür durumlarda yüksek bir başarı oranı sağlamaktadır (Sutton ve Barto 1998). Bu yöntem, tecrübeye dayalı öğrenmede en çok tercih edilen yöntemlerden biridir çünkü, çoğunlukla insan davranışının söz konusu olduğu durumlarda doğru olarak nitelendirilebilecek eylem sayısı birden fazladır. Pekiştirmeli öğrenme kullanılarak, bir eylemin pekiştirme işareti aracılığıyla, diyelim ki, -1 , 1 arasında değişen (1 mükemmel, -1 çok kötü ve arasındaki sonsuz sayıda diğer olasılıklar) yararlılığı ölçülebilir. Bu yöntem, en uygun davranışı seçebilmek için pekiştirme (yarar/zarar) tahminine dayanır. Anlık durum ve belirli bir eylemden sonraki durum karşılaştırılır ve eylemin uygulanmasının ne kadar yararlı olduğunu gösteren bir değer hesaplanır. Pekiştirme, sistemin hedeflerini, isteklerini ve durumu göz önünde bulunduran bir fonksiyonun sonucudur. Örneğin eylem merkezli alt sistemde, durumun hedefe ne kadar yakın olduğunu karşılaştırarak, eylem sonucu geçilebilecek durumların ne kadar istenir olduğunu temsil eden bir değer üretir. En yüksek pekiştirmenin elde edildiği eylem tavsiyesi en iyi tavsiye olarak belirlenir ve seçilir. Q-öğrenme ise, pekiştirmeli öğrenme algoritmaları içinde en çok tercih edilen yöntemlerden biridir (Watkins 1989). Q-öğrenmede, bir eylem gerçekleştirilmeden önceki ve gerçekleştirildikten sonraki toplam tahmin edilen pekiştirme, Q değerlerini geliştirmede kullanılır. Q değerleri, kestirilen toplam pekiştirmeyi temsil eder ve bundan dolayı sıralı eylemleri belirlemek için bir yöntem sağlar. Q-Öğrenme ve pekiştirmeli öğrenme ile ilgili detaylı bilgi bölüm 3.2'de ve bölüm 3.2.5.3'te sunulacaktır.

Taklit öğrenme alt seviyede farklı bir şekilde gerçekleştirilir(Behnke ve Bennewitz 2005). Burada sistem görevi gerçekleştiren başka bir yapıyı gözler ve daha sonra onu taklit etmeye çalışır. Üstten alta benzeşmenin tersine burada operasyon alttan üste doğrudur, yani üst seviye bilgisini ifade etmek için alt seviye bilgi kullanılır. Alt seviyenin bilgiyi elde etmesi için tek bir pekiştirme fonksiyonuna ihtiyaç vardır. Pekiştirme fonksiyonu verilmiş bir durumdaki eylemlere, aynı durumdaki başka bir sistemin değeriyle aynı olan bir değer atar.

Bu özel pekiştirme, diğer sistemi taklit eden eylemlerin seçilme olasılığının daha yüksek olduğu anlamına gelir. Burada belirtilmesi gereken, seçilme olasılıklarının yüksek olduğudur, garanti olmamasıdır. Bunun anlamı şudur, sistem daha iyi bilinen ve sonuç veren bir eylem yerine taklit eylemi kullanmaz. Sistem bir eylemi taklit ettiğinde pekiştirme hissedilir ve geri yayılım ağı güncellenir. Özel olarak taklit eylem ile aktive edilen düğümler arası bağlantılar güçlendirilir. Eylem alt seviyede tutulduğuna göre, üst seviye buradan kural çıkarımı yapabilir. Taklit öğrenme, doğrudan üst seviyede de uygulanabilir, yani bunun anlamı, kuralların, başka bir sistemin doğrudan gözlenmesi yoluyla oluşturulmasıdır.

Altta oluşturulan bilginin de aynı şekilde üst seviyeye transferi gerçekleştirilmelidir. Bu transfer için kullanılan yöntem kural-seçim-iyileştirme yöntemidir (Sun 2003).

Altan Üste Transfer ve Kural – Seçim – İyileştirme Algoritması

Alt seviye tarafından karar verilen eylem başarılıysa (yani belirli bir ölçütü sağlıyorsa) bilişsel eleman kuralı oluşturur ve kuralı üst seviye kural ağına ekler. Daha sonra bilişsel eleman gerçek dünyayla etkileşerek ve uygulanan kuralın sonucunu göz önünde bulundurarak oluşturulmuş kuralı iyileştirir. Eğer çıkış başarılı ise bilişsel eleman kuralın koşulunu daha genel (genişletilmiş) hale getirme yoluna gider; eğer çıkış başarılı değilse, kuralın koşulu daha özel (daraltılmış) hale gelecek ve anlık durumun dışında kalacaktır. Bu işlemin gerçekleştirilebilmesi için Sun (2003) şu akış şemasının kullanılmasını önermiştir:

Kural – Seçim – İyileştirme

1. Kural istatistikleri güncellenir
2. Kural oluşumu, genelleşme ve özelleşme için geçerli başarı ölçütü kontrol edilir.
 - 2.1. Eğer çıkış, geçerli başarı ölçütüne göre başarılı ise ve o koşul ile eylemi karşılayan bir kural yoksa, yeni bir kural (koşul → eylem) oluşturulur ve oluşturulmuş kural kümesine eklenir.

2.2. Sonuç geçerli başarı ölçütüne göre başarısız ise, tüm uyan kurallar özelleştirme yönünde gözden geçirilir.

2.2.1. Uyan kurallar kural kümesinden çıkarılır.

2.2.2. Özelleştirilmiş (daraltılmış) kurallar kümesine eklenir.

2.3. Eğer kural geçerli başarı ölçütüne göre başarılıysa, kural kümesindeki kurallar genişletme yönünde gözden geçirilir.

2.3.1. Uyan kuralları kural kümesinden çıkarılır.

2.3.2. Genişletilmiş kurallar kural kümesine eklenir. “İlave olarak aynı sonuca giden birden fazla kural varsa, her bu tür kural için bir ara düğüm oluşturulur: Her kuralın sol tarafındaki tüm elemanlar aynı ara düğüme bağlanır. Ve daha sonra tüm ara bağlantı düğümleri sonucu temsil eden düğüme bağlanır”. “Ayrıca kurallar, eğer olasıysa, birleştirilebilir. Eğer kural yapımı, daraltma ve genişletme o aşamada gerçekleştiriliyorsa, herhangi iki kuralın koşullarının birbirine yakın olduğu ve bundan dolayı birleştirilip birleştirilemeyeceği denetlenir. Eğer bir kural diğeri tarafından kapsaniyorsa, diğerin alt-kural listesine yerleştirilir. Eğer bir kural, diğeri tarafından bir boyutu hariç olmak üzere tamamen kapsaniyorsa, bu durumda ikisini de kapsayan yeni bir kural oluşturulur.

Başarı ölçütleri

Bir sonucun başarılı olup olmadığının ölçütü nedir? Bunun ölçütü bilgi kazanç ölçümüdür. Bilgi kazanç ölçümünü hesaplamak için her adımda şu işlem gerçekleştirilir: (s, s', r, a) kümesi göz önünde bulundurularak, (burada s , a eylemi gerçekleştirilmeden önceki durumu, ve r 'de a eylemi gerçekleştirildikten sonraki pekiştirmeyi temsil eder), C ile temsil edilen her kural koşulu ve onun minör farklılıkları (yani kural koşulu +/- giriş boyutlarından bir olası değer) için, gerçekleştirilen a eylemine göre, olumlu ve olumsuz çakışma sayısı güncellenir (istatistik güncelleme); yani $PM_a(C)$ (Olumlu çakışma sayısı, yani girişin C koşuluyla çakıştığı, a eyleminin gerçekleştirildiği

ve sonucun olumlu olduğu durumların sayısı) ve $NM_a(C)$ (Olumsuz çakışma sayısı, yani girişin C koşuluyla çakıştığı, a eyleminin gerçekleştirildiği ve sonucun olumsuz olduğu durumların sayısı) tekrar hesaplanır. Buradaki olumluluk ve olumsuzluğu belirlenmesi içinde bir denklem oluşturulmalıdır:

$$\gamma \max_b Q(s', b) + r - Q(s, a) > \text{eşik} \quad 3.1$$

eşitsizliği a eyleminin (kurala göre seçilmiş) kabul edilebilir derecede iyi olup olmadığını belirler (Sun ve ark. 2001). Buradaki Q-ölçümünün nasıl belirlendiği ve kullanıldığı ile ilgili detaylı bilgi bölüm 3.2.5.3'te verilmiştir. Her istatistik aşağıdaki formülle güncellenir:

$$sta = sta + 1$$

burada sta PM veya NM 'yi temsil eder ve her döngü sonunda

$$sta = sta \times 0.9 \quad 3.2$$

ile indirgenir. Bu istatistiğe dayanarak, bilgi kazanç ölçümünü hesaplanır, yani,

$$IG(A, B) = \log_2 \frac{PM_a(A) + 1}{PM_a(A) + NM_a(A) + 2} - \log_2 \frac{PM_a(B) + 1}{PM_a(B) + NM_a(B) + 2} \quad 3.3$$

burada A ve B aynı eylemi tetikleyen iki ayrı koşuldur. Ölçüm zorunlu olarak, A ve B değişik koşulları altında pozitif çakışmaların yüzdesini vermektedir. Eğer A , B 'ye göre, yüzdesini belirli bir değer üstüne taşıyabilirse, A 'nın B 'den daha doğru olduğu düşünülebilir. Algoritmada, eğer herhangi biri kural çakışma açısından diğer tümüne göre daha iyiyse (yani, olası tüm giriş koşullarıyla çakışmaya göre), kural o durumda en başarılı kural olarak kabul edilir.

Anlık adımın başarılı olup olmadığını söyleyen ve (s, s', r, a) dörtlüsünden tümüyle belirlenebilen başarı ölçütüne dayanarak, bir kuralı seçip seçmeyeceğimize karar verebiliriz.

- **Seçim:** Eğer anlık adım olumlu ise (anlık olumluluk ölçütüne göre) ve eğer üst seviyede bu kuralı kapsayan bir başka kural yoksa (hem koşulu hem

de eylemi), kural düzenlenir, $C \rightarrow a$, burada C tüm boyutların değerlerini tam olarak belirtir.

Bu işlem sırasında ortaya çıkacak sorunlardan birisi de kural sayısının hızla artmasıdır ve çözümlenmeye ihtiyacı vardır. Belirtildiği gibi, kuralların seçimi bir olumluluk ölçütüne göredir. Ek olarak bir olasılık değişkeni (p_{re}) de, kural seçimi için verilen kural seçimi ölçütü doğrultusunda, bir kuralın ne olasılıkla seçileceğini belirler. Yoğunluk fonksiyonu (d_r) bir eylem kuralını tutabilmek için gereken en düşük girişle çakışma sıklığını belirler. Örneğin, bu değişken, $\frac{1}{n}$ olarak seçilirse, her n adımda en az bir kere, gelen girişin kuralla çakışması gerekli hale gelir. Aksi takdirde, kural korunmaz ve yok edilir.

Genelleştirme ve özelleştirme operatörleri bilgi kazanç ölçümüne dayandırılmıştır. Genelleştirme, bir kuralın koşuluna bir boyut daha ekler, böylece kuralın gelen girişle çakışma olasılığı artar, özelleştirme ise kuralın koşulundan bir boyut çıkarır, böylece gelen girişle çakışma olasılığı düşer.

- **Genelleştirme:** Eğer $IG(c, all) > eşik$ ve $\max_{C'} IG(C', C) \geq 0$ ise, burada C kuralın anlık koşuluna, “all” hepsiyle-çakış kuralına (orijinal kural tarafından belirtilen aynı eyleme, fakat her duruma uyan koşula sahip) karşılık gelir ve C' , $C' = C$ artı bir değer (yani C' giriş boyutu olarak bir fazlaya sahiptir) şeklinde oluşur, bunun anlamı, eğer anlık kural başarılı ise ve genelleştirilmiş koşul potansiyel olarak daha iyi ise o zaman genelleştirmenin gerçekleştirilmesinin doğru olacağıdır, bu durumda kuralın yeni (genelleştirilmiş) koşulu olarak $C^N = \arg \max_{C'} IG(C', C)$ seçilir.
- **Özelleştirme:** Eğer $IG(C, all) < eşik$ ve $\max_{C'} IG(C', C) > 0$ ise, burada C kuralın anlık koşuluna (anlık duruma ve koşula uyan), “all” hepsiyle çakış (orijinal kuralla aynı fakat her duruma uyan koşuluyla) kuralına karşılık gelir, ve C' $C' = C$ eksi bir değer (yani, C' giriş boyutlarında bir eksik boyuta sahiptir) şeklinde oluşur. Bunun anlamı, eğer anlık kural başarısızsa ve özelleştirilen kural ondan daha iyiyse, kuralın yeni (özelleştirilmiş) koşulu olarak $C^N = \arg \max_{C'} IG(C', C)$ seçilir.

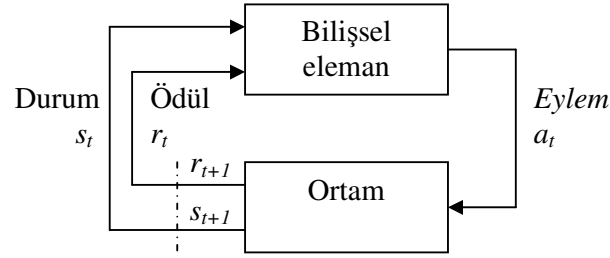
Bilişsel Mimari, herhangi bir an için, şu ana kadar değinilen öğrenme yöntemlerinden birini seçerek devam eder. Hangi öğrenme yönteminin seçileceği de karar destek sistemi birimleri tarafından yönlendirilmektedir. Hangi anda hangi öğrenme yönteminin seçileceği, öğrenme parametrelerinin durumu ve nasıl karar verileceğini belirleyen birimler, karar destek sistemini oluşturmaktadır. Karar destek sistemi ile ilgili yoğun çalışmalar devam etmektedir.

Bu tez çalışmasında, bilişsel elemana hazır bilgi sağlanmadan, bilişsel elemanın araç kullanma görevini sadece gözlemleyerek ve deneyerek öğrendiği bir yapı temel alınmıştır. Bu yapı, üstten-alta öğrenme yaklaşımı yerine alttan-üste öğrenme yaklaşımının kullanımını gerektirecektir, yani öğrenme alt seviyede başlatılacaktır. Giriş-çıkış eşleştirmelerinin kusursuz sağlanmadığı tüm durumlar için, bilişsel mimari dahilinde şu ana kadar üzerinde durulan tüm öğrenme yöntemleri, pekiştirmeli öğrenme yöntemlerinden birisi olan Q-öğrenme yöntemini, çekirdek yöntem olarak kullanmaktadır.

3.2. Pekiştirmeli Öğrenme

Pekiştirmeli öğrenme en basit tanımıyla deneme-yanılma veya ortamla etkileşim yoluyla öğrenme şeklinde tarif edilebilir. Bilişsel elemanın etkileştiği, kendisi dışındaki her şey, ortam olarak adlandırılmaktadır. Bilişsel eleman ve ortam sürekli olarak etkileşir; bilişsel eleman bir eylem seçer, ortam bu eyleme bir tepki verir ve eylem sonucu ortam, yeni bir duruma geçer. Yeni durumun yanı sıra, ortam pekiştirme olarak adlandırabileceğimiz ödül veya cezayı bilişsel elemana iletir. Bilişsel elemanın yapması gereken, alacağı ödülleri en yüksek seviyeye çıkarmak ve ceza getirecek hareketlerden uzak durmaktır.

Daha özel olarak, bilişsel eleman ve ortam, her t ayrık zaman adımında, $t = 0,1,2,3,\dots$, birbirleriyle etkileşir. Bilişsel eleman, her t zaman adımında S olası durumlar kümesinden ortam durumunun $s_t \in S$ ifadesini alır ve s_t durumunda seçilebilecek eylemler kümesi $A(s_t)$ iken, bir $a_t \in A(s_t)$ eylemi seçer. Bir zaman adımı sonra, yaptığı hareketin karşılığı olarak rakamsal bir $r_{t+1} \in \mathfrak{R}$ pekiştirmesi alır ve kendisini yeni bir s_{t+1} durumunda bulur (Şekil 3.3).



Şekil 3.3. Pekiştirmeli Öğrenme

Bilişsel eleman her zaman adımında, her olası eylem için durumlardan o eylemi seçme olasılığına doğru bir eşleştirme uygular. Bu eşleştirmeye bilişsel elemanın 'davranış biçimi' adı verilir ve π_t ile temsil edilir, $\pi_t(s, a)$, $s_t = s$ iken $a_t = a$ eyleminin seçilme olasılığını göstermektedir. Pekiştirmeli öğrenme yöntemleri, tecrübelerinin sonucu olarak bilişsel elemanın davranış biçimini nasıl değiştireceğini belirlemektedir. Bilişsel elemanın amacı, uzun vadede alacağı toplam ödüllerin seviyesini en yükseğe çıkarmaktır.

Eğer t zaman adımından sonra alınacak ödüllerin sıralamasını $r_{t+1}, r_{t+2}, r_{t+3}, \dots$ ile gösterirsek, bu dizinin maksimize edilebilmesi için beklenen dönüşün (expected return), R_t , en yüksek olması sağlanmalıdır:

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \dots \quad 3.4$$

Buna ilave olarak bir de indirim kavramı vardır. Bilişsel eleman zaman içinde aldığı indirimli ödüllerin toplamını maksimize etmelidir. Yani beklenen indirimli dönüşü (discounted return) maksimize edecek a_t eylemini seçmelidir:

$$R_t = r_{t+1} + \gamma \cdot r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad 3.5$$

buradaki γ değişkeni, $0 \leq \gamma \leq 1$, indirim oranıdır. Burada indirim kavramı, gelecek ile ilgili hesapların verilecek karardaki etkisini ayarlamaktadır. $\lim_{\gamma \rightarrow 0} R_t$ olursa bilişsel eleman miyop olacaktır, çünkü bu durumda çok kısa zaman hariç diğer tüm ödüller kısa zamanda sifira gider. Bunun anlamı bilişsel elemanın geleceği hiç hesaba katmamasıdır sadece anlık ödüllerle yetinmesidir. $\lim_{\gamma \rightarrow 1} R_t$

olması durumunda ise uzun vadedeki ödülleri gereğinden fazla hesaba katacaktır.

Örnek 1:

$s_t =$ (arka araç – çok uzak, sol arka araç – uzak, öndeki araç – orta uzak, ...)

sistem durumu göz önünde bulundurarak π davranış biçimi doğrultusunda bir eylem seçer

$a_t =$ (gaz pedalı – bas)

bu eylem uygulandıktan sonra yeni bir duruma geçilir

$s_{t+1} =$ (arka araç – yok, sol arka araç – çok uzak, öndeki araç – yakın, ...)

ve geçilen yeni durum için bir anlık ödül üretilir

$r_{t+1} = 0.376$

ve denklem 3.2'den uzun vadeli ödülleri de hesaplanabilir.

$R_t = 0.421$

Standart pekiştirmeli öğrenme yöntemleri, ele aldığı problemin Markov özelliğine sahip olduğu varsayımından yola çıkmaktadır. Markov özelliğine sahip olmayan ortamlarda pekiştirmeli öğrenme yöntemleri performans sağlayamamaktadır.

3.2.1. Markov Özelliği

t anında uygulanan bir eyleme ortamın $t+1$ anındaki tepkisi, genel olarak, nedensel durumlar için, daha önceden oluşmuş her şeye bağlı olabilir. Sistem dinamikleri, yalnızca tam olasılık dağılımı belirlenerek, tüm s', r ve geçmiş olayların tüm olası değerleri, $s_t, a_t, r_t, s_{t-1}, a_{t-1}, \dots, r_1, s_0, a_0$, için tanımlanabilir:

$$P_r \{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t, r_t, s_{t-1}, a_{t-1}, \dots, r_1, s_0, a_0\} \quad 3.6$$

Eğer ortam Markov özelliğine sahipse, $t+1$ anında ortamın tepkisi, yalnızca t anındaki durum ve eyleme bağlıdır. Bu durumda denklem 3.6'da belirtilen sistem dinamikleri, tüm s', r, s_t ve a_t 'ler için şu şekilde ifade edilebilir:

$$P_r \{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t\} \quad 3.7$$

yani, eğer

$$P_r \{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t, r_t, s_{t-1}, a_{t-1}, \dots, r_1, s_0, a_0\} = P_r \{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t\} \quad 3.8$$

ise durum bir Markov durumudur ve bu ortam Markov özelliğine sahiptir. Eğer bir ortam Markov özelliğine sahipse, bir adımlık dinamiği bize bir sonraki durumu ve beklenen ödül kestirebilmemize olanak verir.

Markov özelliği pekiştirmeli öğrenmede önemlidir çünkü kararlar ve değerler yalnızca anlık durumun bir fonksiyonu olarak kabul edilecektir. Bunun sağlanabilmesi için durum bilgisinin yeterince bilgilendirici olması gerekmektedir.

3.2.2. Markov Karar Verme Süreci

Markov özelliğine uyan pekiştirmeli öğrenme görevine Markov Karar Verme Süreci veya yaygın olarak kullanılan kısaltmasıyla MDP (Markov Decision Process) adı verilmektedir. Eğer durum ve eylem uzayı sonlu ise, sonlu MDP olarak adlandırılır. Sonlu MDP durum-eylem kümeleri ve ortamın tek adımlık dinamiği ile tanımlanır. Verilen bir durum s ve eylem a için olası her s' bir sonraki durumu

$$P_{ss'}^a = P_r \{s_{t+1} = s' \mid s_t = s, a_t = a\} \quad 3.9$$

ile ifade edilir. Benzer şekilde verilen herhangi bir geçerli durum s ve eylem a için, bir sonraki durum s' ile birlikte, bir sonraki ödülün beklenen değeri

$$R_{ss'}^a = E\{r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'\} \quad 3.10$$

olur. Bu iki ifade, bir sonlu MDP'nin en önemli iki dinamiğini oluşturur.

3.2.3. Değer Fonksiyonları

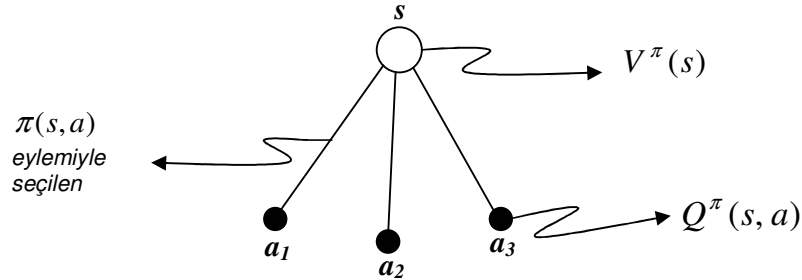
Neredeyse tüm pekiştirme öğrenme algoritmaları değer fonksiyonlarının tahmini - bilişsel elemanın (veya durum-eylem çiftlerinin) verilen bir durumda bulunmasının ne kadar yararlı olduğunu tahmin eden fonksiyon - üzerine kurulmuştur. MDP'ler için değer fonksiyonu şu şekilde tanımlanabilir:

$$V^\pi(s) = E_\pi \{R_t \mid s_t = s\} = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s \right\} \quad 3.11$$

burada $E_\pi\{\}$, t herhangi bir zaman adımı ve π davranış biçimini takip eden bir bilişsel eleman için beklenen değeri temsil etmektedir. V^π fonksiyonu, π davranış biçimi için durum-değer fonksiyonu olarak adlandırılmaktadır. Benzer şekilde, bir π davranış biçimi doğrultusunda s durumunda a eylemini seçmenin değeri de $Q^\pi(s, a)$ ile temsil edilir, s durumundan başlayıp a eylemi uygulandıktan sonra π davranış biçimi doğrultusundaki beklenen dönüşü gösterir:

$$Q^\pi(s, a) = E_\pi\{R_t \mid s_t = s, a_t = a\} = E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\right\} \quad 3.12$$

Q^π , π davranış biçimi için eylem değer fonksiyonu olarak adlandırılır. V^π ve Q^π tecrübelerden tahmin edilebilir. Şekil 3.4'te verilen bir durum ve o durumda seçilebilecek tüm eylemler için davranış biçimini, durum ve durum-eylem değerlerinin temsili görebiliriz.



Şekil 3.4. Davranış Biçimi, V ve Q fonksiyonları

Değer fonksiyonlarının pekiştirmeli öğrenme ve dinamik programlamada kullanılan temel özelliği, özel yinelemeli ilişkileri karşılayabilmelidir.

$$V^\pi(s) = E_\pi\{R_t \mid s_t = s\} = E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right\} = E_\pi\left\{r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s\right\} \quad 3.13$$

$$= \sum_a \pi(s, a) \sum_{s'} P_{ss'}'' \left[R_{ss'}'' + \gamma \cdot E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_{t+1} = s' \right\} \right] \quad 3.14$$

$$= \sum_a \pi(s, a) \sum_{s'} P_{ss'}'' [R_{ss'}'' + \gamma \cdot V^\pi(s')] \quad 3.15$$

burada a eylemleri $A(s_t)$ kümesinden, s' eylemleri ise S kümesinden çekilmektedir. Son denklem V^π için Bellman denklemidir ve bir durumun değeri ile varislerinin durumları arasındaki ilişkiyi göstermektedir. Bellman denklemi basit olarak tüm olasılıkları, her birinin oluşma olasılığıyla ağırlıklandırarak ortalar. Şekil 3.4'ten de bu sonuca ulaşabiliriz.

Örnek 2:

$s_t =$ (arka araç – çok uzak, sol arka araç – uzak, öndeki araç – orta uzak, ...)

Seçilebilecek eylemler

$a_t^1 =$ (gaz pedalı – bas)

$a_t^2 =$ (fren pedalı – bas)

$a_t^3 =$ (direksiyon – sağa çevir)

...

ve geçilebilecek durumlar

$s_{t+1}^1 =$ (arka araç – yok, sol arka araç – uzak, öndeki araç – yakın, ...)

$s_{t+1}^2 =$ (arka araç – yok, sol arka araç – çok uzak, öndeki araç – orta uzak, ...)

...

$Q(s, a)$ fonksiyon değerleri

$Q(s_t, a_t^1) = 0.562$

$Q(s_t, a_t^2) = 0.033$

...

$V(s)$ fonksiyon değerleri

$V(s_{t+1}^1) = 0.622$

$V(s_{t+1}^2) = 0.321$

3.2.4. En Uygun Değer Fonksiyonu

Bir π davranış biçiminin denklem 3.5'teki beklenen dönüşü, eğer başka bir π' davranış biçiminin beklenen dönüşünden daha iyi veya ona eşitse, π' davranış biçiminden daha iyidir veya eşittir denebilir. Diğer bir deyişle, eğer $V^\pi(s) \geq V^{\pi'}(s)$ ise $\pi \geq \pi'$ olur. Her zaman için diğer davranış biçimlerinden daha iyi veya onlara eşit olan bir davranış biçimi vardır. Bu en uygun davranış biçimidir. Birden fazla olabileceği için, tüm en uygun davranış biçimleri π^* ile

temsil edilir, ve aynı değer fonksiyonu olan, en uygun değer fonksiyonunu, V^* kullanırlar:

$$V^*(s) = \max_{\pi} V^{\pi}(s) \quad 3.16$$

En uygun davranış biçimleri aynı zamanda en uygun eylem-değer fonksiyonlarını, Q^* , kullanırlar:

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a) \quad 3.17$$

Q^*, V^* cinsinden şu şekilde yazılabilir:

$$Q^*(s, a) = E\{r_{t+1} + \gamma \cdot V^*(s_{t+1}) \mid s_t = s, a_t = a\} \quad 3.18$$

V^* için Bellman en uygunluk denklemini şu şekilde yazabiliriz:

$$V^*(s) = \max_{a \in A(s)} E\{r_{t+1} + \gamma \cdot V^*(s_{t+1}) \mid s_t = s, a_t = a\} \quad 3.19$$

$$= \max_{a \in A(s)} \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma \cdot V^*(s')] \quad 3.20$$

Q^* için Bellman en uygunluk ise şu şekildedir:

$$Q^*(s) = E\{r_{t+1} + \gamma \cdot \max_a Q^*(s_{t+1}, a) \mid s_t = s, a_t = a\} \quad 3.21$$

$$= \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma \cdot \max_{a'} Q^*(s', a')] \quad 3.22$$

Elimizde bir V^* olduktan sonra en uygun davranış biçimini belirlemek kolaydır. Her s durumu için, Bellman en uygunluk denkleminde, maksimumun elde edildiği bir veya daha fazla eylem olacaktır. Bu eylemlere sıfır olmayan olasılık yükleyen her davranış biçimi en uygun davranış biçimidir. Q^* fonksiyonu da en uygun eylem seçiminde oldukça kolaylık sağlamaktadır. Q^* ile bilişsel eleman, $Q^*(s, a)$ değerini maksimum yapan herhangi bir eylemi basitçe bulabilir.

3.2.5. Çözümleme Yöntemleri

3.2.5.1. Dinamik Programlama

Bir Markov İşlemi olarak ortamın eksiksiz bir modelinin verildiği durumda en uygun davranış biçimini hesaplayabilmek için kullanılacak algoritmalar kümesine, Dinamik Programlama adı verilmektedir. Dinamik Programlamanın temel fikri, iyi davranış biçimlerini aramayı yapılandırmak ve organize etmek için değer fonksiyonlarının kullanılmasıdır. Denklem 3.12'deki Bellman değer fonksiyonlardan yola çıkarak yinelemeli bir şekilde durum değerlerini günceller. İşletim prosedürü çizelge 3.1'deki gibidir:

Çizelge 3.1. Dinamik Programlama İşletim Prosedürü

1. Her $s \in S$ için $V(s) = 0$ yap.
2. Tekrarla
 $\Delta \leftarrow 0$
3. $s \in S$ için
 $v \leftarrow V(s)$

$$V(s) = \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V(s')]$$

$$\Delta \leftarrow \max(\Delta, |v - V(s)|)$$

$$\Delta < \theta \text{ (küçük bir pozitif sayı) olana kadar 2. adıma dön}$$
4. Çıkış $V \approx V^\pi$

En uygun davranış biçimine ulaşmak için gereken zamanın durumların ve eylemlerin polinomu şeklinde artmasından dolayı dinamik programlamanın büyük problemler için pek pratik olmadığı görülebilmektedir.. Üstelik ortamın mükemmel bir modeline sahip olması zorunluluğuyla kurulduğu için araç sürme gibi bir insan davranışı modeli için uygulanabilir değildir.

3.2.5.2. Monte Carlo Yöntemi

Dinamik Programlamanın tersine Monte Carlo Yöntemi, sistem modeline ihtiyaç duymaz. Tek ihtiyacı olan deneyimlerdir – yani ortamla olan gerçek veya benzeşimli etkileşimler sonucu elde edilen durum, eylem ve ödül

bilgileridir. Ortamla ilgili hiçbir ön bilgiye sahip olmayıp, yalnızca deneyimlerle en uygun davranışı yakalayabilmesi bu yöntemin dikkat çekici özelliğidir.

Monte Carlo modelinin çalışabilmesi için, görevin zaman içinde bölümlenmesi gerekmektedir. Belirlenen bölümlenme tamamlandıktan sonra, değer atamaları yapılacaktır. Genel işletim prosedürü çizelge 3.2'deki gibidir:

Çizelge 3.2. Monte Carlo Yöntemi İşletim Prosedürü

$\pi \leftarrow$ değerlendirilecek davranış biçimi
 $V \leftarrow$ rasgele bir durum değer fonksiyonu
 $\text{Ödüller}(s) \leftarrow$ her $s \in S$ için boş bir ödül kümesi
 Sürekli tekrarla:
 (a) π davranışına göre bir bölüm oluştur
 (b) Bölüm içinde görülen her s için
 $R \leftarrow s$ ile ilk karşılaşmadan sonraki ödül
 R 'yi $\text{Ödüller}(s)$ 'e ekle
 $V(s) \leftarrow \text{ortalama}(\text{Ödüller}(s))$

Monte Carlo Yöntemi ortamın tam bir modeline ihtiyaç duymamasına karşın, bölümlere ayrılmak zorunda kalması ve çevrimdışı çalışması bu yöntem için bir dezavantaj olarak görülebilir. Sürücü davranışının bu şekilde bölümlenmesi pek pratik olmadığı için bu yöntemde bizim açımızdan pek uygulanabilir olmadığını söyleyebiliriz.

3.2.5.3. Geçici Farklar Yöntemi ve Q-Öğrenme

Geçici Farklar Yöntemi Monte Carlo ve Dinamik Programlama yöntemlerinin bir bileşimi gibidir. Monte Carlo Yöntemi gibi doğrudan ham deneyimden öğrenebildiği gibi, dinamik programlama yöntemleri gibi bir bölümün tamamlanmasını beklemeye gerek kalmadan andan ana değer güncellemesi yapabilmekte, yani çevrimiçi çalışabilmektedir. Genel çalışma prensibi Çizelge 3.3'teki gibidir:

Çizelge 3.3. Geçici Farklar Yöntemi İşletim Prosedürü

1. $\pi \leftarrow$ değerlendirilecek davranış biçimi
2. $V(s)$ 'e rasgele bir şekilde başlangıç değerleri atanır
3. Tekrarla(Herhangi bir örnek dizisi için)
 - s durumuna bak
 - Tekrarla(Örnek dizisinin içinde)
 - $a \leftarrow \pi$ tarafından s durumu için seçilen eylem
 - a eylemini uygula, r ödülünü ve oluşan yeni s' durumunu gözlemler
 - $V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$
 - $s \leftarrow s'$

Pekiştirmeli öğrenmedeki en çarpıcı ilerlemelerden biri, bir geçici farklar kontrol algoritması olan Q-Öğrenmenin (Watkins 1989) geliştirilmesiyle gerçekleşmiştir. En basit formuyla, tek-adımlık Q-Öğrenme şu şekilde tanımlanmıştır:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad 3.23$$

Bu yöntemde öğrenilmiş eylem-değer fonksiyonu Q , hangi davranış biçimi takip edilirse edilsin, doğrudan en uygun eylem-değer fonksiyonuna Q^* , yakınsamaktadır. Bu durum, algoritmanın analizini oldukça basitleştirmekte ve erken yakınsama ispatlarını hazırlamaktadır. Akış diyagramını çizelge 3.4'te inceleyecek olursak,

Çizelge 3.4. Q-Öğrenme İşletim Prosedürü

1. $Q(s, a)$ için rasgele başlangıç değerleri ata
2. Bunu sürekli tekrar et:
 - s 'i başlat
 - Sürekli tekrar et
 - Q değerini kullanan davranış biçimi doğrultusunda s durumundan bir a seç
 - a eylemini uygula, r ve s' 'i gözlemler
 - $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_a Q(s', a') - Q(s, a)]$
 - $s \leftarrow s'$
3. s sonlandırıcı değilse 2'ye dön

Böylece, her durum-eylem çiftleri için bir değer tablosu oluşacaktır. Bu çizelge 3.5'de gösterilmiştir.

Çizelge 3.5. Q-değerleri tablo yapısı

<i>durum/eylem</i>	Eylem 1	Eylem 2	Eylem 3	Eylem n
Durum 1	0.002	0.342	0.188	0.862
Durum 2	0.221	0.541	0.358	0.452
Durum 3	0.479	0.323	0.655	0.224
...
...
Durum m	0.552	0.711	0.123	0.122

3.2.6. Genelleştirme ve Fonksiyon Yakınlaştırıcı

Değer fonksiyonlarıyla ilgili kestirimler bir tabloyla temsil edilirse, yalnızca çok büyük tabloları tutmak için gereken bellek ihtiyacı değil, aynı zamanda onları doğru bir şekilde doldurmak için gereken zaman ve veri miktarı da bir sorun teşkil edecektir. Bu sorun genelleştirme yöntemiyle aşılabilir. Yapılması gereken pekiştirmeli öğrenme yöntemleriyle genelleştirme yöntemlerinin birleştirilmesidir. Genelleştirme tipinin böyle bir uygulamadaki şekline fonksiyon yakınlaştırıcısı adı verilmektedir, istenen bir fonksiyondan örnekleri alır (örneğin değer fonksiyonu) ve bunlardan fonksiyonun tamamına bir yaklaştırma oluşturarak, genelleştirme girişiminde bulunur.

t anındaki yaklaşık değer fonksiyonu V_t , bir tablo olarak değil, fakat $\bar{\theta}$ parametre vektörüyle değişken fonksiyonel bir yapıyla temsil edilir. Bunun anlamı, V_t değer fonksiyonunun, $\bar{\theta}$ değiştikçe, her zaman adımında, tamamen $\bar{\theta}$ 'ye bağımlı olmasıdır. Örneğin, V_t , bağlantı ağırlıklarının $\bar{\theta}$ vektörü olduğu bir yapay sinir ağı olabilir. Ağırlıklarını ayarlayarak, çok çeşitli V_t fonksiyonları ağa uygulanabilir.

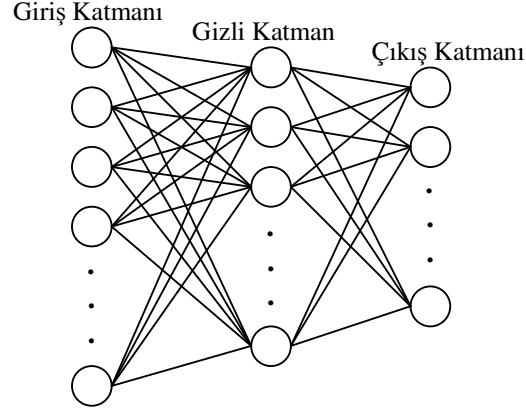
Değişken sayısı, ($\bar{\theta}$ 'nin bileşenlerinin sayısı) durum sayısından çok daha azdır, ve bir parametrenin değişimi, bir çok durumun tahmini değerini değiştirir.

Fonksiyon yakınlaştırıcı için önerilen çeşitli yaklaşımlar vardır. Anderson (1987) ve Lin (1991a, 1991b, 1991c) geçici farklar yöntemi ile bağlantısal hata geri yayılım algoritmasını başarılı bir şekilde birleştirmiştir. Watkins (1989) CMAC algoritmasını, Grefenstette (1990) genetik algoritmayı, Mahadevan ve Connell (1991) istatistiksel kümeleme yöntemini, Chapman ve Kaelbling (1991) karar ağacına benzer bir yöntemi ve Moore (1991) basit olarak değişken durum çözümlemesini kullanan arama (look-up) tabloları yöntemini önermiştir.

Geri yayılım ağının, gizli katmanlarıyla bilgi işlemenin örtülü yapısını karşılayabildiğine bölüm 3.1.1'de değinilmişti. Lin'in (1992) önerdiği geri yayılım ağları ile geçici farklar algoritmasının bileşimi, aslında bilişsel mimarinin alt seviye pekiştirmeli öğrenme yapısının kurulmasında en etkili yöntemlerden biri gibi görünmektedir.

3.2.6.1. Geri Yayılım Ağları

Geri yayılım ağı sanal nöronlardan oluşan bir yapıdır (Şekil 3.5). Her nöron, her birine belirli ağırlıklar atanmış, belirli sayıda girişe sahiptir. Ağırlıklar girişin ne kadar önemli olduğunu tarif etmektedir. Nöronun net değeri ağırlıklandırılmış toplamdır, yani gelen girişlerin kendi ağırlıklarıyla çarpılarak toplanmasıdır. Her nöron bir eşik değerine sahiptir, eğer girişlerden gelen değer eşik değerini geçerse nöron aktive olur (yani çıkışı 1 yapar), aksi takdirde aktif değildir (çıkışı 0 olur).



Şekil 3.5. Geri Yayılım Ağı

Geri yayılımın temel mantığı ise çıkış değişimlerinde gözlenen hatayı azaltmak için, ağırlıklarda değişiklik yapmaktır. Her örnek giriş kombinasyonu için her çıkışın istenen değeri, d 'yi, geçerli değeri o 'yu, $d-o$ hatasındaki özel bir w ağırlığının etkisini düşünelim. w ağırlığındaki büyük bir değişim eğer büyük bir çıkış hatasını azaltabiliyorsa ve bu azalma miktarı dikkate değerse anlamlıdır. Geri yayılımın üç önemli ifadesi vardır:

$$\Delta w_{i \rightarrow j} = \alpha \cdot o_i \cdot o_j (1 - o_j) B_j \quad 3.24$$

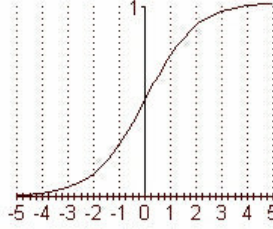
$$B_j = \sum_k w_{j \rightarrow k} o_k (1 - o_k) B_k \quad \text{Gizli katmanlardaki tüm düğümler için} \quad 3.25$$

$$B_z = d_z - o_z \quad \text{Çıkış katmanındaki tüm düğümler için} \quad 3.26$$

Bu ifadelerdeki $\Delta w_{i \rightarrow j}$ i düğümünden j düğümüne giden bağlantının ağırlığındaki değişimi, o_i ve o_j , i ve j düğümlerinin çıkışını, d düğümün doğru çıkışını, B ise çıkış hatasını temsil etmektedir. Geri yayılım algoritması, ağırlık değerlerinin geri yönde değiştirmek için, hesaplanmış çıkış hatasını kullanmaktadır. Bu hatayı elde etmek için öncelikle ileri yönde bir yayılım gerçekleştirilmelidir. İleri yönde yayılım gerçekleştirilirken, nöronlar şekil 3.6'daki sigmoid aktivasyon fonksiyonuyla çalıştırılır:

$$f(x) = \frac{1}{1 + e^{-\text{giriş}}}$$

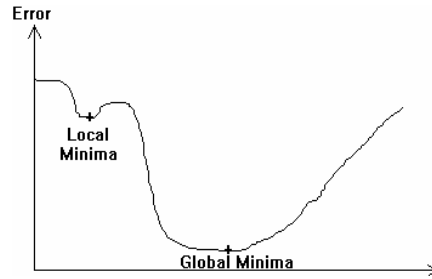
3.27



Şekil 3.6. Sigmoid Fonksiyonu

Prosedürü şu şekilde işlemektedir:

1. Giriş örnekleri doğrultusunda ileri yönde yayılım gerçekleştirilir ve çıkış hatası hesaplanır.
2. Her ağırlık matrisinin tüm ağırlık değerleri şu formüle göre değiştirilir:
Ağırlık(eski) + öğrenme oranı * çıkış hatası * çıkış(nöron i) * çıkış(nöron i+1) * (1 - çıkış(nöron i+1))
3. 1. adıma geri dönülür
4. Eğer çıkış örüntüleri, hedef örüntülerle uyumluysa prosedür sonlanır.



Şekil 3.7. Hata Grafiği

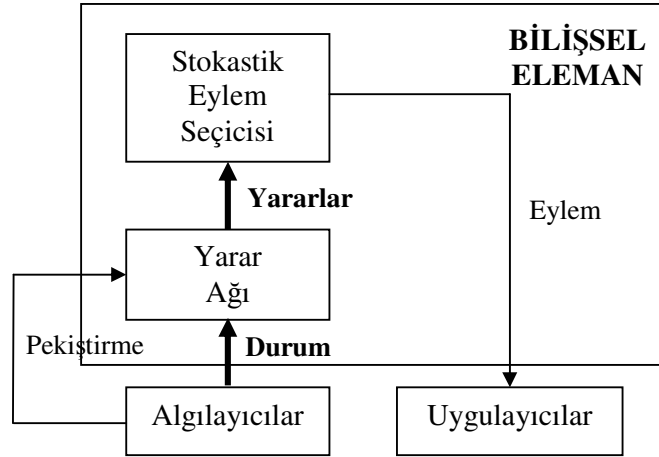
Öğrenme oranı, 0 ile 0.9 arasındadır. Sinirsel ağın en uygun durumuna doğru ilerleyeceği adımın büyüklüğünü belirler. Geri-hata algoritmasının altında yatan geçerli fikir, global minimayı bulmak için hata yüzeyini taramaktır. Şekil 3.7'de hata grafiği görülmektedir. Öğrenme oranı için bir değer saptamak oldukça zordur. Eğer büyük bir değer verilirse, lokal minimalardan korunulabilir,

fakat bu sistemi salınımına sürükleyecektir. Buna rağmen, eğer öğrenme oranı küçük tutulursa, sistem lokal bir minimada sıkışıp kalabilir.

3.2.6.2. Geri Yayılım Ağları ve Q-Öğrenme

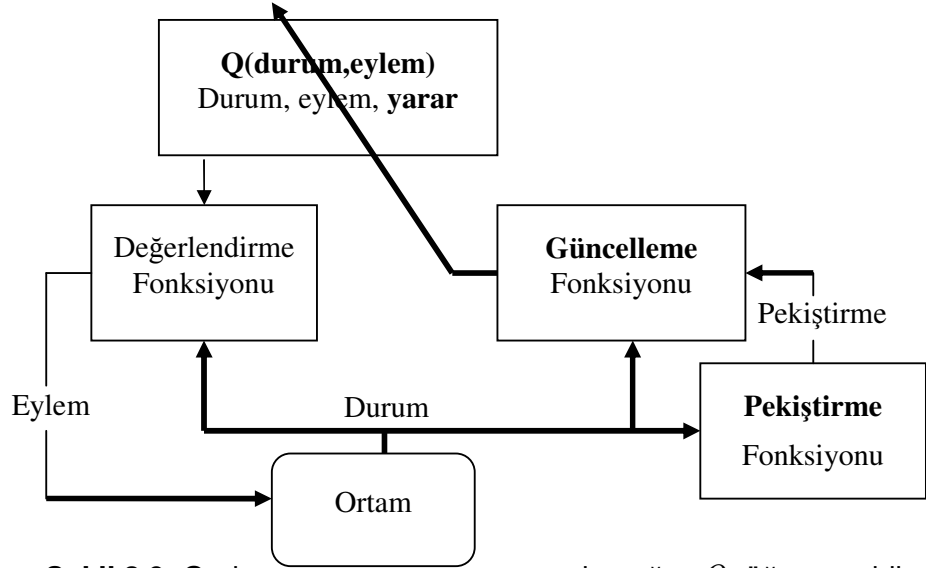
Lin (1992), Q-değerlerinin hesaplanabilmesi için geçici farklar yöntemiyle elde edilen hatayı geri yayılım hatası olarak kabul eden bir geri yayılım ağının (Rumelhart 1986) kullanılmasını önermiş ve bu amaçla, $Q(s,a)$ fonksiyonunu modelleyen bir geri yayılım ağı kullanmıştır. Lin (1992) bu ağa yarar ağı adını vermiştir. Şekil 3.8'de yarar ağının blok şeması görülmektedir.

Verilen bir yarar ağı ve s durumu için, bilişsel elemanın davranış biçimi, $Q(s,a)$ fonksiyonunu en yüksek değere getiren a eyleminin seçilmesi olacaktır (aslında daha farklı seçim ölçütleri de kullanılabilir). Bundan dolayı yarar ağı yalnızca değerlendirme fonksiyonunu değil, aynı zamanda davranış biçimini de temsil edebilir.



Şekil 3.8. Yarar ağı taslağı. Kalın çizgiler vektörleri, ince çizgiler skalar sinyalleri işaret eder.

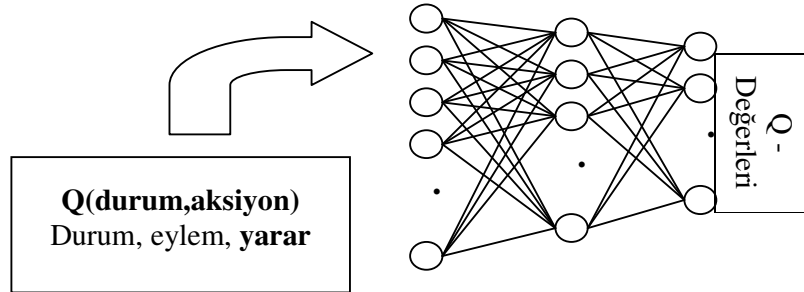
Touzet (1999) Lin'in önerdiği taslak ise Şekil 3.9'daki gibi ifade etmiştir.



Şekil 3.9. Geri

yayılım ağı - Q -öğrenme bileşimi

Bu yapıda yer alan değerlendirme fonksiyonunu Şekil 3.10'daki gibi ifade edebiliriz.



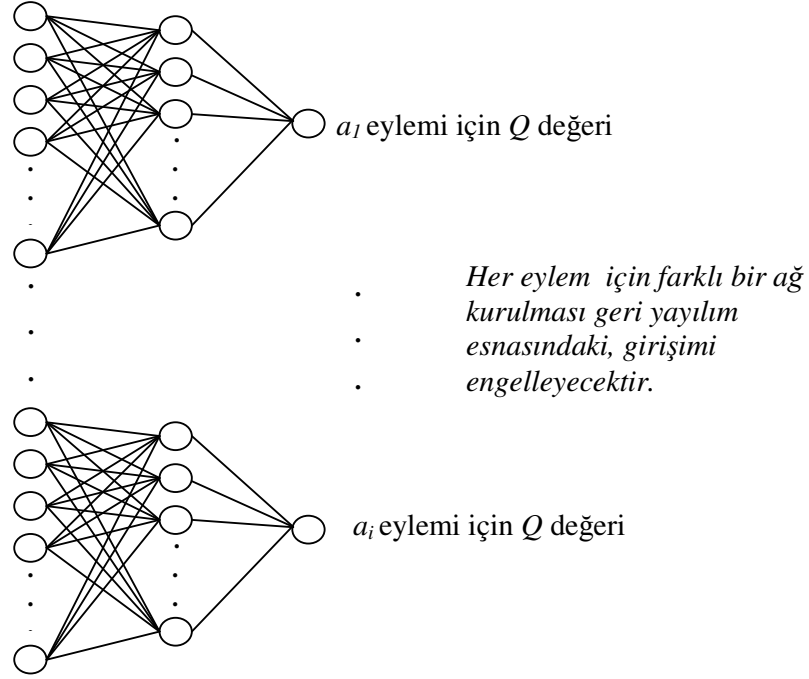
Şekil 3.10. Q-değeri ve Geri Yayılım Ağları

Bu yapının genel işletim prosedürü Çizelge 3.6'daki gibidir:

Çizelge 3.6. Geri Yayılım Ağıyla Q-Öğrenme İşletim Prosedürü

1. $s \leftarrow \text{anlık durum}$; her a eylemi için, $Q_a \leftarrow Q(s, a)$
2. $a \leftarrow \text{seç}(U, T)$;
3. a eylemini çalıştır; $(s', r) \leftarrow$ yeni durum ve pekiştirme
4. $Q' \leftarrow r + \gamma \cdot \text{Max}\{Q(s', k \mid k \in \text{eylemler})\}$
 Yarar ağını geri yayılım hatası ΔU ile s girişi
 doğrultusunda düzelt, $\Delta Q_a = \begin{cases} Q' - Q_a & a_i = a \\ 0 & \text{veya} \end{cases}$
6. 1. adıma geri dön

Burada görüldüğü gibi, tek bir deneyimle diğer eylemler hakkında fikir sahibi olamayacağımız için, a dışındaki diğer eylemlerde ağ değiştirilmemektedir. Şekil 3.10'de tanımlanan ağ, çoklu çıkış düğümlerine sahip olabilir (her biri bir eylem için), veya her biri tek çıkışa sahip birçok geri yayılım ağ kullanılabilir (her bir eylem için). İkincisi daha çok tercih edilendir. Bunun sebebi aradaki gizli katman aracılığıyla, gerçekleşen bir eylemle birlikte bir değişiklik yapılacağı zaman, ağ bağlantı ağırlıkları tamamen güncelleneceği için, istensin veya istenmesin diğer eylemler açısından da bir değişiklik oluşmaktadır. Eğer her eylem için ayrı bir ağ düşünülecek olursa, oluşan eylem yalnızca kendi ağının güncellenmesini sağlayacaktır. Şekil 3.11'de her eylem için ayrı bir ağın kullanıldığı yapı görülmektedir.



Şekil 3.11. Her eylem için ayrı bir ağ

3.2.6.3. Öğreticili Q-Öğrenme

Öğretim insanın öğrenmesinde kritik bir role sahiptir. Çoğu zaman öğretim öğrenme zamanımızı kısaltır ve çözülemez görevleri çözülebilir duruma getirir. Eğer öğrenme bir arama problemi (Mitchell, 1982) olarak düşünülürse, öğretim, bir anlamda bu arama için bir harici rehber görevi yapmaktadır. Öğretim iki açıdan faydalıdır: Birincisi, öğreneni hedef durumlarının bulunduğu arama uzayının gelecek vaat eden bölümlerini araştırmaya yönlendirir. Bu arama uzayının çok büyük olduğu ve tam aramanın uygun olmadığı durumlarda çok önemlidir. İkincisi öğrenenin yerel bir maksimumda sıkışıp kalmasını engeller. Öğretim öğreticili Q-öğrenmede şu şekilde gerçekleştirilir:

- Öncelikle öğretmen belirli bir başlangıç aşamasından hedef bir göreve nasıl ulaşabileceğini gösterir. Gösterilen eylemler, durum geçişleri ve alınan pekiştirmeler bir *öğretim dersi* olarak kaydedilir.
- Çeşitli öğretim dersleri toplanır ve deneyim dersleri gibi tekrarlanır.
- Deneyim dersleri gibi öğretim dersleri de dikkatli seçilmelidir, yalnızca davranış biçimine uygun olanlar seçilmelidir. Ancak davranış biçiminin en uygun olduğu biliniyorsa bütün dersler tekrarlanabilir.

- Aslında bilişsel eleman hem olumlu hem de olumsuz örneklerle öğrenebilir. Bu özellik bu yaklaşımı öğreticili öğrenmeden farklı hale getirmektedir (Mozer 1986, Pomerleau 1989) Öğreticili öğrenmede, öğrenen bilişsel eleman durumlardan gösterilen eylemlere doğru bir eşleştirme oluşturarak ve eşleştirmeye genelleştirerek öğreticiyi taklit etmeye çalışır. Sonuç dezavantajları şu şekilde oluşur:
 - Öğretici, bilişsel eleman tarafından karşılaşılabilecek tüm durumları içeren çok sayıda eğitim örnekleme oluşturmak zorundadır.
 - Yeni bir durumla karşılaşıldığı ve ajanın ona göre bir stratejisi olmadığı durum için öğretici çözümü vermek için hazır bulunmak zorundadır.
 - Öğretici bir uzman olmak zorundadır, aksi takdirde bilişsel eleman en uygun stratejiyi öğrenemeyecektir.

Fakat öğreticili Q-Öğrenme bu dezavantajlardan hiçbirisine sahip değildir.

3.3. Problem Tanımı

Bu tez çalışmasında amaçlanan araç sürücü yardımcı sistemleri için bir yöntem önerilmektedir. Araç sürücü yardımcı sistemlerinin oluşturulabilmesi için bilişsel mimarinin en uygun seçeneklerden biri olduğu kaynak araştırmasında gösterilmiştir. Bilişsel mimari, matematik probleminden havaalanı uçuş kontrol problemine, araç sürme probleminde, çoklu robot uygulamalarına kadar pek çok alanda başarıyla kullanım alanı bulmuştur.

Sürücü modellemesi alanında yapılan çalışmalarda ise Aasman (1995), kavşağa giren araç sürücüsünü bilişsel mimari ile modellemiş ve daha sonra Salvucci ve arkadaşları (2002), sürücü davranışlarını modellemede, algılama ve bilişsel işlemlerini sıralı görev şeklinde düzenleyerek dikkat yönetimi ile birleştiren, ACT-R adını verdikleri bilişsel mimarinin kullanılmasını önermiştir. Bu çalışmalarda iki temel sorun vardır:

- Bu çalışmalarda problem bir bütün olarak düşünülmemiş ve daha çok araç sürme görevinin parçaları üzerinde çalışılmıştır. Örneğin Aasman, sadece

kavşağa girme durumunu modellerken, diğer çalışmada düz bir yolda sadece sollama olayının modellenmesi söz konusudur.

- Bu çalışmalarda görev sıralı bir dizi olarak düşünülmüştür, fakat günlük insan davranışlarının sıralı bir yapıya sahip olduğunu söylemek pek doğru olmayacaktır.

Bu çalışmada önerilen mimari, araç sürme problemini bütün olarak ele almış, yalnızca eyleme dönüşen sistemi (eylem merkezli sistem) hem alt hem de üst seviyede incelemeyi amaçlamıştır. Fakat bu mimaride, özellikle insan davranışı söz konusu olduğunda, alt seviyenin çekirdeğini oluşturan pekiştirmeli öğrenme yönteminin halen yaşadığı bir takım sorunlar mevcuttur. Özetle bunlar:

- Gerçek hayatın sürekli doğasından dolayı, sonsuz sayıda anlamlı durum söz konusudur. Sürekli durum uzayı, şu anki teknikler açısından zorluklar getirir (Sutton ve Barto 1998). Araç sürme olayında durum, aracın kendi pozisyonu ve hızıyla birlikte ortamdaki diğer araçların veya nesnelerin de pozisyon ve hızlarından oluşur. Yalnızca bir şeritteki yanal pozisyonların sonsuz sayısını düşünürsek, pozisyondaki her değişim, ne kadar küçük olursa olsun, en uygun davranışı etkileyebilir. Dahası, eylemler de sürekli ve onları basitçe ayıklaştırmak yeterli değildir. Bir şeritte devam ederken, dönüş açılarının herhangi bir ayık seti, şeridin merkezinde ilerlemeye çalışan aracın salınımına sebep olabilir. Böylece, sürekli durumun ifadesine yaklaşabilmek için, ayık kümenin çok geniş tutulması gerekir. Ayık kümeyi genişletmek, durum uzayının da büyümesi anlamına gelir. Büyük durum ve eylem uzayları için pekiştirmeli öğrenme yöntemleri ve özellikle Q-öğrenme, doğru davranış biçimine yakınsayamamakta veya çok geç yakınsamaktadır.
- Markov özelliği anlık durumun bir sonraki eylemin ne olacağına karar vermek için gereken bilgiyi taşıması anlamına gelmektedir. Araç sürme olayını düşünürsek, anlık durumun bir sonraki hareketin ne olacağına dair bilgiyi tam anlamıyla taşımadığını görebiliriz. Örneğin, bize göre sol şeritte ve arkada ancak bize yakın olarak niteleyeceğimiz başka bir aracın varlığını düşünelim. Algılayıcılardan gelen bilgiler, o aracın yalnızca yakın olduğunu söylemektedir, fakat o araç uzun bir süredir aynı yakınlıkta olabilir, hızla o yakınlığa gelmiş ve yanımızdan geçmek üzere olabilir, veya biz onun

yanından geçmiş olabiliriz. Bu durumda onun şeridine girmek için vereceğimiz karar, onun o an nerede olduğundan çok, geçmişte nasıl bir hareket dizisi izleyerek oraya geldiğine bağlıdır. Bu Markov özelliğinin ihlali anlamına gelmektedir. Algılayıcıların anlık durumlar arasındaki farkı algılayamamasına algısal örtüşme (perceptual aliasing) adı verilmektedir (Whitehead ve Ballard 1991).

- Bir MDP eğer ödül fonksiyonu ve durum geçiş olasılıkları zamandan bağımsızsa stasyonerdir, görevin kendisi zaman içinde değiştiğinden, ortada *stasyoner olmayan* bir görev dağılımı vardır. Bir sürücü, trafikte deneyim sahibi oldukça, görevlerin şekillerinde değişiklikler gözlenmektedir.
- Bilişsel eleman tüm durumları ziyaret eder ve tüm eylemleri sonsuz sayıda seçerse, davranış biçimi en uygun davranış biçimi sınırına, ve Q kestirimi de 1 olasılıkla en uygun Q-fonksiyonu olan Q^* 'ya yakınsar. Fakat pratikte, özellikle araç sürme olayı için, her durum ve eylemi sonsuz sayıda ziyaret etmek pek uygulanabilir değildir.
- Pekiştirme fonksiyonunun elle belirlenmesi oldukça zordur. Araç kullanma örneği düşünülürse, sürüş anında, güvenli takip mesafesini korumak, kaldırımdan uzak kalmak, yayalardan uzak kalmak, mantıklı bir hızı korumak, belki orta şeritte ilerleme tercihi, şeritleri çok sık değiştirmemek vb. gibi istenen veya istenmeyen durumlar arasından seçimler yapılabilir. Sürüş olayındaki pekiştirme fonksiyonunu belirlemek için, bu farklı durumlar arasında nasıl seçim yapılacağını açıkça belirten ağırlık atamaları yapılmalıdır. Aracın yetenekli bir şekilde sürülebilmesinin sağlanmasına rağmen, '*düzgün sürüş*' için güvenilir bir pekiştirme fonksiyonun belirlenebileceği şüphelidir. Örneğin, sürüş anında maksimize edilmek istenen 'doğru' pekiştirme fonksiyonu, belki, 'kişisel memnuniyet' olabilir. Burada uygulamadaki sorun bu fonksiyonun, pekiştirmeli öğrenmenin uygulanabileceği şekilde, durumların bir fonksiyonu şeklinde nasıl ifade edilebileceğidir.

Bu çalışmada, ele alınan bilişsel mimarinin sürücü yardımcı sisteminin oluşturulması amacıyla insan davranışını modelleyebilmesi için varolan problemlere çözümler önerilmiştir. Bu çözümler özetle:

- Alt görevlere bölünerek durum uzayının küçültülmesi amacıyla yeni bir yöntemin önerilmesi, bu yöntemin mimariye adapte edilmesi ve öğrenmenin yakınsamasının sağlanması
- Pekiştirmeli öğrenme yönteminin pekiştirme fonksiyonun gözlenen diziler aracılığıyla otomatik olarak belirlenmesini sağlayan yöntemin ele alınan mimariye adapte edilmesi
- Geçmiş zaman bilgisinin anlık durum bilgisi dahilinde ele alınmasını sağlayacak yapının mimariye adapte edilmesi ve Markov olmayan ortamlarda da çalışmanın sağlanması

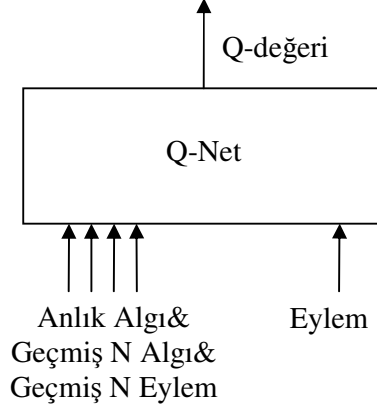
Bu çözüm önerileri, ele alınan mimariye uyarlanmış, daha sonra deneysel olarak test edilmiş ve sisteme katkıları gösterilmiştir.

3.4. Bilişsel Mimari ve Önerilen Çözüm Yolları

3.4.1. Markov Özelliğine Sahip Olmayan Ortamlarda Pekiştirmeli Öğrenme

Markov özelliğine sahip ortamlarda sistemin girişi olarak düşünülen anlık durum, bilişsel elemanın doğru karar verebilmesi için ihtiyacı olan tüm bilgiyi içerir. Fakat her ortam Markov özelliğine sahip olmayabilir. Markov özelliğinin geçerli olmadığı yani anlık durum bilgisinin bir çok farklı anlama sahip olduğu duruma algısal örtüşme adı verilmektedir (Whitehead ve Ballard 1991). Araç sürme olayından örnek verecek olursak, bize göre sol şeritte ve arkada başka bir aracın varolduğunu düşünelim, aracın o anlık pozisyon bilgisi, bir sonraki pozisyonunun ne olacağına ait bilgiyi bize sağlamaz, yani araç bize yaklaşmakta olabilir, bir süredir aynı pozisyonda devam ediyor olabilir veya onu geçmiş ve uzaklaşıyor olabiliriz. Bu durumda sistemin girişi olarak düşünülen anlık durum bilgisinin, yeterince bilgilendirici hale gelmesi sağlanmalıdır. Lin (1992), Q-öğrenme ağına giriş olarak aldığı anlık durum ifadesine geçmiş durum bilgisini de ilave etmiş ve giriş kümesini bu şekilde zenginleştirerek

Pencere-Q mimarisini önermiştir (Şekil 3.12). Bu mimaride geçmiş N adet bilgi anlık durum bilgisinin içine yerleştirilecektir.



Şekil 3.12. Pencere – Q Mimari

Pencere-Q mimarisinin özelliği geçmiş bilgiye olan ihtiyacın sığ olduğu durumlarda avantaj sağlamasıdır. Lin'in gerçekleştirdiği deneysel çalışmalarda, Pencere-Q mimarisinin genel olarak çok doğru çalışmadığı görülmüştür. Çünkü pencere boyutu N en uygun davranış biçimini temsil için yeterince büyük seçilse bile, durumları ayırt etme konusunda başarısız olabilmektedir. Bu çözüm, bölüm 3.4.2' de değinilecek olan 'seçenekler' yapısı çerçevesinde biraz daha geliştirilerek performans açısından kullanılabilir hale getirilecektir.

3.4.2. Süreklilik – Büyük Durum Uzayı, Durum Uzayının Parçalanması ve Seçenekler Yapısı

İnsan karar verme mekanizması, alt-görevler arasından, veya farklı bir deyişle belirli zaman dilimine yayılmış olan eylemler dizisi arasından seçim yapmayı içermektedir. Örneğin araç sürme olayını ele alırsak, birkaç farklı alt görev vardır ve tüm araç sürme olayı, aslında, bu alt görevlerin arka arkaya sıralanmasından oluşmaktadır. Bu alt görevler aracın durumunu korumak olabilir, şerit değiştirmek olabilir, veya araç sollamak olabilir. İnsan tarafından gerçekleştirilen görevlerin veya olayların bu şekilde çeşitli alt görevlere bölünerek uzayın küçültülmesi ve hepsinin paralel ve bağımsız olarak öğrenilmesi yöntemi, durum uzayının küçültülmesi ve Q-öğrenmenin yakınsamasının sağlanabilmesi için en önemli yaklaşımlardan birisini

oluşturmaktadır (Dayan ve Hinton 1993, Dietterich 2000, Sutton ve ark. 1999, Barto ve Mahadevan 2003). Bu alandaki çalışmalar genel olarak hiyerarşik pekiştirmeli öğrenme, Hiyerarşik parçalama, Seçenekler, Makro-eylemler ve geçici soyutlama gibi isimler altında toplanmaktadır. İsimler farklı olsa da hepsinde amaç aynıdır, düşük seviyeli eylemleri daha yüksek seviyeli akıl yürütmelerle soyutlamak, veya başka bir deyişle ele alınan görevleri alt-görevlere veya alt-hedeflere bölmektir. Bu yaklaşım öğrenme problemini iki şekilde basitleştirir. Birincisi, her alt-görev kendini ilgilendiren durumları göz önünde bulunduracağı için durum uzayının etkili boyutu küçültülmüş olur. İkincisi, alt görevlerin öğrenilmesi çok daha hızlanmış olacağı için, toplam öğrenme süresi dikkate değer şekilde düşürülmüş olacaktır. Bu çalışmalar arasında Sutton tarafından öne sürülen Seçenekler yapısı yaygın olarak kabul görmüş ve çeşitli deneylerle kendisini ispatlamış bir yöntemdir (Sutton ve ark. 1999).

Sutton, en basit eylemlerden yola çıkarak oluşturduğu, eylemlerin geçici olarak genişletilmiş dizilerine seçenekler adını vermiştir. Seçenekler üç temel bileşen içerir: Davranış Biçimi, $\pi : S \times A \rightarrow [0,1]$, Bitiş Koşulu, $\beta : S^+ \rightarrow [0,1]$ ve Başlangıç Koşulu, $I \subseteq S$. Bir $\langle I, \pi, \beta \rangle$ seçeneği, yalnızca $s_t \in I$ olduğu durum için s_t durumunda kullanılabilir. Eğer seçenek çalıştırılırsa, artık eylemler, β bitiş koşuluyla karşılaşıp seçenek sonlandırılana kadar π davranış biçimine göre eylem seçer. Bir Markov seçeneği şu şekilde davranacaktır. Öncelikle, bir sonraki a_t eylemi şu şekilde çalıştırılır: Öncelikle, bir sonraki a_t eylemi, $\pi(s_t, \cdot)$ olasılık dağılımına göre seçilir. Ortam kendini s_{t+1} durumuna geçirir, bu noktada seçenek $\beta(s_{t+1})$ olasılığıyla sonlanabilir veya $\pi(s_{t+1}, \cdot)$ 'e göre bir a_{t+1} eylemi seçer ve bu şekilde sonlanana kadar devam eder. Seçenek sonlandığı zaman, bilişsel eleman yeni bir seçenek seçebilme şansına sahip olur.

Seçeneğin başlangıç ve bitiş koşulu birlikte uygulamanın çalışma alanını kısıtlarlar. Yalnızca seçeneğin davranış biçiminin ihtiyaç duyduğu durumlar çalışma alanını belirler. Bu durum için π tüm s uzayı yerine I uzayında tanımlı olacaktır.

Bazı durumlarda, seçenekler için bir süre-sonu tanımlamak da faydalı olacaktır, belirli bir süre geçmesine rağmen halen belirlenmiş bir duruma gelinmediyse, seçeneği sonlandırmak gerekebilir. Bu Markov seçenekleriyle mümkün değildir. Çünkü sonlandırma kararları yalnızca anlık duruma dayanarak yapılmaktadır, ne kadar zamandır seçeneğin devam ettiğinin bir önemi yoktur. Bu sorunu çözebilmek için, yarı-Markov seçenekleri kavramı ortaya atılmıştır. Yarı-Markov seçeneklerinde davranış biçimleri ve sonlandırma koşulları, seçeneğin başlatılmasından sonra oluşan tüm geçmiş olaylara bağlı olarak seçim gerçekleştirilmektedir. Genel olarak, seçenek, bir t anında başlatılır, k adım devam eder ve s_{t+k} durumunda sonlandırılır. Her ara τ zamanında, $t \leq \tau < t+k$, bir Markov seçeneğinin kararları yalnızca s_τ durumuna bağlıdır, fakat yarı-Markov seçeneğinin kararları, tüm geçmiş $s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}, \dots, r_\tau, s_\tau$ dizisine bağlıdır. Ancak s_t 'den önceki veya s_τ 'dan sonraki olaylara bağlı olmayacağını da belirtelim. Bu diziye t 'den τ 'ya kadar olan geçmiş olarak adlandırılır ve $h_{t\tau}$ ile temsil edilir. Tüm geçmişlerin kümesi ise Ω ile gösterilir. Yarı-Markov seçeneklerde, davranış biçimi ve sonlandırma koşulu olası geçmişlerin fonksiyonlarıdır, yani, $\pi: \Omega \times A \rightarrow [0,1]$ ve $\beta: \Omega \rightarrow [0,1]$.

Verilmiş bir seçenekler kümesinde, başlangıç kümeleri her $s \in S$ durumu için bir O_s uygun seçenekler kümesi tanımlıdır. Bu O_s kümesi tıpkı uygun eylemlerin bulunduğu A_s kümesine benzer. Her a eylemi, a 'nın uygun olduğu durum için ($I = \{s: a \in A_s\}$), her zaman tek adımda sonlanan ($\beta(s) = 1, \forall s \in S$) ve her zaman a eylemini ($\pi(s, a) = 1, \forall s \in I$) seçer. Bundan dolayı bilişsel eleman seçimini her zaman adımı için seçenekler arasından yapar, bazıları tek zaman adımında tamamlanır, bazıları ise geçici olarak uzatılmıştır. Öncekine tek-adımlık veya basit seçenek adı verilirken, diğeri çok-adımlı seçenek olarak adlandırılır.

Daha ilginç bir tanımlama seçenek seçimi için davranış biçimleri kavramıdır. s_t durumuyla başlatıldığı zaman, seçenekler üzerindeki Markov davranış biçimi, $\mu: S \times O \rightarrow [0,1]$, $\mu(s, \cdot)$ olasılığıyla bir $o \in O_s$ seçeneği seçecektir. Her yeni seçenek seçildiğinde, s_t durumunda o seçeneği başlatılır,

s_{t+k} durumunda son bulana kadar $\mu(s_{t+k})$ dağılımına göre eylemler belirlenir. Yarı-Markov seçeneklere benzer şekilde, bu tarzdaki davranış biçimlerine de yarı-Markov davranış biçimi adı verilmektedir.

Bu fikirler verilen bir davranış biçimi için geleneksel değer fonksiyonlarının doğal genelleştirmelerine götürür. Bir yarı-Markov davranış biçimi altında bir $s \in S$ durumunun değerini, π 'nin s 'te başlatıldığı durum için, şu şekilde tanımlayabiliriz:

$$V^\pi(s) = E\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots \mid \varepsilon(\pi, s, t)\} \quad 3.28$$

burada $\varepsilon(\pi, s, t)$ t anında s durumunda başlatılan π davranış biçiminin olayını temsil etmektedir. Eylem-değer fonksiyonları, seçenek-değer fonksiyonlarına dönüşür. Bu aşamada, $Q^\mu(s, o)$, $s \in S$ durumunda o eylemini seçmenin değerini tanımlayabiliriz:

$$Q^\mu(s, o) = E\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots \mid \varepsilon(o, \mu, s, t)\} \quad 3.29$$

burada $o\mu$, sonlana kadar o seçeneğini takip eden ve sonlandıktan sonra μ vasıtasıyla yeni bir seçim yaparak başlayan yarı-Markov davranış biçimini temsil etmektedir. Yarı-Markov seçenekler için $\varepsilon(o, h, t)$, t anından itibaren h boyunca devam eden, burada h s_t ile biten geçmişi temsil etmektedir, olay tanımı yapmak faydalı olacaktır. Yani, a_t eylemi $o(h, \cdot)$ 'a göre seçilir, ve o $t+1$ anında $\beta(h a_t, r_{t+1}, s_{t+1})$ olasılığıyla sonlanır; eğer o sonlanmazsa, bu durumda a_{t+1} $o(h a_t, r_{t+1}, s_{t+1})$ 'e göre seçilir ve böylece devam eder. Bu tanımla, (5) s 'i bir durum olmasının yanında bir de geçmiş olarak tutar.

3.4.2.1. SMDP VE Q-Öğrenme

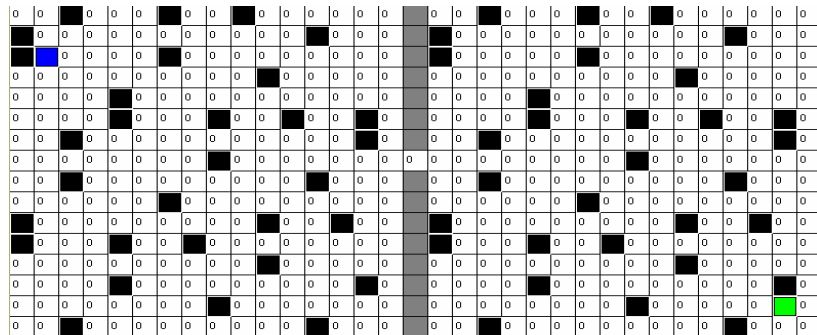
Bir o seçenekler kümesi üzerinden en uygun davranış biçimini bulma problemi, öğrenme yöntemleri ile de çözülebilir. o seçeneği s durumunda çalışmaya başladığı zaman, bir sonraki adımda o 'nun sonladığı s' durumuna geçilir. Bu tecrübeye dayanarak, bir yaklaşık seçenek-değer fonksiyonu, $Q(s, o)$ güncellenir. Örneğin, Q-öğrenmenin, SMDP şekli, SMDP Q-öğrenme olarak adlandırılacağı şekilde, her seçenek sonlanmasından sonra

$$Q(s,o) \leftarrow Q(s,o) + \alpha[r + \gamma^k \max_{o' \in O_s} Q(s',o') - Q(s,o)] \quad 3.29$$

değeri güncellenir, burada k , s ile s' arasında geçen zaman adımlarını temsil ederken, r bu zaman içindeki toplam indirimli ödülü temsil eder.

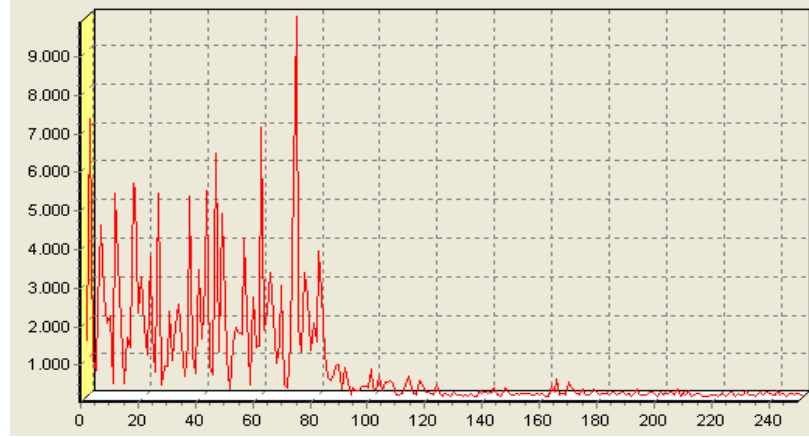
3.4.2.2. Alt hedeflerin Kullanımı ve Gridworld Ortamı Deneyi

Problem uzayının alt hedeflere bölünerek pekiştirmeli öğrenme yönteminin performansının ne kadar arttığı, literatürde çok sık rastlanan çok temel bir problem olan şekil 3.13'teki iki odalı bir gridworld problemi kullanılarak incelenebilir. Hücrelerin her birinin pozisyonu, ortamın bir durumuna karşılık gelmektedir. Herhangi bir durumda yani pozisyonda bilişsel eleman yukarı, aşağı, sağa, sola olmak üzere dört yönde hareket edebilir. Burada seçeneklerin kullanılmadığı ve seçeneklerin kullanıldığı iki ayrı problemi inceleyebiliriz. Birinci problemde başlangıç hücresi birinci odada ve pozisyonu (2,1), varış hücresi ise ikinci odada ve pozisyonu yani durumu (14,14) şeklindedir. Bilişsel elemanın hedefi, başlangıç hücresinden varış hücresine ulaşabildiği en kısa yoldan ulaşmaktır. Siyah hücreler ise yol üzerine yerleştirilmiş engelleri temsil etmektedir. Belirlenen yolun bu engellerin üzerinden geçmemesi gerekmektedir. Üzerinden geçtiği her engel bilişsel elemana ceza getirirken, hedefe varması ödülle pekiştirilmektedir. Tamamen rasgele seçilen olasılıklarla, bilişsel eleman 0.9 olasılıkla bulunduğu durumdan en yüksek değere sahip yeni duruma doğru hareket edecektir, 0.1 olasılıkla diğer üç durumdan birini rasgele olarak seçecektir. Şekil 3.13'te de görülebildiği gibi başlangıçta tüm hücrelerin Q-değerleri sıfır kabul edilmiştir.



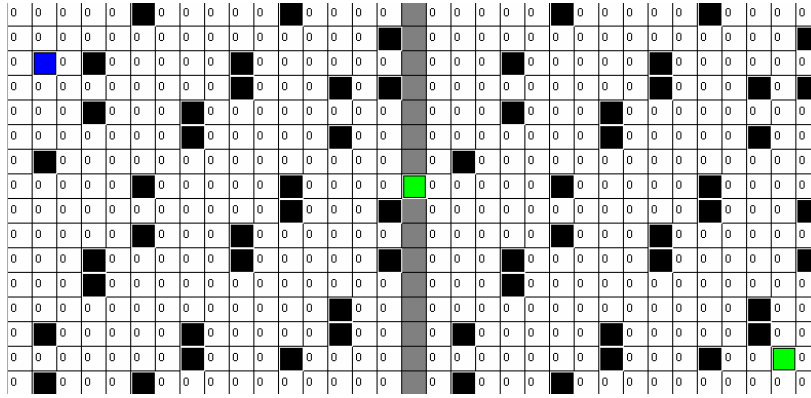
Şekil 3.13. Gridworld Problemi

Burada asıl problem şekil 3.16'da görülebilir. Bilişsel eleman, problem son derece basit olmasına rağmen, başlangıç hücresiyle bitiş hücresi arasındaki en uygun yola yakınsayabilmek için çok fazla çaba harcamaktadır. Şekil 3.16'da görülebildiği gibi ilk adımlarda 5000'ler mertebesinde adım atarak hedefe varmaktadır. Şekil 3.14'ten de görülebileceği gibi aslında bu 42 adımlık bir yoldur. En uygun yola yakınsayabilmek için bilişsel eleman 250 döngüde yaklaşık 350.000 – 400.000 civarında adım atmaktadır. Ama görülebildiği gibi aslında en uygun yola yakınsamaya doğru 100-120. adımdan sonra önemli bir kırılma gerçekleştirilmektedir. Bilişsel eleman en uygun yola yakınsamada attığı adımların büyük kısmını bu kırılmaya kadar atmaktadır.



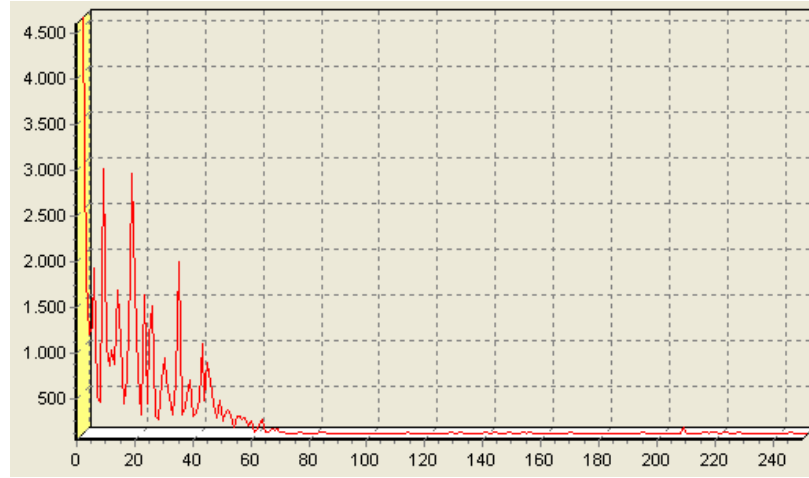
Şekil 3.16. Hedefe varmak için her denemede bilişsel elemanın kaç hücreden geçtiği, yani her denemede çıkıştan hedefe kaç adımda vardığı

İkinci problemde, birinci problemden farklı olarak gridworld problemi iki farklı problem haline dönüştürülmüştür. Bunu gerçekleştirebilmek için iki oda arasına şekil 3.17'deki gibi bir geçiş hücresi yerleştirilmiştir ve bilişsel elemanın oda değiştirebilmesi için o hücreden geçmesi gerekmektedir. Seçenekler yapısına göre birinci odanın iç davranışı bilişsel elemanı başlangıç noktasından geçiş hücresine doğru gidebileceği en kısa yoldan götürmelidir. İkinci odanın iç davranışı ise bilişsel elemanı geçiş hücresinden varış hücresine doğru en kısa yoldan götürmelidir.



Şekil 3.17. Seçenekler yapısı çerçevesinde problem, iki pekiştirmeli öğrenme problemi şekline dönüştürülmüştür.

250 döngü sonunda bu yöntemle de bilişsel elemanın en uygun davranış biçimine yakınsadığı görülmüştür. Fakat iki noktada farklılık mevcuttur. Bu farklılıklar Şekil 3.18'de net bir şekilde görülebilmektedir. İlk farklılık, önceki çözüm yöntemine göre bilişsel elemanın çok daha az adım attığı yani çok daha az çaba harcadığıdır. Bu yöntemde bilişsel eleman 50.000 – 60.000 civarında adım atarak en uygun değere yakınsamaktadır. İkinci farklılık ise bu yöntemle en uygun değere çok daha hızlı yakınsanmasıdır. Yaklaşık 50. adımda kırılma gerçekleşmekte ve bilişsel eleman en uygun değerine yakınsamaktadır. Bu örnek, alt göreve bölümlenmenin pekiştirmeli öğrenme yöntemi açısından ne kadar performans artırıcı olduğunu göstermektedir. Burada problem domeni son derece basit ve problem yalnızca ikiye bölünmüş olduğu halde performansın ne kadar arttığı Şekil 3.18 ve Şekil 3.16 karşılaştırarak görülebilmektedir. İnsan davranışının söz konusu olduğu durumda, problemin ikiden fazla probleme dönüşeceği düşünülürse, yöntemin sisteme katkısı tartışılmaz olacaktır.



Şekil 3.18. Yöntemin hedefe varmak için her denemede bilişsel elemanın kaç hücreden geçtiği

Ele alınan problem, tek bir pekiştirmeli öğrenme problemi olarak düşünüldüğü takdirde bu, çözüm açısından tek seviyeli bir problem anlamına gelecektir. Gridworld örneğinde problemin iki pekiştirmeli öğrenme problemine dönüştürülerek, tek seviyede çözümlenen problemin iki alt seviyeye bölünerek iki seviyede de ayrı ayrı çözümlendiği görülmektedir. Burada problemin bir bütün olarak düşünüldüğü seviye yüksek seviye, bütünüün parçalarını oluşturan her alt görev ise bu üst seviyenin alt seviyesi anlamına gelmektedir. Örneğimizde, çıkış hücresinden varış hücresine hareket görevi üst seviyede yer alırken, her odadaki hareket bu bütünüün alt seviyesini oluşturmaktadır. Hiyerarşik bakış açısıyla, araç sürme görevi de alt seviyelere bölünebilir, bölünen alt seviyeler de daha sonra kendi içinde alt seviyelere bölünebilirler, ve bu artık bölünebilecek alt seviye kalmayınca kadar devam eder. Örneğin, araç sollama görevi, araç sürme görevinin alt görevlerinden biri olarak düşünülebilirken, sol şeride geçme görevi de araç sollama görevinin alt görevlerinden biri olarak düşünülebilir.

Bu tez çalışmasında, bu hiyerarşik bölümlenmeyle ilgili, varolan sisteme bir katkı olarak düşündüğümüz, insan davranışı gibi karmaşık yapılarda kullanılacak, alt görevlerin otomatik olarak belirlenmesini sağlayan yeni bir yöntem geliştirilmiştir.

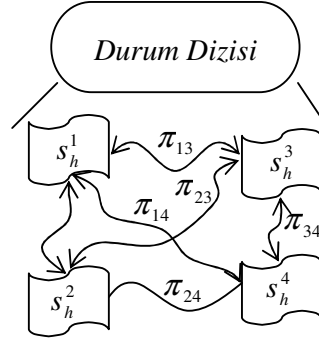
3.4.3. Seçenekler ve Israrcı Aralıklardan Alt Görev Belirleme Yöntemi

İnsan bir eylemi gerçekleştirirken, her zaman adımında nedensiz olarak bulunduğu konumu değiştirmez. Örneğin, araç sürme olayında, sürücü aracıyla belirli hızda seyrederken, her zaman adımında nedensiz olarak hızını sürekli azaltmaz veya arttırmaz. Başka bir örnek olarak sürücünün orta şeritte gittiğini düşünelim, nedensiz yere sürücü sağ şeride veya sol şeride geçmeyecektir. Fakat bir pekiştirme olduğu durumda, örneğin yolda ilerlerken önündeki araca fazla yaklaşırsa, ya hızını düşürecek veya şeridini değiştirecektir. Bir pekiştirme oluşmadan sürücünün içinde bulunduğu durumu koruması için hiçbir şey yapma eylemi de bir çıkış olarak düşünülebilir. Bu çalışmada, insan davranış dizilerinde gözlenen hiçbir şey yapma eylem çıkışının meydana getirdiği aralıklar, kararlı durum aralıkları olarak tanımlanmıştır. Sezgisel olarak, insanın bir an önce kararlı bir duruma geçmeye çalıştığı ve herhangi bir pekiştirmenin oluşmadığı durumda bu durumunu korumak istediği söylenebilir. Bir pekiştirme olduğu zaman yapmaya çalıştığı ise bir kararlı durumdan başka bir kararlı duruma geçmektir.

Davranış bilimleri alanında bu kararlı durumlara 'davranış ısrarcılığı' adı verilmektedir (Mace 1996). Davranışsal momentum üzerindeki temel ve uygulamalı araştırmalarda belirli bir durum için, belirli davranışlarda ısrar etmekle gözlenen pekiştirme arasında genel bir davranışsal ilişki olduğu gösterilmiştir. Sonuç olarak, insan bir itici koşul oluşmadığı sürece genel olarak içinde bulunduğu üst seviye davranışını korumak isteyecektir.

Bu bakış açısından yola çıkarak insan davranış dizilerindeki davranışsal ısrar aralıklarının bulunmasının, üst seviye davranışlarının arasındaki geçiş durumlarını yani alt görev sınırlarını bulmak anlamına geleceği söylenebilir. Bu çalışmada, davranış dizisi içinde bu ısrarcı aralıkların keşfedilmesini sağlayan doğal bir yöntem önerilmektedir. Bilişsel eleman, $n = 0,1,2,3,\dots$ için, bir s_h^n üst seviye davranışında bir pekiştirme oluşmadığı takdirde ısrarcı olacaktır. Pekiştirme, bilişsel elemanın, s_h^n üst seviye davranışından başka bir üst seviye davranışına geçmesini sağlayacaktır. Bu

durumlar arasındaki geçiş, yerel davranış biçimi tarafından şekillendirecektir. Temel yapı şekil 3.19' da görülebilir.



Şekil 3.19. Dört ısrarcı üst seviye davranışına sahip bir durum dizisi.

Bilişsel eleman zamanının büyük kısmını, bu ısrarcı üst seviye davranışlarını takip ederek geçirir. Bir pekiştirme oluşmadığı sürece bilişsel eleman bu durumu korur. Üst seviye davranışlar arasındaki bağlantılar, yerel davranış biçimi fonksiyonu tarafından oluşturulan geçişleri temsil etmektedir.

3.4.3.1. Alt Görev Sınırlarının Otomatik Olarak Belirlenmesi

İnsanın bir ön bilgi sahibi olmadan bir görevi nadiren veya mecbur kaldığından gerçekleştirdiğini söyleyebiliriz. Görevi gerçekleştirmeden önce, insan görevle ilgili bilgi toplar. Görev gösterimi, insanlara görevin nasıl yapılacağını öğretmede kendini ispatlamış bir yöntemdir (Amit ve Mataric 2002). Bu çalışmada, insanların kendi arasında kullandıkları öğrenme yöntemlerinden biri olan göstererek öğretme yaklaşımı kullanılmıştır: Öncelikle bir gösterim gerçekleştirilir, daha sonra öğrencinin gösterilmiş diziler üzerinden pratik yaparak öğrenmeyi gerçekleştirmesine izin verilir. Durum dizileri $s_t, s_{t+1}, s_{t+2}, \dots$, yani insan davranışlarından gözlenen durum dizileri, kararlı durumları veya ısrarcı aralıkları belirlemek için bir araya getirilir.

3.4.3.2. İsrarcı Durumların Belirlenmesi

Tercüme edilebilir ve uzman açısından faydalı bilginin belirlenebilmesi için, sonuç aralıklarının ele alınan dizi içerisinde bir anlam taşıması önemli bir gerekliliktir. İnsan davranış dizileri bir zaman serisi olarak düşünülebilir. Eğer zaman serisi, arka planda sürekli tekrarlanan durumlar tarafından oluşturulan

aralıklardan oluşuyorsa, bu dizi içindeki değer boyutları bu durumları açıklayabilir. En yaygın kullanılan bölümlenme yöntemleri eş-uzunluklu veya eş frekanslı histogram yöntemleridir. Bu çalışmada, Mörchen ve Ultsch (2005) tarafından önerilen, durum dizilerindeki ısrarcı aralıkları belirlemek için geçici bilgiyi hesaba katan ısrarcılık belirleme yöntemini, kendi amacımıza uyumlu hale getirdik. Aşağıda yöntemin nasıl amacımıza uygun hale getirildiği açıklanmaktadır.

3.4.3.3. Algoritma

Morchen ve Ultsch (2005) zaman serilerindeki ısrarcılığı belirlemek için bir yöntem önermişlerdir. Önerilen yöntem değerleri geçici olarak değerlendirmeye alır. Bölümlenme ise sonuç durumların ısrarcılığı kullanılarak gerçekleştirilir. $S = \{S_1, \dots, S_k\}$ olası sembollerin bir kümesi ve $s = \{s_i \mid s_i \in S, i = 1 \dots n\}$ S_j sembolünün marjinal olasılığı, $A(j, m) = P(s_i = S_j \mid s_{i-1} = S_m)$ ise geçiş olasılıklarının $k \times k$ matrisi olsun. Kendinden kendine geçiş olasılıkları ise A matrisinin ana diagonalıdır.

Eğer zaman serisinde bir geçici yapı yoksa, semboller, marjinal dağılımlarına göre rasgele değişkenlerin bağımsız gözlemleri olarak tercüme edilebilir. Her sembolü gözlemlenmenin olasılığı, bir önce gözlenmiş sembolden bağımsızdır, yani, $P(s_i = S_j \mid s_{i-1}, \dots, s_{i-m}) = P(S_j)$. Geçiş olasılıkları ise $A(j, m) = P(S_j)$ olur. En basit geçici yapı birinci dereceden Markov modelidir. Her durum yalnızca bir önceki duruma bağlıdır, yani, $P(s_i = S_j \mid s_{i-1}, \dots, s_{i-m}) = P(S_j \mid s_{i-1})$. İsrar bu iki model kıyaslanarak belirlenebilir. Eğer bir geçici yapı yoksa, Markov modelin geçiş olasılıkları marjinal olasılıklara yakın olmalıdır. Eğer durumlar ısrarcı bir davranış gösterirse, kendinden kendine geçiş olasılıkları marjinal olasılıklarından daha yüksek olacaktır. Eğer, belirli bir durumda kalma sürecine girildiyse, geçiş olasılığı karşılık gelen marjinal değerinden daha düşük olur.

İki olasılık dağılımının kıyaslanması için iyi bilinen bir ölçüm Kullback-Leibler ıraksaması yöntemidir (Kullback ve Leibler 1951). $P = \{p_1, \dots, p_k\}$ ve

$Q = \{q_1, \dots, q_k\}$ gibi iki ayrık olasılık dağılımının yönlendirilmiş ve simetrik ölçümleri denklem 1 ve denklem 2'de verilmiştir.

$$KL(P, Q) = \sum_{i=1}^k p_i \log\left(\frac{p_i}{q_i}\right) \quad 3.30$$

$$SKL(P, Q) = \frac{1}{2}(KL(P, Q) + KL(Q, P)) \quad 3.31$$

Bu durumda j durumunun ısrarcılığı ise şu şekilde tanımlanır:

$$Pers(S_j) = \text{sgn}(A(j, j) - P(S_j))SKL(A(j, j), P(S_j)) \quad 3.32$$

bu ifade kendi kendine veya kendi kendine olmayan geçiş ve marjinal olasılık dağılımlarının simetrik Kullback-Leibler ıraksamasının bir gösterge değişkeniyle çarpılmasını göstermektedir. Gösterge değişkeni skorun işaretini belirler. Kendi kendine geçiş olasılıkları marjinal değerlerinden daha yüksek olan durumlar pozitif değer verirken, diğerleri negatif değer üretecektir. Yalnız ve yalnızca olasılık dağılımlarının eşit olması durumunda skor sıfır olacaktır.

Tüm durumlar için özet skoru, durum başına değerlerin ortalaması olarak elde edilebilir. Bu ortalama ısrar kavramını da kapsar, yani, yüksek ısrar skoruna erişebilmek için durumların tümü veya büyük çoğunluğu yüksek ısrarcılığa sahip olmalıdır. İsrarcılık ölçümü olarak, durumları ısrarcı durumlar (bilişsel elemanın bir pekiştirme almadığı sürece kaldığı durumlar) veya ara durumlar (ısrarcı durumlar arasındaki geçişler) olarak iki kategoriye ayırabiliriz. İsrarcı durumları belirlemenin anlamı, seçenekler çerçevesi için darboğazların tespit edilmesi anlamına gelmektedir.

3.4.3.4. İsrarcılık ve Geçiş Durumları

Üst seviye ısrarcı durumları belirlemek için, bir insan öğreticiden gözlenen durum dizileri, $s_t, s_{t+1}, \dots, s_{t+n}$ kullanılmıştır. Gerçek dünyada, bu durumların her biri çok sayıda algısal değişkeni içerebilir. Her zaman adımında, öğrenci gerçek sayılardan oluşan q boyutlu bir vektörü giriş olarak alır. q 'nun değeri algılayıcıların sayısı olabilir ve algılayıcılardan gelen bilgiler birbirleriyle ilintili olmayabilir ve her zaman adımında değişebilir.

Eğer diğer durum değişkenleriyle ilintili olmayan durum değişkenlerimiz varsa, üst seviye davranışları arasındaki geçiş durumlarını temsil etmesi açısından, tüm durum değişkenlerinin katılımıyla oluşan bir durum ifadesi oluşturmak zordur. Araç sürme olayı gibi, gerçek dünya davranışlarında da, deneysel olarak tüm algılayıcıların oluşturduğu bir ortak durum ifadesinden bahsedilemeyeceği görülmüştür. Diğer bir deyişle, deneysel çalışmanın ardından, ısrarcı durumların belirlenmesi açısından, algılayıcılardan gelen verinin kendi değişkenleri arasında bir ilinti olmadığı görülmüştür, örneğin, hızı ve üzerinde bulunduğu şerit arasında bir bağlantı yoktur, herhangi bir hızda ve herhangi bir şeritte bulunabilmektedir, veya öndeki araçla belirli bir mesafeyi korumaktadır ama herhangi bir hızda olabilir. Bundan dolayı, durum değişkenleri diğer durum değişkenlerinden bağımsız olarak düşünülebilir ve her durum değişkeni kendi durum dizisi içinde geçiş durumlarına sahip olabilir. Bununla birlikte, durum değişkenlerinin birbirinden tamamen bağımsız olduğunu söylemek de bir hata olacaktır. Örneğin, hız çok yüksekken direksiyon hızlı bir şekilde çevrilemez, böyle bir hareket kazaya yol açacaktır.

Araç sürme problemi için başlıca dört farklı durum değişkeni kullanılmıştır. Bunlar aracın hızı, etraftaki diğer araçlarla aradaki mesafe, aracın bulunduğu şerit ve aracın yönüdür. Diğer araçlarla aradaki mesafede, diğer aracın aracımıza göre bulunduğu konum itibarıyla altı farklı durum değişkeniyle temsil edilmektedir. Bölüm 3.7'deki deneysel çalışmada durum değişkenlerine ayrıntılarıyla değinilecektir.

3.4.4. Pekiştirme Fonksiyonu ve Çıraklık Dönemi Eğitimi

Markov Karar Verme Süreci (MDP) dahilindeki ardışık karar verme probleminde, en uygun veya en uyguna yakın çözümü bulmak için birkaç standart algoritma mevcuttur. MDP düzeninde, bir pekiştirme fonksiyonun verildiğini varsayarız. Bir pekiştirme fonksiyonu ve MDP'nin durum geçiş olasılıklarıyla, değer fonksiyonu ve en uygun davranış biçimi açıkça belirlenebilir.

Fakat buradaki asıl sorun, özellikle insan davranışlarının söz konusu olduğu bazı durumlarda pekiştirme fonksiyonunun elle belirlenmesinin

zorluğudur. Araç kullanma örneğini düşünelim; sürüş anında güvenli takip mesafesini korumak, kaldırımdan uzak kalmak, yayalardan uzak kalmak, mantıklı bir hızı korumak, belki orta şeritte ilerleme tercihi, şeritleri çok sık değiştirmemek vb. gibi istediğimiz durumlara geçmek veya istemediğimiz durumlardan uzak durmak arasında değişimler yapabiliriz. Sürüş olayı için bir pekiştirme fonksiyonunun belirlenmesi için, bu farklı faktörler arasında nasıl değişim yapılacağını açıkça belirten ağırlık atamaları yapılmalıdır. Aracın yetenekli bir şekilde sürülebilmesine rağmen, 'düzgün sürüş' için güvenilir bir pekiştirme fonksiyonunun belirlenebileceği şüphelidir. Örneğin, sürüş anında maksimize edilmek istenen 'doğru' pekiştirme fonksiyonu, belki, 'kişisel memnuniyet' olabilir. Burada uygulamadaki sorun, bu fonksiyonun pekiştirmeli öğrenme yönteminde uygulanabilecek şekilde ele alınan problemin durumlarının bir fonksiyonu şeklinde nasıl modellenebileceğidir.

Pratikte bunun anlamı, istenen davranış elde edilene kadar, pekiştirme fonksiyonunun çoğunlukla elle bükülmesidir (Ng ve ark. 1999). Şunu rahatlıkla söyleyebiliriz ki, çoğu problem için, pekiştirme fonksiyonunun elle belirlenmesinin zorluğu, pekiştirmeli öğrenme ve en uygun kontrol algoritmasının yaygın kullanımının önünde dikkate değer bir engeldir.

Araç sürmeyi bilmeyen birine sürmeyi öğretirken, ona pekiştirme fonksiyonunun ne olduğunu anlatmaktansa, ona aracın nasıl kullanılacağını göstermek çok daha basit ve doğal bir yoldur. Bir uzmandan öğrenme görevi *çıraklık dönemi öğrenimi* olarak adlandırılır (aynı zamanda izleyerek, taklit ederek veya gösterimden öğrenmek olarak da adlandırılabilir). Pekiştirmeli öğrenmenin, davranış biçimi veya değer fonksiyonundan ziyade, pekiştirme fonksiyonunun ön varsayımına dayandırıldığını düşünerek, çıraklık dönemi eğitiminin pekiştirme fonksiyonun öğrenildiği bir yaklaşıma dönüştürmek en doğal yol gibi görünmektedir.

Gözlemlenen davranıştan pekiştirme fonksiyonunun türetilmesi problemi *tersine pekiştirmeli öğrenme* olarak adlandırılmıştır (Ng ve Russell 2000). Uzmanın (başarmasına gerek olmadan) bilinen "özellik"lerin lineer bir kombinasyonu ile ifade edilebilecek bir pekiştirme fonksiyonunu en uygun hale

getirmeye çalıştığını kabul edelim. Ng, algoritmasının uzmanın doğru pekiştirme fonksiyonuna tam anlamıyla ulaşmayı garanti etmemekle birlikte, yine de uzman kadar iyi performans veren bir davranış biçimini bulduğunu göstermiştir, burada performans uzmanın bilinmeyen pekiştirme fonksiyonuna göre ölçülmektedir (Ng ve Abbeel 2004). Algoritma Prosedürü şu şekildedir:

Çizelge 3.7. Geri Pekiştirmeli Öğrenme İşletim Prosedürü

1. Rasgele bir $\pi^{(0)}$ davranış biçimi seç ve çalıştır, $\mu^{(0)} = \mu(\pi^{(0)})$ 'i hesapla ve $i=1$ yap.
2. $t^{(i)} = \max_{w: \|w\|_2 \leq 1} \min_{j \in \{0, \dots, (i-1)\}} w^T (\mu_E - \mu^{(j)})$ denklemini hesapla burada $w^{(i)}$ eşitliği maksimum yapan w değeri olsun.
3. Eğer $t^{(i)} < \epsilon$, sonlandır
4. RL algoritmasını kullanarak, $R = (w^{(i)})^T \phi$ ödülünü kullanarak $\pi^{(i)}$ en uygun davranış biçimini hesapla.
5. $\mu^{(i)} = \mu(\pi^{(i)})$ 'yi hesapla.
6. $i = i + 1$ yap ve 2. adıma geri dön.

Burada ϕ daha öncede değindiğimiz özellik vektörü, μ özellik beklentisi vektörü ve w ağırlık vektörüdür.

$$\mu_E = \frac{1}{m} \sum_{i=1}^m \sum_{t=0}^{\infty} \gamma^t \phi(s_t^{(i)}) \quad 3.33$$

olarak ifade edilen gözlenen sürücünün beklenti değeridir, burada sisteme giriş olarak kullanılacak veri anlamında, insan sürücü gözlenerek m adet s_0, s_1, \dots dizilerinden oluşan örnek toplanmıştır.

3.5. Eylem Seçimi – Faydalanma – Araştırma

Daha fazla ödül alabilmesi için bir bilişsel elemanın, geçmişte denediği ve ödül kazandıran eylemleri seçmesi normaldir. Fakat ya bilmediği ve onlardan daha çok ödül kazandıracak eylemler varsa, bu tür eylemleri bulmak için bilişsel elemanın eylem uzayında daha önce seçmediği aksiyonlardan birisini seçmesine araştırma adı verilmektedir. Bilişsel eleman geçmişle ilgili bilgilerinden faydalanmalıdır, fakat daha iyilerini bulmak için araştırmadan asla vazgeçmemelidir. Bilişsel elemanın eylem seçme yöntemi araştırmayı engelleyecek şekilde olmamalıdır.

3.5.1. İki Seviyenin Çıktılarının Entegre Edilmesi

Rasgele seçim kullanarak, her adımda, P_{KSI} olasılığıyla, eğer mevcut durumda uygun bir eylem gösteren en az bir KSI kuralı varsa, o kural setinden bir çıkış kullanılır (Kural faydasına dayalı yarışma yoluyla), aksi takdirde alt seviyeden bir çıkış kullanılır (Her zaman hazır).

Kural Faydası

Kuralın faydası *maliyet ve kazanç* cinsinden verimliliği veya faydalılığı ölçer. Fayda, fayda değerlerini anında hesaplayan fonksiyonlarla belirlenebilir (Anderson 1993, Luce 2000). Normalde kazanç

$$Kazanç_j = \frac{c_7 + PM(j)}{c_8 + PM(j) + NM(j)} \quad 3.34$$

burada $c_7 = 1$ ve $c_8 = 2$ şeklinde seçilebilir ve maliyet

$$Maliyet_j = \frac{j - kuralı - uygulama - süresi}{kurallar - ortalama - uygulama - süresi} \quad 3.35$$

şeklinde, burada *j-kuralı-uygulama-süresi* ve *kurallar-ortalama-uygulama-süresi* tahmin edilip oluşturulmalıdır. Genel hesap şu şekildedir:

$$U_j^T = kazanç_j - v * maliyet_j \quad 3.36$$

anlamı ise bir kuralın kazancı olumlu çakışma oranına dayanılarak hesaplanır, kural tarafından oluşturulabilecek tüm çakışmalar bağlamında kural kaç adet olumlu çakışma üretir. Maliyet, uygulama süresini göz önünde bulundurur. Kesin doğru olmak zorunda değildir. Kuraldan kurala değişen (içerdiği işlem tiplerini temel alan) bir değer içerebilir ve bu kabaca kuralın uygulama zamanına karşılık gelir.

Üst seviyede kuralı faydalılığa dayanarak rasgele olarak seçeriz.

$$p(i | x) = \frac{e^{u_i^T / \tau}}{\sum e^{u_j^T / \tau}} \quad 3.37$$

bu yöntem Luce seçim aksiyonu (Luce 1959) olarak adlandırılır.

Alt seviyeden çıkış kullandığımız zaman, bir eylemi seçmek için Q değerlerinin Boltzmann dağılımına dayalı, rasgeleliği barındıran bir süreç kullanırız. Araç kullanma söz konusu olduğunda, araştırma adına kötü bir eylem

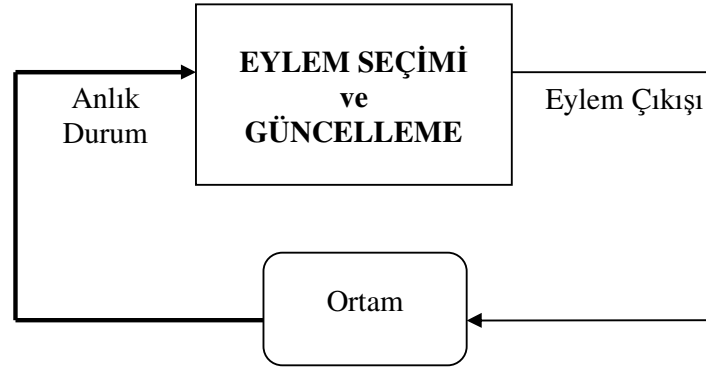
seçmek oldukça tehlikeli sonuçlar doğurabilecektir. Kötü eylem seçimlerinin kötü sonuçlar doğuracağı ortamlarda, eylem seçme olasılıklarının kestirilen değer fonksiyonuyla basamaklandırılması bir çözüm yöntemidir. Tecrübeyle sabit olan eylem halen seçim olasılığı en yüksek eylemdir, fakat diğerleri değer kestirimlerine göre ağırlıklandırılıp derecelendirilmiştir. Bunlara genel olarak Softmax eylem seçme yöntemleri adı verilmektedir. En yaygın Softmax yöntemleri Gibbs veya Boltzmann dağılımıdır. t . Adımda şu olasılıkla bir a eylemi seçer:

$$p(a | x) = \frac{e^{Q(x,a)/\tau}}{\sum_i e^{Q(x,a)/\tau}} \quad 3.38$$

burada x anlık durum, a eylem, τ ise sürecin sıcaklığını kontrol eden değişkendir. Yüksek sıcaklıklar bütün eylemlerin eşit olasılıklı olmasına yol açarken, düşük sıcaklıklar değer kestirimleri arasında fark olan eylemlerin seçim olasılıkları arasında büyük farklılıklar oluşturmaktadır. $\tau \rightarrow 0$ sınırında, seçim en bilindik eyleme doğru olacaktır.

3.6. Model Kurulumu

Bilişsel mimari genel olarak, herhangi bir ortama bırakılan bir bilişsel elemanın, zaman içinde ortamla etkileşerek ve ortamdaki geri beslemeler doğrultusunda kendi davranış biçimini düzenleyerek, en uygun davranış biçimini oluşturabilmesini amaçlar. Yani bilişsel eleman en uygun davranış biçimini oluşturabildiği zaman, ortam içerisinde karşılaştığı herhangi bir durum karşısında en doğru kararı verebilecektir.



Şekil 3.20. Bilişsel Eleman Eylem Seçimi ve güncelleme

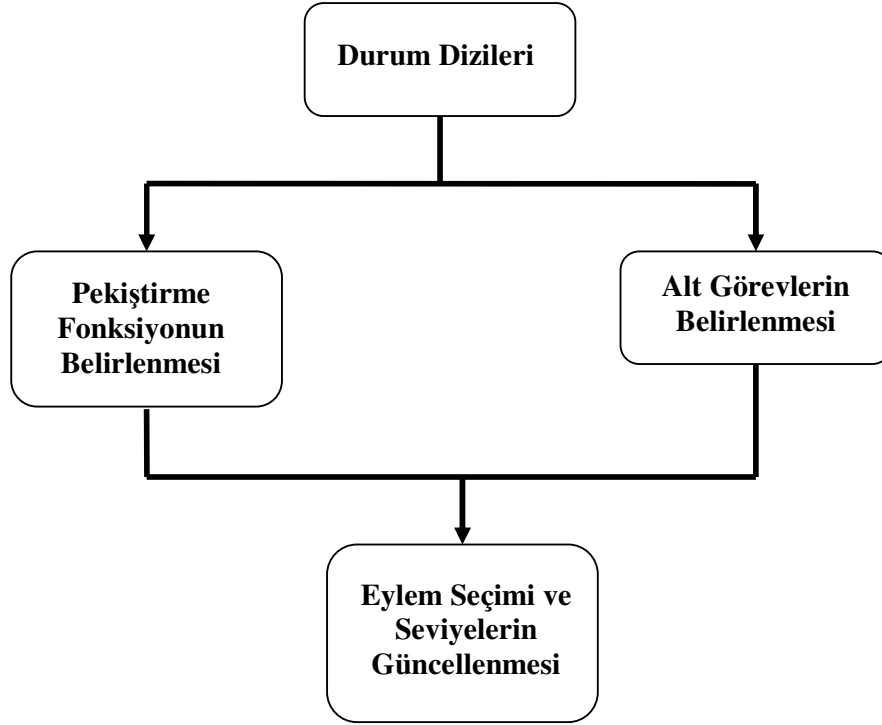
Önerdiğimiz mimaride, öğrenme ve eylem seçme işlemi, birbirine paralel hem alt seviyede hem de üst seviyede gerçekleştirilmektedir. Eğer bilişsel eleman üst seviyesine sabit kurallar yerleştirilmemişse, başlangıç aşamasında üst seviyede hiçbir kural yer almaz, diğer bir deyişle üst seviye tamamen boştur. Alt seviyede eylem seçiminin çekirdeğinde geri yayılım ağları yer almaktadır. Alt seviye ağ düğümleri algılayıcı girişlerini ifade etmekle birlikte, ağ ağırlıkları da henüz bir anlam taşımamaktadır. Üst seviyede ise bu seçim bir fayda ölçütüne göre gerçekleştirilmektedir. Bu yöntemde bilişsel eleman hiç bilmediği bir ortamda bulursa bile, bir eylem seçebilmekte, eylem sonucu yeni bir duruma geçmekte, ve geçtiği bu durumun karşılığında eylem sonucu olarak ortamdaki bir pekiştirme işareti almaktadır. Bu pekiştirme işareti doğrultusunda önce alt seviyedeki geri yayılım ağlarının ağırlıkları daha sonra üst seviyedeki çakışma oranları tekrar güncellenmekte ve yeni eylem seçimi aşamasına gelindiğinde bilişsel eleman bir önceki duruma göre biraz daha tecrübeli hale gelmektedir. Bu yöntemde ele alınan problemler bir bütün halinde düşünülmekte

gözlemleyerek eğitimin ilk aşamasını gerçekleştirecektir, burada kullanılan öğrenme yöntemi, göstererek öğrenme yöntemidir ve bölüm 3.2.6.3'te değinilen Öğreticili Q- Öğrenme yaklaşımı bu aşamaya temel oluşturmaktadır. Burada başarı ölçütü, öğretici tarafından verilen kararlar ile eğitilen bilişsel elemanın verdiği kararlar arasındaki oran tarafından belirlenmektedir. Başarı ölçütünü ise sürücü adayının sürücü olabilmek için tabi tutulduğu sınavdan aldığı puanla benzer görülebilir. Bu oranın belirli bir eşik seviyesinin üzerine geçmesiyle birinci aşama tamamlanacaktır.

2. İkinci aşamada, bilişsel eleman kendi kararlarını kendisi vermektedir. Verdiği kararların sonucunda gerçekleştirdiği eylemle oluşan yeni durum ve bu durumun getirdiği pekiştirmeye kendi davranış biçimini düzenlemeye devam edecektir. Zaman içinde bilişsel eleman deneyimler yoluyla kendini geliştirdikçe daha doğru kararlar üretecektir.

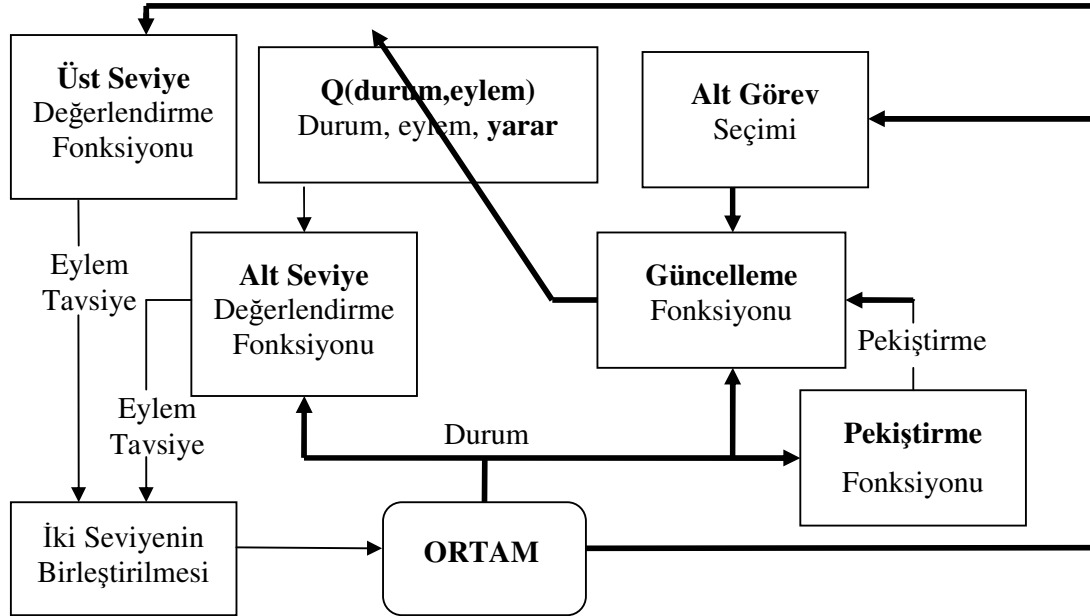
3.6.1. Öğreticili Q-Öğrenme

Bu tez çalışmasında literatürde varolan bilişsel mimari yöntemlerine göre iki farklı yaklaşım yer almaktadır. Birinci nokta problemin alt problemlere bölünerek durum uzayının küçültülmesi ve alt seviyedeki pekiştirmeli öğrenmenin yakınsamasını sağlayacak alt görev bölümlenmesinin gerçekleştirilmesi, ikinci nokta ise önceden belirlenmesi zor olan ve kişisel farklılıkları da bünyesinde barındıran pekiştirme fonksiyonunun belirlenmesidir. Durum dizileri, bilişsel elemana sağlanarak ilk aşamada pekiştirme fonksiyonu ve alt görev bölümlenmesinin gerçekleştirilmesi sağlanır, ikinci aşamada ise bu alt görev bölümlenmeleri ve pekiştirme fonksiyonu kullanılarak geri yayılım ağlarının eğitimi ve daha sonra üst seviyenin eğitimi gerçekleştirilir. Bu durumda önerdiğimiz mimarinin genel yapısı şekil 3.22'deki gibi olacaktır.



Şekil 3.22. Önerilen mimarinin blok şeması

Eylem seçimi ve Seviyelerin güncellenmesiyle ilgili oluşturulan yeni yapı ise şekil 3.23'deki gibi olacaktır.



Şekil 3.23. Önerilen mimari eylem seçimi ve güncelleme

Şekil 3.22'deki yapıdan da görülebileceği gibi, pekiştirme fonksiyonu ve alt görevler otomatik olarak belirlenerek güncellenen yapıyla birime sağlanmaktadır. Akış diyagramı ise şu şekilde ifade edilebilir:

1. Eğitici tarafından sağlanan yeni durum dizisi alınır.

Durum dizisi bir eğitici tarafından sağlanan ve bilişsel eleman tarafından gözlenen, boyut/değer çiftleriyle ifade edilen algılayıcılardan gelen bilgileri içeren dizidir ve $(boyut_1, deęer_1), (boyut_2, deęer_2), \dots, (boyut_n, deęer_n)$ yapısında ifade edilir. Her boyut ayrıktır ve deęeri sabit bir deęer kümesinin elemanıdır.

2. Dizi başlatılır.

3. Pekiştirme fonksiyonu eğitilir.

Pekiştirme fonksiyonunu eğitimiyle ilgili prosedüre bölüm 3.4.4'te ayrıntılı bir şekilde deęinilmiştir. İnsan sürücü gözlenerek s_0, s_1, \dots dizilerinden oluşan m adet örnek yöntem için toplanmıştır. Burada genel akış çizelge 7'deki şekilde olacaktır.

- a. Rasgele bir $\pi^{(0)}$ davranış biçimi seç ve çalıştır, $\mu^{(0)} = \mu(\pi^{(0)})$ 'ı hesapla ve $i=1$ yap.
- b. $t^{(i)} = \max_{w: \|w\|_2 \leq 1} \min_{j \in \{0, \dots, (i-1)\}} w^T (\mu_E - \mu^{(j)})$ denklemini hesapla burada $w^{(i)}$ eşitliği maksimum yapan w deęeri olsun.
- c. Eğer $t^{(i)} < \epsilon$, sonlandır
- d. RL algoritmasını kullanarak, $R = (w^{(i)})^T \phi$ ödülünü kullanarak $\pi^{(i)}$ en uygun davranış biçimini hesapla.
- e. $\mu^{(i)} = \mu(\pi^{(i)})$ 'yi hesapla.
- f. $i = i + 1$ yap ve 2. adıma geri dön.

Burada

ϕ daha öncede deęindiğimiz özellik vektörü,

$$\mu_E = \frac{1}{m} \sum_{i=1}^m \sum_{t=0}^{\infty} \gamma^t \phi(s_t^{(i)})$$

μ özellik beklentisi vektörü

w ağırlık vektörü

4. Alt görevler belirlenir.

Alt görevlerin belirlenmesi ile ilgili prosedür bölüm 3.4.3'te ayrıntılı bir şekilde açıklanmıştır.

5. Dizi tekrar başlatılır.

6. Eylem seçme prosedürü çalıştırılır:

a. Anlık koşul s gözlemlenir.

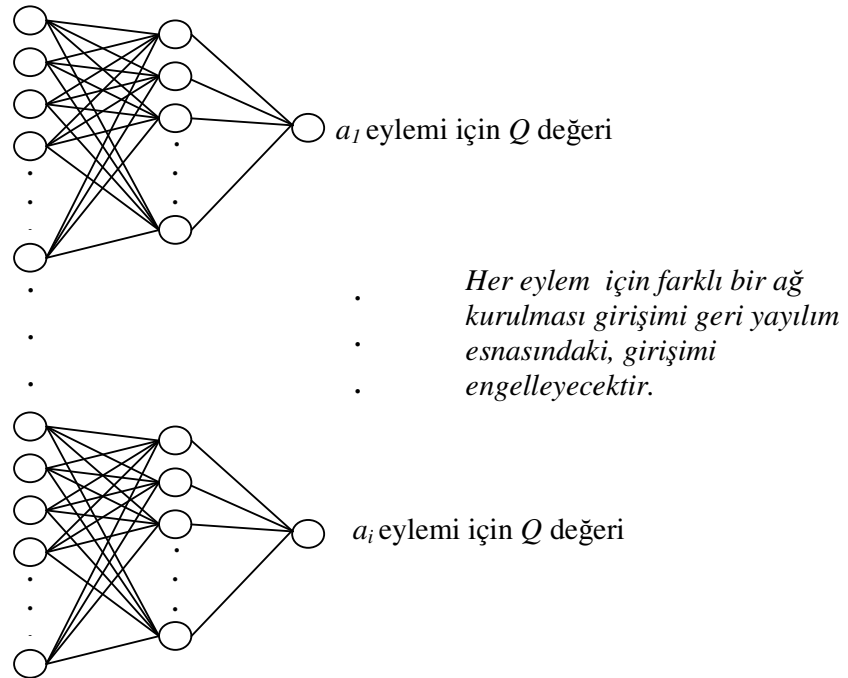
Alt seviyede gözlemlenen her boyut/değer çifti giriş düğümlerine transfer edilir. Üst seviyede gözlenen boyut/değer çiftleri ise sırasıyla kural koşullarıyla karşılaştırılır.

- Algılayıcı girişleri. Algılayıcı girişleri, her birinin belirli aralıklarda değere sahip olabildiği, belirli sayıda boyuta bölünmüştür. Her adımda, o anki durumun bir parçası olarak, $(boyut_1,değer_1).(boyut_2,değer_2)...(boyut_n,değer_n)$ yapısındadır. Her boyutun her farklı değeri, ayrık bir düğümlerle temsil edilir. Çalışmamızda algılayıcı girişleri ortamda bulunan diğer araçların pozisyonları, bulunduğumuz aracın hızı, yol üzerindeki konumu bilgilerini içermektedir.
- Çalışan hafıza elemanları. Çalışan hafıza elemanları, eylem karar verme algoritmasının her adımında, anlık durumun bir parçası olarak temsil edilir. Çalışan hafıza, belirli bir limite kadar belirli sayıda elemanlar içerir. Her elemanda bir boyut/değer çifti seti vardır. Çalışan hafıza elemanlarının temel görevlerinden biri, Markov özelliğine sahip olmayan ortam için geçmiş bilgisinin de değerlendirmeye alınabilmesinin sağlamasıdır.
- Hedef yapısından bir eleman. Her hedef bir boyut/değer kümesiyle temsil edilir. Her hedefte, her birinin olası hedef kodlarının ayrık bir düğümlerle temsil edildiği bir hedef boyutu vardır. Bu hedef boyutu doğrultusunda, her hedefin belirli sayıda değişken boyutu vardır. Her

değişken boyutunun her bir değeri ayrı bir düğümle temsil edilir. Bir t anı için yalnızca bir hedef aktif olabilir. Her adımda hedef anlık durumun bir parçasıdır. Hedef yapısının elemanları, otomatik hedef bulma yöntemi tarafından oluşturulmaktadır.

- b.** Alt seviyede koşulla ilişkili $s : Q(s, a_1), Q(s, a_2), \dots, Q(s, a_n)$ tüm olası a_i eylemlerinin değerlerini hesaplanır.

Durum bilgisi şekil 3.23'deki gibi geri yayılım ağlarına uygulanarak her eylem için bir çıkış değeri oluşturulur.



Şekil 3.24. Geri Yayılım ağlarıyla eylem seçme

Her eylem için ayrıntılı bir ağ kullanılmasının gerekçelerine bölüm 3.2.6.2'de detaylı bir şekilde değinilmişti.

- c.** Üst seviyedeki eylem kurallarına dayalı olarak, tüm olası eylemleri (b_1, b_2, \dots, b_m) bulunur.

Üst seviyedeki eylem değerlerinin belirlenmesi için anlık duruma uyan kurallara bakılmaktadır. Kuralın koşulu anlık duruma karşılık gelmektedir.

Bu koşul için denklem 3.34'ten

$$Kazanç_j = \frac{c_7 + PM(j)}{c_8 + PM(j) + NM(j)}$$

ve denklem 3.35'ten

$$Maliyet_j = \frac{j - kuralı - uygulama - süresi}{kurallar - ortalama - uygulama - süresi}$$

ve denklem 3.36'dan fayda ölçümü belirlenir.

$$U_j^T = kazanç_j - v * maliyet_j$$

- d. a_i (alt seviyeden) ve b_j (üst seviyeden) alt ve üst seviye eylemlerine bir birleştirme algoritması uygulayarak, **en uygun eylem seçilir.**

Denklem 3.37'deki Luce'nin seçim aksiyonuyla üst seviyeden bir eylem seçilir.

$$p(i | x) = \frac{e^{u_i^T / \tau}}{\sum e^{u_i^T / \tau}}$$

Alt seviye eylem seçimi denklem 3.37'deki Softmax eylem seçimi yöntemiyle seçilir.

$$p(a | x) = \frac{e^{Q(x,a) / \tau}}{\sum_i e^{Q(x,a) / \tau}}$$

İki seviyeden bir çıkış alınması ise şu şekilde gerçekleştirilir: Eğer üst seviyede hazır bir kural varsa çıkış oradan alınır eğer yoksa alt seviyeden seçilir.

7. Güncelleme Prosedürü çalıştırılır.

- Seçilen a eylemini ve uzman tarafından seçilen a eylemine göre başarı ölçütü güncellenir.
- Uzman tarafından seçilen a eylemini bir sonraki koşul s' (olası) ile r pekiştirmesi gözlemlenir.

- c. Q -öğrenme geri yayılım algoritmasına göre geri besleme bilgisine dayanarak **alt seviye güncellenir.**

Q -öğrenme-geri yayılım algoritması güncelleme prosedürü şu şekilde oluşturulur.

Çizelge 3.4'teki güncelleme prosedürüne göz atarsak

1. $s_t \leftarrow$ *anlık durum*; her a eylemi için, $Q_a \leftarrow Q(s_t, a)$

Anlık durumu s_0 olarak seçer ve prosedürü başlatırız. Her eylem için Q -değerlerinin başlangıç durumları ayarlanır.

2. $a \leftarrow$ *seç*(U, T);

Bu aşamada eylem seçimi, örneklerin alındığı insan sürücünün davranış dizilerindeki geçerli eylem olarak seçilecektir.

3. a eylemini çalıştır; $(y, r) \leftarrow$ yeni durum ve pekiştirme

a eylemi çalıştırıldıktan sonra geçilen örnek dizilerden elde edilen geçerli yeni duruma bakılır.

4. $Q' \leftarrow r + \gamma \cdot \text{Max}\{Q(y, k \mid k \in \text{aksiyonlar})\}$

Yeni Q değeri hesaplanır. Bunun için gereken r değeri pekiştirmeli öğrenme fonksiyonundan elde edilir. $\gamma = 0.9$ olarak seçilebilir.

5. Fayda ağını geri yayılım hatası ΔU ile s girişi doğrultusunda düzelt,

$$\Delta Q_a = \begin{cases} Q' - Q_a & a_i = a \\ 0 & \text{veya} \end{cases}$$

Hangi eylemin ağı güncelleniyorsa o ağ için geri yayılım hatası yukarıdaki ifade olarak alınır ve ilgili ağ geri yönde (hata düzeltme ve ağırlıkları ayarlama yönünde) çalıştırılarak ağırlıkları düzenlenir

6. 1. adıma geri dön

- d. Kural-Seçim-İyileştir algoritmasına göre **üst seviye güncellenir.**

Kural-seçim-iyileştir algoritmasına bölüm 3.1.3.2'de ayrıntılı bir şekilde değinilmiştir.

8. Eğer dizi sonlanmadıysa 6. adıma geçilir, dizi sonlandıysa 1. adıma geçilir.
9. Başarı oranı güncellenir ve birinci adıma geçilir.

3.6.2. Kendi Kendine Öğrenme

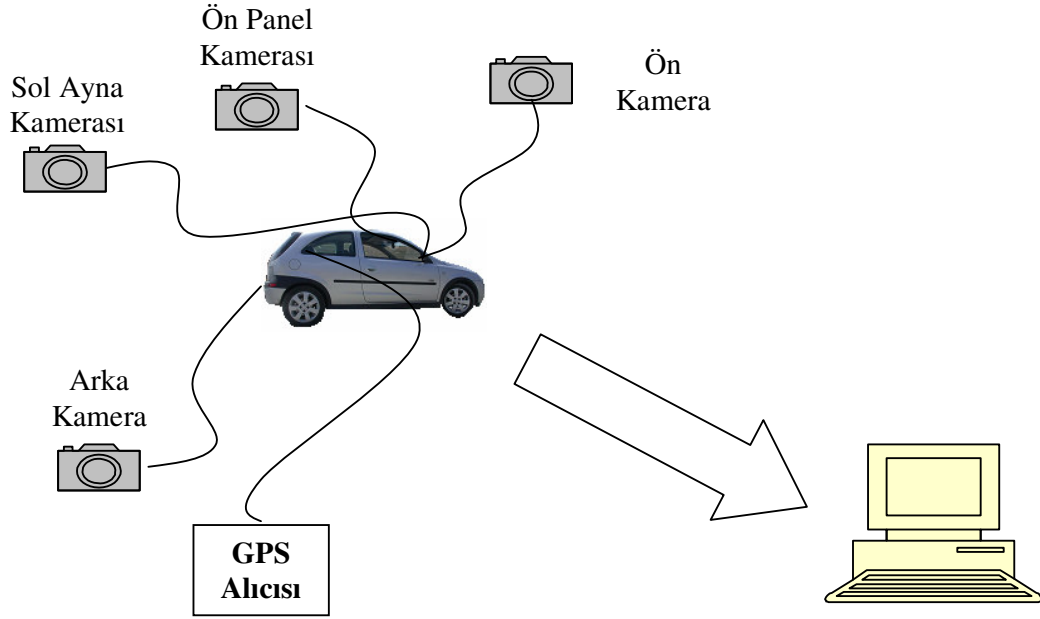
Kendi kendine öğrenme aşamasında, bilişsel eleman büyük oranda doğru kararlar verebilmekle birlikte, eğitimine devam etmektedir, bununla birlikte artık eğitimine insan sürücüden gelen davranış dizileri üzerine değil kendi verdiği kararlar doğrultusunda aldığı pekiştirmelerle devam edecektir.

1. Anlık koşul s 'i gözlemlenir.
2. Alt seviyede koşulla ilişkili $s : Q(s, a_1), Q(s, a_2), \dots, Q(s, a_n)$ tüm olası a_i eylemlerinin değerlerini hesaplanır.
3. Üst seviyedeki eylem kurallarına dayalı olarak, tüm olası eylemler (b_1, b_2, \dots, b_m) bulunur.
4. a_i (alt seviyeden) ve b_j (üst seviyeden) alt ve üst seviye eylemlerine bir birleştirme algoritması uygulayarak, en uygun eylem seçilir.
5. Seçilen a eylemini uygula ve bir sonraki koşul s' (olası) ile r pekiştirmesini gözlemlenir.
6. Q -öğrenme geri yayılım algoritmasına göre geri besleme bilgisine dayanarak alt seviye güncellenir.
7. Kural-Seçim-İyileştir algoritmasına göre üst seviye güncellenir.
8. Birinci adıma geri dönülür.

3.7. Deneysel Çalışma ve Sonuçlar

3.7.1. Deneysel Çalışma

Deneysel çalışma, normal yol şartlarında gerçekleştirilmiştir. Deneysel çalışmada, araca bağlı çeşitli algılayıcılar vasıtasıyla anlık durum ve bu duruma karşı gözlenen sürücünün yaptığı eylem ortaya çıkarılmıştır. Araca bağlı algılayıcılar, şekil 3.25'te görülebilir.



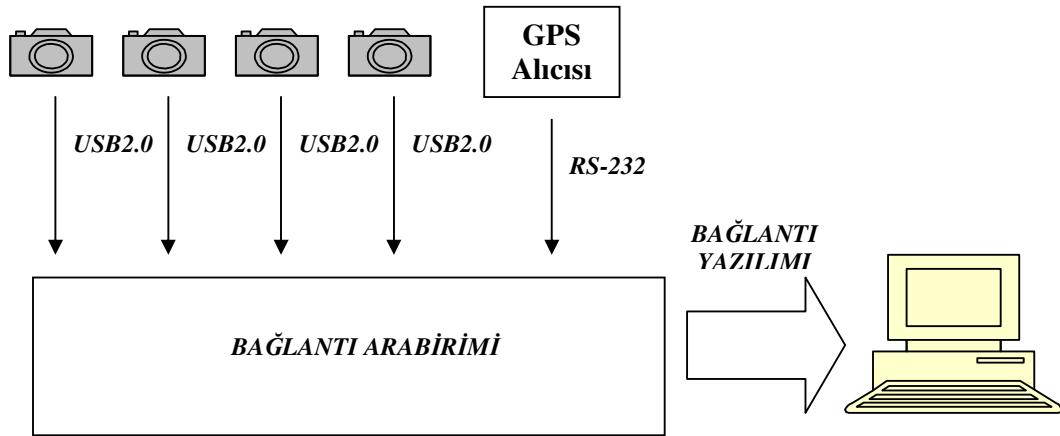
Şekil 3.25. Sistem genel yapısı

Araca bağlı dört adet kamera mevcuttur ve kameralardan gelen bilgileri senkronize etmek ve hız bilgisini alabilmek için bir GPS alıcısı kullanılmıştır. Algılayıcılardan gelen bilgiler birer saniye aralıkla bilgisayara aktarılır ve bilgisayarda değerlendirilerek, ortamın anlık algısal durumu ortaya çıkarılır. Aynı şekilde GPS alıcısı ve ön panel kamerasından elde edilen hız bilgisinden sağlanan hız farkları ve ön kameralarda elde edilen görüntülerden yola çıkılarak yola göre oluşan açı farkları belirlenerek, aracın dikey ve yatay ekseninde gerçekleştirdiği eylemler ortaya çıkarılmıştır. Kameralardan gelen bilgilerin nasıl değerlendirildiği bölüm 3.7.2'de açıklanmaktadır.

Kullanılan kameralar, 640x480 piksel çözünürlüklü, USB arabirimine sahip standart kamera tipindedir. GPS alıcısı ise 5 m pozisyon ve 3km/saat hız bilgisi hassasiyetli *SirfStarIII* yonga setine sahip, RS-232 arabirimi aracılığıyla bilgisayarla haberleşebilen bir cihazdır.

Kameralar Şekil 3.26'da görülebildiği gibi, çektiği görüntüleri *jpeg* formatında, USB portu üzerinden bilgisayara göndermektedir. GPS alıcısı ise bilgisayarın seri portu aracılığıyla elde ettiği konum bilgilerini, *ASCII-text* formatında bilgisayara iletmektedir.

Hem kameralar hem de GPS alıcısı her saniye için örnekleme gerçekleştirmektedir. Bu örneklemler bilgisayara gönderilirken, dosya isimleri örnekleminin gerçekleştirildiği zaman damgası olarak kaydedilmekte ve bu doğrultuda yazılım tarafından eşzamanlı yani aynı isme sahip dosyalar değerlendirmeye alınmaktadır.



Şekil 3.26. Algılayıcılardan gelen bilgilerin yazılıma aktarılması.

Elde edilen eşzamanlı bilgiler anlık algısal durumu oluşturmakta ve anlık algısal durum girdi olarak sisteme sunulmaktadır. Benzer şekilde bu anlık algısal duruma karşı oluşan anlık algısal çıkış da ortaya çıkarılmakta ve bu algısal çıkış sistemin çıkışı olarak öğrenme sürecinde kullanılmak üzere geri yayılım ağlarına sunulmaktadır.

3.7.2. Algısal Girişler

3.7.2.1. Algılayıcılar

Yapılan çalışmada, araca bağlı dört farklı kameradan elde ettiğimiz bilgilerle, aracın içinde bulunduğu durum bilgisi elde edilmektedir. Şekil 3.25'te araca bağlı ön kamera görüntüsü, şekil 3.26'da sol ayna kamera görüntüsü, şekil 3.27'de arka kamera görüntüsü ve şekil 3.28'de araç hız ve yön bilgisinin çıkarıldığı ön konsol görüntüsü yer almaktadır. Durum bilgisi, çevre bilgileri ve aracın kendi bilgileri olmak üzere temel olarak ikiye ayrılmaktadır. Çevre bilgilerinden kasıt aslında trafik bilgileridir. Yolun coğrafi şekli de sistem açısından önemli bir bilgi olarak temsil edilse de, veri topladığımız yol neredeyse düz sayılabileceği için, bu bilgi örneklendirme açısından ihmal edilmiştir. Kendisi ile ilgili bilgilerse aracın bu ortam içindeki hızı ve konumu ile ilgili bilgilerden oluşturmaktadır. Bu bilgiler şu şekilde ifade edilebilir:

Çevre Şartları(Trafik):

- Arkamızdaki aracın bize göre uzaklığı (Şekil 3.29'dan)
- Önümüzdeki aracın bize göre uzaklığı (Şekil 3.27'den)
- Sol şeritte ve arkamızda kalan aracın bize göre uzaklığı (Şekil 3.28'den)
- Sol şeritte ve önümüzde kalan aracın bize göre uzaklığı (Şekil 3.29'dan)
- Sağ şeritte ve arkamızda kalan aracın bize göre uzaklığı (Şekil 3.29'den)
- Sağ şeritte ve önümüzde kalan aracın bize göre uzaklığı (Şekil 3.27'den)

Araç Şartları

- Aracımızın hız bilgisi (Şekil 3.30'den)
- Aracımızın hangi şeritte olduğu (Şekil 3.27'den)
- Aracımızın yönü (Şekil 3.27'den)



Şekil 3.27. Ön Kamera



Şekil 3.28. Sol Ayna Kamerası



Şekil 3.29. Arka Kamera



Şekil 3.30. Ön Panel Kamerası

Bir bütün olarak düşünüldüğü zaman bu görüntülerden elde edilen bilgiler, sürücü için durum ifadesini oluşturmaktadır. Her durum karşısında sürücü bir eylem sergileyecektir. Temel olarak sergilenen eylemler de ikiye ayrılmaktadır. Dikey eylemler ve yatay eylemler. Dikey eylemler adından da anlaşılacağı şekilde aracın dikey doğrultudaki hareketlerini yani hızıyla ilgili düzenlemeleri içermektedir, yatay eylemler ise aracın yatay yöndeki hareketlerini yani direksiyon aracılığıyla gerçekleştirilen hareketlerini düzenlemektedir.

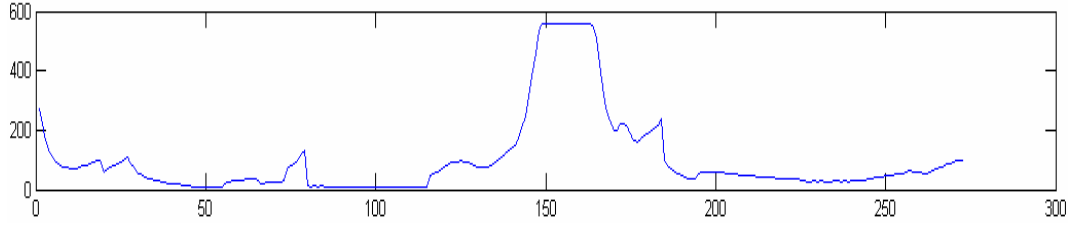
Dikey Eylem Uygulayıcıları:

- Gaz Pedalı (Şekil 3.30'dan)
- Fren Pedalı (Şekil 3.30'dan)

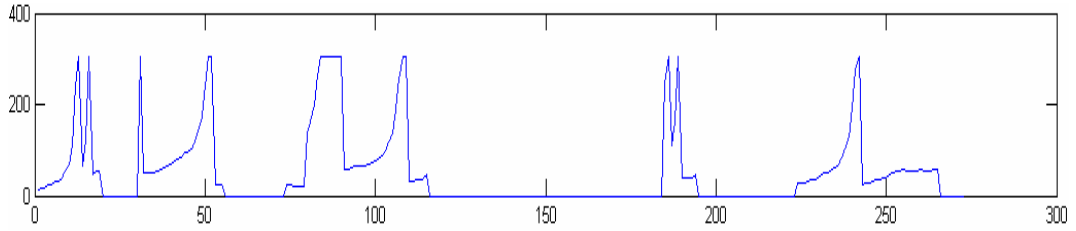
Yatay Eylem Uygulayıcıları

- Direksiyon (Şekil 3.30'dan)

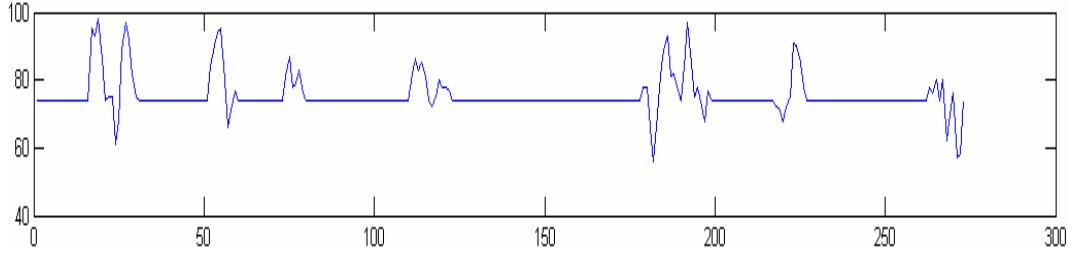
Deneyisel çalışmada bu kameralardan elde edilen 1 saniye zaman aralıklı bilgilerden oluşturulan 273 adet örnekli durum-eylem dizileri kullanılmıştır. Şekil 3.31'den 3.34'e kadar kameralardan toplanan görüntülerden elde edilen, bize göre arkada yer alan aracın bize uzaklığını, bize göre sol ve arkada kalan aracın bize uzaklığını, direksiyonun dönme durumunu ve aracımızın hızını gösteren grafikler yer almaktadır.



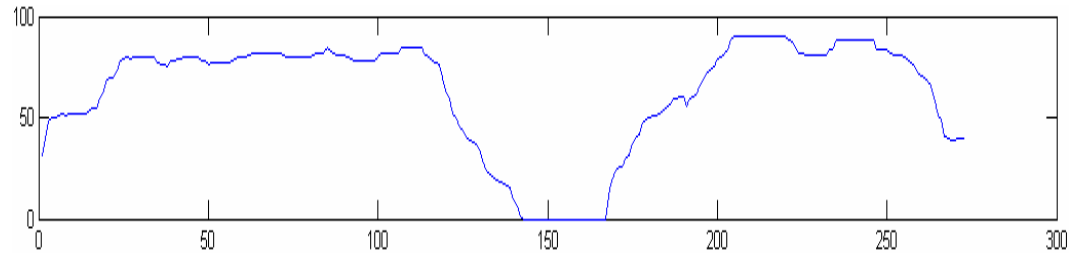
Şekil 3.31. Tam Arkamızdaki Araç ile aramızdaki mesafe(pixel olarak&ters orantılı)



Şekil 3.32. Sol Arkamızdaki araç ile aramızdaki mesafe(pixel olarak&ters orantılı)

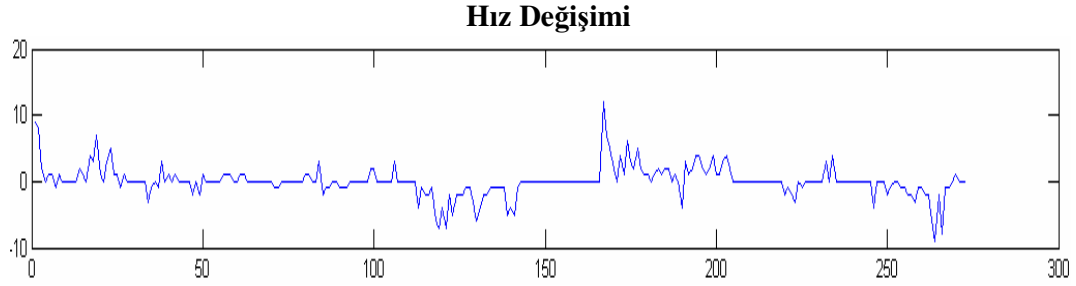


Şekil 3.33. Direksiyon Pozisyonu. Düz konum 74. piksel ve oynamalar piksel olarak belirtilmiştir(Doğru Orantılı)

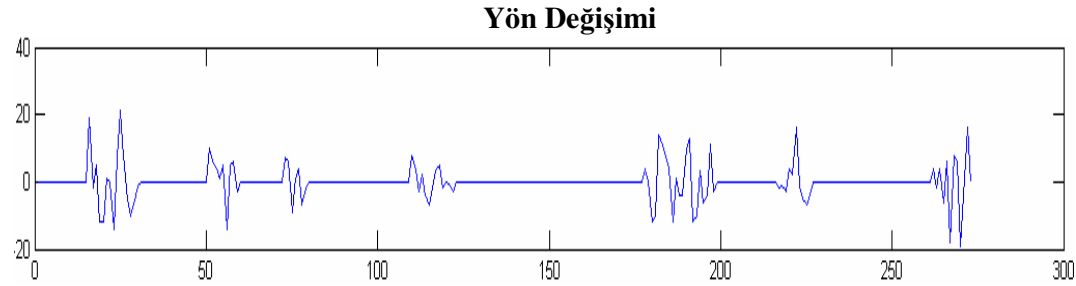


Şekil 3.34. Aracımızın hızı.(km/s olarak)

Şekil 3.35 ve 3.36'daki hız değişimi ve yön değişimi, sürücünün gerçekleştirdiği eylemler sonucu oluşan değişikliklerdir. Bu değişikliklerden sürücünün o anda uyguladığı eylem çıkartılabilir. Sürücü eylemini gerçekleştirdikten sonra ortam yeni durumuna geçiş yapar.



Şekil 3.35. Her saniyedeki hız değişimleri (km/s olarak)



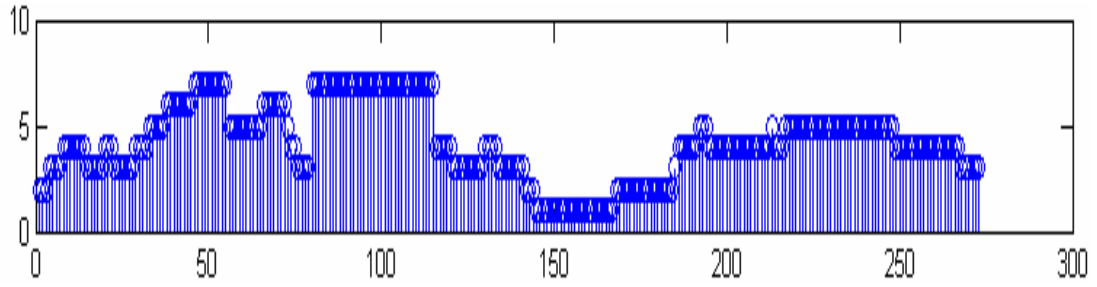
Şekil 3.36. Her saniyedeki direksiyonun yönündeki değişim(piksel olarak)

3.7.2.2. Ayrıklaştırma

Bu verilerin sisteme giriş olarak ifade edilebilecek durum-eylem dizilerine dönüştürülebilmesi için bir ayrıklaştırma işlemine ihtiyaç vardır. Ayrıklaştırma şu şekilde gerçekleştirilmiştir:

Durumlar:

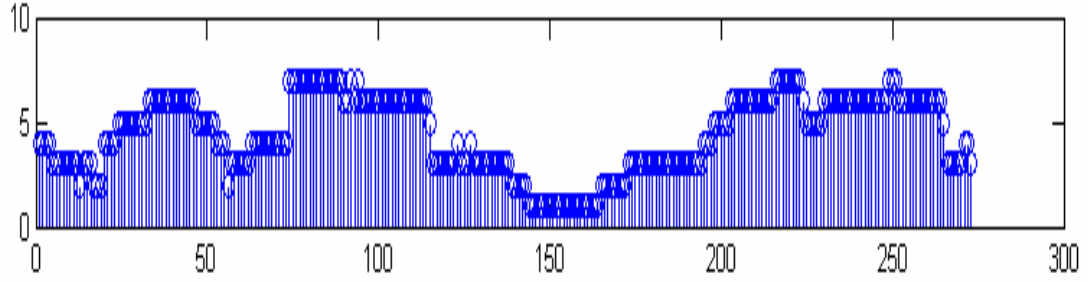
- Şekil 3.37'de yer alan arkamızdaki aracın bize göre uzaklığı – 8 kademe; en yakından en uzağa doğru – bilgisi



Şekil 3.37. Arkamızdaki aracın bize göre ayrıklaştırılmış uzaklığı.

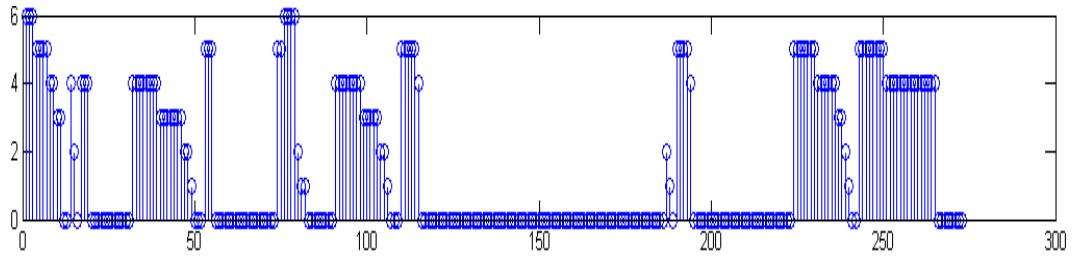
0 – Çok yakın, ..., 8 – çok uzak

- Şekil 3.38'de yer alan önümüzdeki aracın bize göre uzaklığı – 8 kademe; en yakından en uzağa doğru – bilgisi



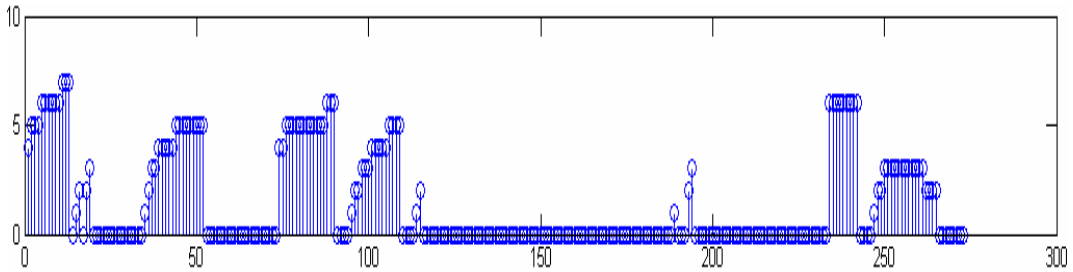
Şekil 3.38. Önümüzdeki aracın bize göre ayrılaştırılmış uzaklığı. 0 – Çok yakın, ..., 8 – çok uzak

- Şekil 3.39'da yer alan bize göre sol şeritte ve arkada kalan aracın bize göre uzaklığı – 8 kademe; en yakından en uzağa doğru – bilgisi



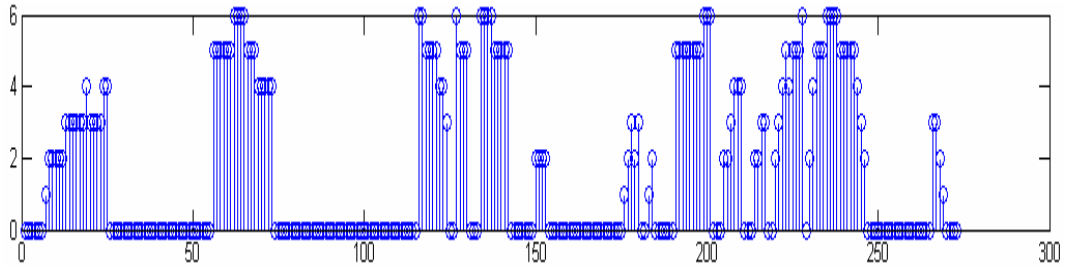
Şekil 3.39. Sol şeritte ve arkamızdaki aracın bize göre ayrılaştırılmış uzaklığı. 0 – Çok yakın, ..., 8 – çok uzak

- Şekil 3.40'ta yer alan bize göre sol şeritte ve önde kalan aracın bize göre uzaklığı – 8 kademe; en yakından en uzağa doğru – bilgisi



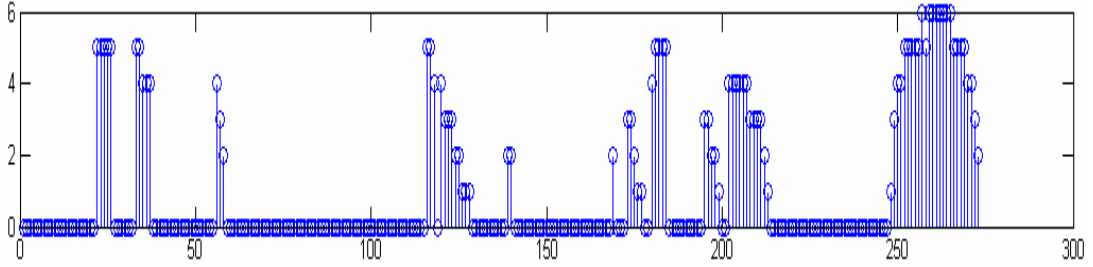
Şekil 3.40. Sol şeritte ve önümüzdeki aracın bize göre ayrılaştırılmış uzaklığı. 0 – Çok yakın, ..., 8 – çok uzak

- Şekil 3.41'de yer alan bize göre sağ şeritte ve arkada kalan aracın bize göre uzaklığı – 8 kademe; en yakından en uzağa doğru – bilgisi



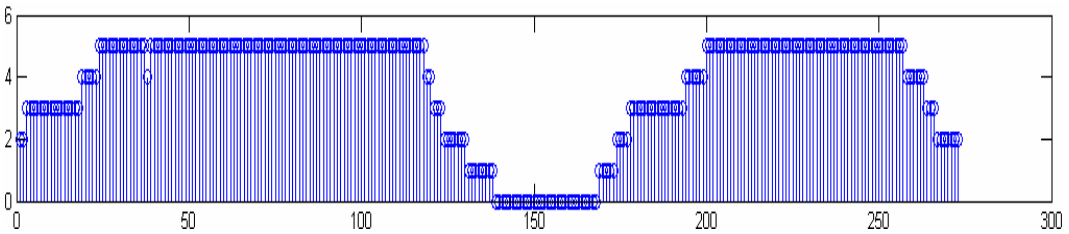
Şekil 3.41. Sağ şeritte ve arkamızdaki aracın bize göre ayrıklaştırılmış uzaklığı. 0 – Çok yakın, ..., 8 – çok uzak

- Şekil 3.42'de bize göre sağ şeritte ve önümüzde kalan aracın bize göre uzaklığı – 8 kademe; en yakından en uzağa doğru – bilgisi



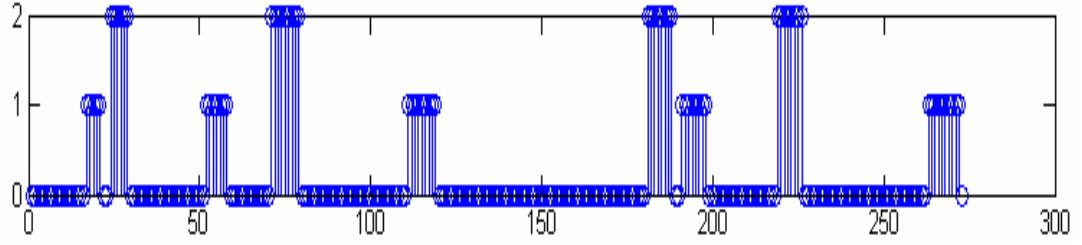
Şekil 3.42. Sağ şeritte ve önümüzdeki aracın bize göre ayrıklaştırılmış uzaklığı. 0 – Çok yakın, ..., 8 – çok uzak

- Şekil 3.43'te yer alan aracımızın hız – 8 kademe en yavaştan en hızlıya doğru – bilgisi



Şekil 3.43. Ayrıklaştırılmış hız bilgisi. 0 – Çok yavaş, ..., 8 – çok hızlı

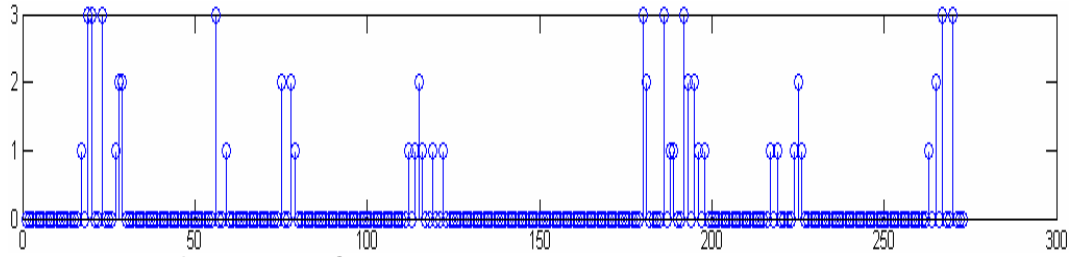
- Şekil 3.44'te yer alan aracımızın hangi şeritte – 3 kademe – olduğunun bilgisi



Şekil 3.44. İçinde bulunduğumuz şerit. 0 – En Sol, 1 – Orta, 2 – En sağ

Eylemler:

- Şekil 3.45'te yer alan gaz pedalına basma – 5 kademe; En zayıftan kuvvetliye doğru, 0 – hiç basma, 1 hafif bas, ... – bilgisi

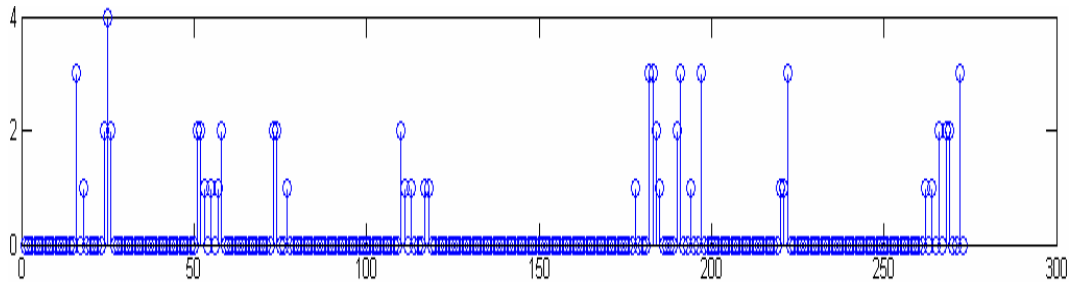


Şekil 3.45. Gaz pedalına basmanın ayrılaştırılmış şiddeti.

0 – Basma, ..., 4 – Kuvvetli Bas

- Şekil 3.46'da yer alan fren pedalına – 5 kademe; En zayıftan kuvvetliye doğru, 0 – hiç basma, 1 – hafif bas, ... – basma derecesi bilgisi

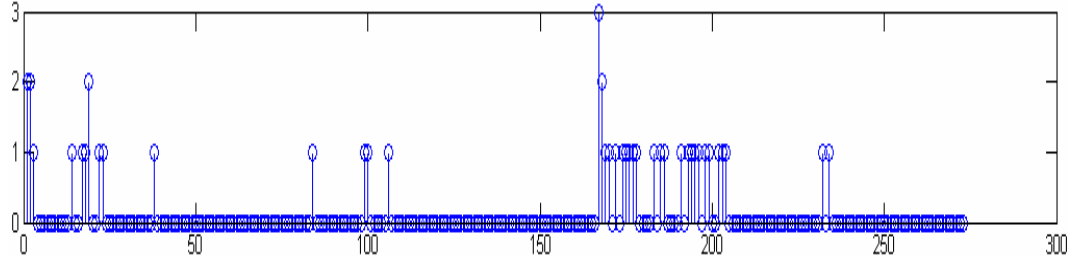
Fren Pedalı



Şekil 3.46. Fren pedalına basmanın ayrılaştırılmış şiddeti.

0 – Basma, ..., 4 – Kuvvetli Bas

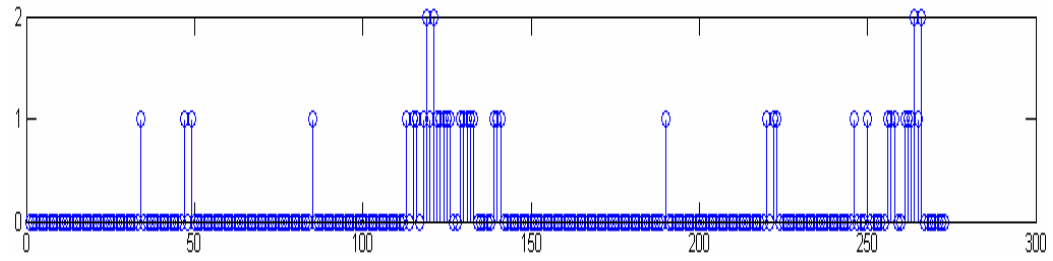
- Şekil 3.47’de yer alan direksiyonun sağa doğru dönme bilgisi



Şekil 3.47. Direksiyonu sağa doğru çevirmenin ayrılaştırılmış şiddeti.

0 – Çevirme, ..., 3 –Çok Çevir

- Şekil 3.48’de yer alan direksiyonun sola doğru dönme bilgisi



Şekil 3.48. Direksiyonu sola doğru çevirmenin ayrılaştırılmış şiddeti.

0 – Çevirme, ..., 3 –Çok Çevir

Buradaki ayrıklaştırma, kameradan gelen bilgilerin doğrusal bir şekilde bölümlenmesiyle gerçekleştirilmiştir. Örneğin, hız için her bir kademe 15 km/s’lik bölmeye karşılık gelirken. Hızı $-1 < \Delta v < 1$ (km/s) aralığında tutan gaza veya frene basmama işlemi iken $2 \leq \Delta v < 6$ (km/s) gaza basmanın 1. kademesine, $-6 < \Delta v \leq -2$ (km/s) ise frene basmanın birinci kademesine karşılık gelmektedir. Burada her bir özelliğin her bir kademesi sistemiz için bir giriş düğümünü oluştururken, her bir eylem özelliğinin her kademesi sistemimiz açısından bir çıkış düğümü anlamına gelmektedir.

3.7.3. Pekiştirme Fonksiyonunun Elde Edilmesi

Pekiştirme fonksiyonunun otomatik olarak elde edilmesi oluşturduğumuz modelin önemli aşamalarından birisidir. Pekiştirme fonksiyonunu elde etmek için istediğimiz dizileri oluşturmamızı sağlayabilecek bir sürücü simülatörü oluşturulmuştur. Sürücü simülatörüyle elde edebildiğimiz diziler, araç üzerinden topladığımız veriler doğrultusunda oluşturabildiğimiz her türlü durum kombinasyonunu içerdiği için pekiştirme fonksiyonunun elde edilebilmesi açısından yeterlidir. Bunu yanında araçtan toplanan diziler ise uzman davranış biçimini temsil edecek ve denklem 3.33'teki

$$\mu_E = \frac{1}{m} \sum_{i=1}^m \sum_{t=0}^{\infty} \gamma^t \phi(s_t^{(i)})$$

ifadesinden μ_E vektörü hesaplanacaktır. Burada bu vektörün hesaplanmasında kullanılan ϕ özellik vektörü ise şu şekilde ifade edilebilir:

$\phi = \{(Arka Araç - 8. kademe uzak), (Arka Araç - 7. kademe uzak), (Arka Araç - 6. kademe uzak), (Arka Araç - 5. kademe uzak), (Arka Araç - 4. kademe uzak), (Arka Araç - 3. kademe uzak), (Arka Araç - 2. kademe uzak), (Arka Araç - 1. kademe uzak), (Arka Araç - 0. kademe uzak), (Öndeki Araç - 8. kademe uzak), (Öndeki Araç - 7. kademe uzak), (Öndeki Araç - 6. kademe uzak), (Öndeki Araç - 5. kademe uzak), (Öndeki Araç - 4. kademe uzak), (Öndeki Araç - 3. kademe uzak), (Öndeki Araç - 2. kademe uzak), (Öndeki Araç - 1. kademe uzak), (Öndeki Araç - 0. kademe uzak), (Sağ Arka Araç - 8. kademe uzak), (Sol Arka Araç - 7. kademe uzak), (Sol Arka Araç - 6. kademe uzak), (Sol Arka Araç - 5. kademe uzak), (Sol Arka Araç - 4. kademe uzak), (Sol Arka Araç - 3. kademe uzak), (Sol Arka Araç - 2. kademe uzak), (Sol Arka Araç - 1. kademe uzak), (Sol Arka Araç - 0. kademe uzak), (Sağ Ön Araç - 8. kademe uzak), \dots, (Sol Arka Araç - 8. kademe uzak), \dots, (Sol Ön Araç - 8. kademe uzak), \dots, (Yola göre yön - 3. kademe sol), \dots, (Hız - 3. kademe yüksek), (Hız - 2. kademe yüksek), (Hız - 1. kademe yüksek), (Hız - 0. kademe yüksek), (Şerit - En sol), (Şerit - Orta), (Şerit - Sağ)\}$

Her durumda bu algılayıcı kademelerinin bir kısmı aktif olacaktır, örneğin durum 0, basitçe şu şekilde ifade edilebilirken

$s_0 = \{(Arka Araç - 5. kademe uzak), (Öndeki Araç - 3. kademe uzak), (Sağ Arka Araç - 5. kademe uzak), (Sağ Ön Araç - 3. kademe uzak), (Sol Arka Araç - 8. kademe uzak), (Sol Ön Araç - 8. kademe uzak), (Yola göre yön - 3. kademe sol), (Hız - 3. kademe yüksek), (Şerit - En sol)\}$

Bu durum için ϕ özellik vektörü şu değerde olacaktır:

$\phi(s_i) = \{(Arka Araç - 8. kademe uzak), (Arka Araç - 7. kademe uzak), (Arka Araç - 6. kademe uzak), (Arka Araç - 5. kademe uzak), (Arka Araç - 4. kademe uzak), (Arka Araç - 3. kademe uzak), (Arka Araç - 2. kademe uzak), (Arka Araç - 1. kademe uzak), (Arka Araç - 0. kademe uzak), (Öndeki Araç - 8. kademe uzak), (Öndeki Araç - 7. kademe uzak), (Öndeki Araç - 6. kademe uzak), (Öndeki Araç - 5. kademe uzak), (Öndeki Araç - 4. kademe uzak), (Öndeki Araç - 3. kademe uzak), (Öndeki Araç - 2. kademe uzak), (Öndeki Araç - 1. kademe uzak), (Öndeki Araç - 0. kademe uzak), (Sol Arka Araç - 8. kademe uzak), (Sol Arka Araç - 7. kademe uzak), (Sol Arka Araç - 6. kademe uzak), (Sol Arka Araç - 5. kademe uzak), (Sol Arka Araç - 4. kademe uzak), (Sol Arka Araç - 3. kademe uzak), (Sol Arka Araç - 2. kademe uzak), (Sol Arka Araç - 1. kademe uzak), (Sol Arka Araç - 0. kademe uzak), (Sol Ön Araç - 8. kademe uzak), \dots, (Sol Ön Araç - 8. kademe uzak), \dots, (Sol Ön Araç - 0. kademe uzak), (Sağ Arka Araç - 8. kademe uzak), \dots, (Sağ Arka Araç - 5. kademe uzak), \dots, (Sağ Arka Araç - 0. kademe uzak), (Sağ Ön Araç - 8. kademe uzak), \dots, (Sağ Ön Araç - 3. kademe uzak), \dots, (Sağ Ön Araç - 0. kademe uzak), (Direksiyon - 3. kademe sol), \dots, (Hız - 3. kademe yüksek), (Hız - 2. kademe yüksek), (Hız - 1. kademe yüksek), (Hız - 0. kademe yüksek), (Şerit - En sol), (Şerit - Orta), (Şerit - Sağ)\}$

Bir başka ifadeyle

$\phi(s_i) = \{0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0\}$

Denklemler 3.33'e

$$\mu_E = \frac{1}{m} \sum_{i=1}^m \sum_{t=0}^{\infty} \gamma^t \phi(s_t^{(i)})$$

araçtan topladığımız durum dizilerini uygularsak çizelge 3.8'deki μ_E vektörüne ulaşabiliriz.

Çizelge 3.8. μ_E vektörü

0	1	2	3	4	5	6	7
Sol Ön Araç							
0	0.038	0.063	0.052	0.029	0.016	0.018	0.009
Sağ Arka Araç							

0.044	0.121	0.190	0.093	0.056	0.048	0.038	0.025
Sağ Ön Araç							
0.012	0.178	0.166	0.102	0.085	0.038	0.33	0.011
Ön Araç							
0	0.065	0.288	0.145	0.092	0.047	0.071	0.011
Sol Arka Araç							
0.026	0.059	0.054	0.052	0.037	0.022	0.019	0.007
Arka Araç							
0	0.151	0.236	0.197	0.068	0.107	0.039	0.020
Hız							
0.030	0.025	0.061	0.065	0.109	0.133	0.176	0.341
Yola göre yön							
0.995	0.001	0.0005	0.001	0.0004	0.0015	0.0003	0.0018
Şerit							
0.647	0.309	0.044					

Simülâtörün tamamen rasgele bir davranış biçimiyle başladığı eğitimin her adımında da aynı şekilde bir $\mu^{(i)} = \mu(\pi^{(i)})$ vektörü oluşturulacaktır. Burada $i = 0,1,2,\dots$ adımları temsil etmektedir. Yöntemin temel amacı μ vektörünü μ_E vektörüne mümkün olduğunca yakınlaştırmaktır. Bölüm 3.4.4, çizelge 3.7'deki prosedüre göre yöntemin adımlarını inceleyecek olursak,

- a.** Rasgele bir $\pi^{(0)}$ davranış biçimi seç ve çalıştır, $\mu^{(0)} = \mu(\pi^{(0)})$ 'ı hesapla ve $i=1$ yap.

Bu adımda, pekiştirme fonksiyonun eğitimi başlatılmaktadır. Simülâtör tamamen rasgele seçtiği bir davranış biçimiyle başlamakta ve hareket etmektedir. Oluşan durum dizilerinden ϕ özellik vektörü oluşturulmakta ve μ vektörü hesaplanmaktadır.

- b.** $t^{(i)} = \max_{w: \|w\|_2 \leq 1} \min_{j \in \{0, \dots, (i-1)\}} w^T (\mu_E - \mu^{(j)})$ denklemi hesaplanır burada $w^{(i)}$ eşitliği maksimum yapan w değeridir.

Bu adımda bir optimizasyon problemi çözülmektedir. Burada amaç $R = (w^{(i)})^T \phi$ pekiştirmesini, uzmana göre en uygun değerine getirmeye çalışmaktır. İfadedeki norm-2 kısıtlamasından dolayı bu problem lineer programlamayla çözülemez, ancak quadratik programlamayla çözülebilir. Böyle bir optimizasyon problemini çözebilmek için MATLAB® dahilinde araçlar mevcuttur ve bu araçlar kullanılarak her adımda problem çözülmüştür.

- c.** Eğer $t^{(i)} < \varepsilon$ ise, yöntem sonlandırılır.

İterasyonun sonlandırılabilmesi için sonucun bir eşik değerinin altına düşmesi beklenmektedir. Burada eşik değeri, 0.01 olarak kabul edilmiş ve 2400 adımda bu eşik değerinin altına ulaşılmıştır. Eşik değerinin altına düştüğü zaman, bulunan pekiştirme fonksiyonunun uzmanın pekiştirme fonksiyonuna yakınsadığı kabul edilmiştir.

- d.** RL algoritmasını kullanarak, $R = (w^{(i)})^T \phi$ pekiştirmesiyle $\pi^{(i)}$ en uygun davranış biçimi hesaplanır.

Kullandığımız pekiştirmeli öğrenme yöntemine bulduğumuz yeni pekiştirme fonksiyonunun yerleştirilmesi, yeni bir davranış biçiminin oluşturulması anlamına gelmektedir. Bu yeni davranış biçimi $\pi^{(i)}$ olarak ifade edilir, ve bu davranış biçimi doğrultusunda simülatör tekrar çalıştırılarak yeni durum dizileri elde edilir.

- e.** $\mu^{(i)} = \mu(\pi^{(i)})$ 'yi hesapla.

Yeni μ vektörü hesaplanır.

- f.** $i = i + 1$ yapılır ve 2. adıma geri dönülür.

Yöntemin çalıştırılması sonucu oluşan ω vektörü Çizelge 3.9'daki gibidir. Bu vektör vasıtasıyla herhangi bir durum için, o durumun pekiştirmesi

$$R = (w^{(i)})^T \phi$$

denkleminden hesaplanabilir.

Çizelge 3.9. ω vektörü

<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>
Sol Ön Araç							
0.002	0.004	0.003	0.006	0.004	0.007	0.002	0.004
Sağ Arka Araç							
0.002	0.005	0.003	0.003	0.004	0.003	0.003	0.003
Sağ Ön Araç							
0.002	0.014	0.013	0.009	0.008	0.005	0.002	0
Ön Araç							
-0.096	0.003	0.0789	0.0256	0.008	0.003	0.005	0.001
Sol Arka Araç							
0.001	0.005	0.005	0.005	0.004	0.004	0.002	0.001
Arka Araç							
-0.088	0.024	0.0234	0.0126	0.007	0.011	0.004	0.001
Hız							
0.004	0.002	0.001	0.007	0.009	0.011	0.018	0.0412
Yola göre yön							
0.0886	0.001	0	0.001	0	0.001	0	0
Şerit							
0.0654	0.0432	0.004					

3.7.4. Alt Görevlerin Otomatik Belirlenmesi

Önerilen modelde ikinci adım alt görevlerin belirlenerek, problemin birden fazla pekiştirmeli öğrenme problemi haline dönüştürülmesidir. Alt görevlerin belirlenmesinde kullanılan yöntem, ısrarcı aralıkların belirlenmesi yöntemidir. Yöntem temelde, uzman sürücünün hangi durumlarda ısrarla durduğunu, hangi durumları ise geçiş için kullandığını belirlemektedir. Bu hesaplama ise, durum değişkeninin durum dizisi içindeki ardışıklık olasılığının, aynı durum değişkeninin toplam durum dizisi içindeki marjinal olasılığına olan oranına göre belirlenmektedir. Bu oran bölüm 3.4.3.3'te değinilen Kullback-Leibler ıraksaması yöntemine göre belirlenmekte ve sonuç olarak her durum değişkeni için denklem 3.32'deki

$$Pers(S_j) = \text{sgn}(A(j, j) - P(S_j))SKL(A(j, j), P(S_j))$$

ifadeden yola çıkılarak bir ısrarcılık puanı ataması yapılmaktadır. Uzman durum dizisine uygulandığından her durum için ısrarcılık puanı tablosu çizelge 3.10'daki gibi ortaya çıkmaktadır.

Çizelge 3.10. Her durum için ısrarcılık puanları matrisi

0	1	2	3	4	5	6	7
Sol Ön Araç							
0	0.001	0.007	0.006	0.001	0	0	0
Sağ Arka Araç							
0.025	0.005	0.021	0.005	0.002	0.001	0.001	0.002
Sağ Ön Araç							
0	0.026	0.017	0.007	0.013	0.001	0.001	0.001
Ön Araç							
0	0.031	0.178	0.046	0.022	0.004	0.015	0
Sol Arka Araç							
0.003	0.009	0.005	0.006	0.005	0.001	0.001	0
Arka Araç							

0	0.071	0.096	0.072	0.012	0.054	0.007	0.002
Hız							
0.034	0.002	0.013	0.008	0.023	0.036	0.049	0.207
Şerit Durumu							
0.693	0.247	0.033					

Elde edilen ısrarcılık puanları içinde eşik değerinin üzerinde kalan değerler bizim için alt görev anlamı taşımaktadır. Bu alt görevlerin ne olduğu ve anlamları Çizelge 10'da açıklanmaktadır.

Çizelge 3.11. Alt görev oluşumları

Alt Görev	Alt Görev Başlangıç Durum Değişkeni	Alt Görev Bitiş Durum Değişkeni	Açıklama
ag0	<i>Bitiş sınır değerlerinden birisi</i>	<i>Bitiş sınır değerlerinden birisi</i>	<i>Hiçbir şey yapma</i>
ag1	<i>Herhangi bir durum</i>	<i>8 kademe hız</i>	<i>Maxl hız seyretme</i>
ag2	<i>Herhangi bir durum</i>	<i>Öndeki Araç-2 kademe uzak</i>	<i>Takip Mesafesi</i>
ag3	<i>Orta şerit veya sağ şerit</i> <i>Sol Şerit veya Orta şerit</i>	<i>Sol şerit veya orta şerit</i> <i>Orta şerit veya sağ şerit</i>	<i>Şerit değiştirme</i>

Alt görev belirleme sisteminin genel çalışma prensibi ise şu şekilde oluşmaktadır:

- Her adımda bir alt-görev değerlendirmesi gerçekleştirilir:
 1. Bir alt görev seçilebilir.
 2. Seçili bir alt görev devam ettirilebilir.
 3. Seçili bir alt görev sonlandırılır ve yeni bir alt görev seçilebilir.
 4. Seçili bir alt görev tamamlanır ve yeni bir alt görev seçilebilir.
- Seçilen alt göreve ait durum ve eylem uzayı göz önünde bulundurularak alt eylemler seçilir.

1. Alt görev tamamlanana kadar devam ettirilebilir.
2. Alt görev yarıda kesilebilir.

Burada hem alt görev seçimi, hem de alt görevlerin gerçekleşmesiyle ilgili ayrı geri yayılım ağları çalıştırılmaktadır. Alt görevin seçiminden sorumlu bir geri yayılım ağı mevcuttur, buna ek olarak her alt görevin de kendine ait bir geri yayılım ağı mevcuttur. Alt görev seçimi ile ilgili ağlar her adımda çalıştırılmaktadır. Bu ağların çalıştırılmasıyla içinde bulunan alt görevin tamamlanıp tamamlanmayacağına karar verilmesi, eğer tamamlandıysa yeni alt görevin seçilmesi gibi fonksiyonlar gerçekleştirilmektedir. Örneğin, sol şeride geçme alt görevine karar verildiyse, normal şartlar altında araç bulunduğu şeritten bir sol şeride geçince görev otomatik olarak tamamlanacak ve yeni alt görevin seçimi aşamasına geçilecektir. Fakat, sol arka taraftan gelen aracın tahmin edilenden daha hızlı yaklaştığı bir durumda, sol şeride geçme alt görevi yarıda kesilerek alt görevin başlatıldığı şeride geri dönülebilir. Böyle bir durumda alt görev tamamlanana kadar alt görevi devam ettirmek kazayla sonuçlanabilecek bir durum dizisi oluşturacaktır. Alt görev seçimi yapan geri yayılım ağı global olarak tüm girişleri kullanmaktadır. Fakat alt görev bir kez seçildiği zaman, alt görevi gerçekleştirecek alt seviye eylemlerin seçiminde kullanılacak durum ve eylem uzayı oldukça sınırlı bir uzay olacaktır. Hangi alt görevin hangi durum uzayı içinde çalıştığının hesaplanmasında, alt görevin sınırlarını belirleyen durum değişkenleri ile diğer durum değişkenlerinin ilintisi temel alınmıştır. Elde ettiğimiz sonuçlara göre alt görevler ve durum-eylem uzayları şu şekilde oluşmuştur.

ag1 – Şerit değiştirme

1. Şerit durumu
2. Kabul edilen şerit
3. Yola göre yön
4. Yola göre yön işaret – Değişikliğin yönü
5. Hız kademesi
6. Eylem

ag2 – Maksimum hızda ilerleme

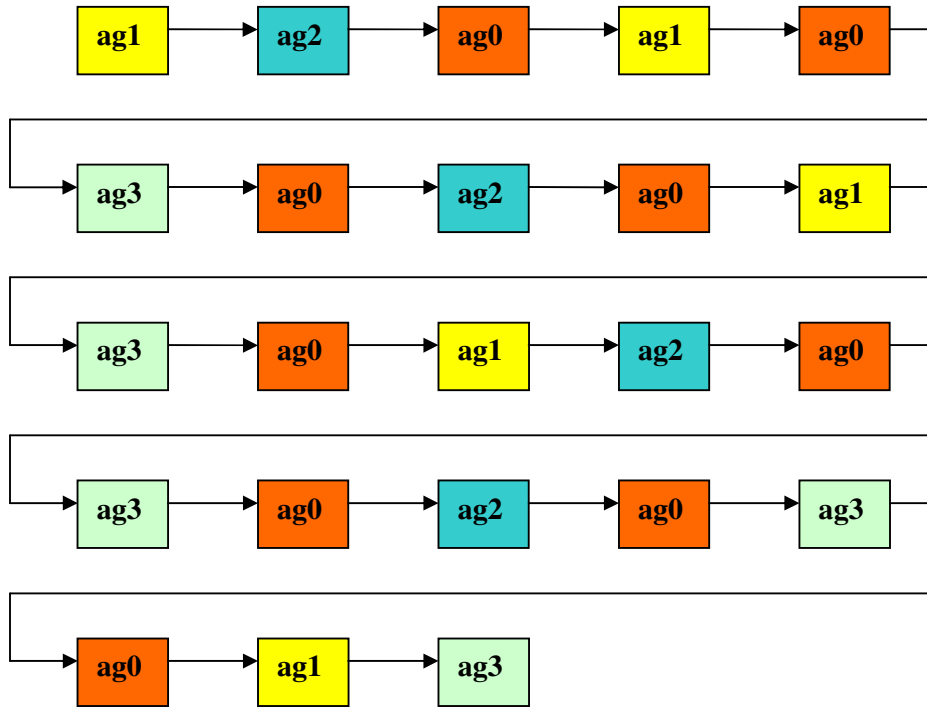
1. Hız kademesi
2. Öndeki araçla mesafe
3. Eylem

ag3 – Takip mesafesini koruma

1. Hız kademesi
2. Öndeki araçla mesafe
3. Eylem

En üstteki global geri yayılım ağının ve alt görevleri gerçekleştiren yerel alt görev ağlarının nasıl oluşturulduğuna eylem seçimi ve güncelleme bölümünde değinilecektir.

Araçtan toplanan ham verinin durum dizisi, aynı zamanda bir alt görev dizisi olarak da ifade edilebilir. Örnek olarak elimizde toplamış bulunduğumuz 1047 kayıtlı, yani 17 dakika araç sürmeye karşılık gelen bir durum dizisini şekil 3.49'daki gibi 23 elemanlı bir alt görev dizisi olarak ifade edebiliriz. Bu alt görevlerden dokuzu ag0 – yani sadece bulunduğu durumu koruyan alt görevdir, ve her hangi bir eylem üretimine dolayısıyla alt görev ağlarının çalıştırılmasına yol açmaz. Bu diziden de görülebileceği gibi, uzman sürücü belirli alt görevler arasında sürekli hareket etmektedir.



Şekil 3.49. 1047 kayıtlı durum dizisinin alt görev dizisi şeklinde ifadesi

3.7.5. Eylem Seçimi ve Güncelleme

3.7.5.1. Anlık Koşul

- *Algılayıcı girişleri:*

Sistemin algılayıcı girişleri, bölüm 3.7.1.2'de ayrıklaştırdığımız durum – eylem dizilerinden elde edilmektedir. Örneğin:

$$s_0 = \{(Arka Araç - 5. kademe uzak), (Öndeki Araç - 3. kademe uzak), (Sağ Arka Araç - 5. kademe uzak), (Sağ Ön Araç - 3. kademe uzak), (Sol Arka Araç - 8. kademe uzak), (Sol Ön Araç - 8. kademe uzak), (Direksiyon - 3. kademe sol), (Hız - 3. kademe yüksek), (Şerit - En sol)\}$$

- *Çalışan hafıza elemanları:*

Çalışan hafıza elemanları, geçmişe ait bilgileri depolayarak, ağ girişlerinde kullanılmasını sağlamaktadır. Çalışmamızda geçmişe doğru 8 adet durum bilgisi çalışan hafızada tutulmaktadır. Geçmişe doğru 7 adet durum bilgisinin kullanılmasının sebebi, ayrıklaştırmanın 8 olarak alınmasıdır. Bu şekilde, geçmişe doğru bakıldığında en düşük çözünürlükteki hareketin de yapı içinde değerlendirilebilmesi hedeflenmiştir. Yapısı

$$\mathcal{C}.H. = \{S_{t-7}, S_{t-6}, S_{t-5}, S_{t-4}, S_{t-3}, S_{t-2}, S_{t-1}\}$$

- *Hedef yapısından bir eleman:*

Hedef yığının en tepesinde yer alan görev o an için aktif görevdir. O görevin sınırları ve o görevin algılayıcı girişleri Q-ağının eğitiminde aktif rol oynamaktadır. Yapısı

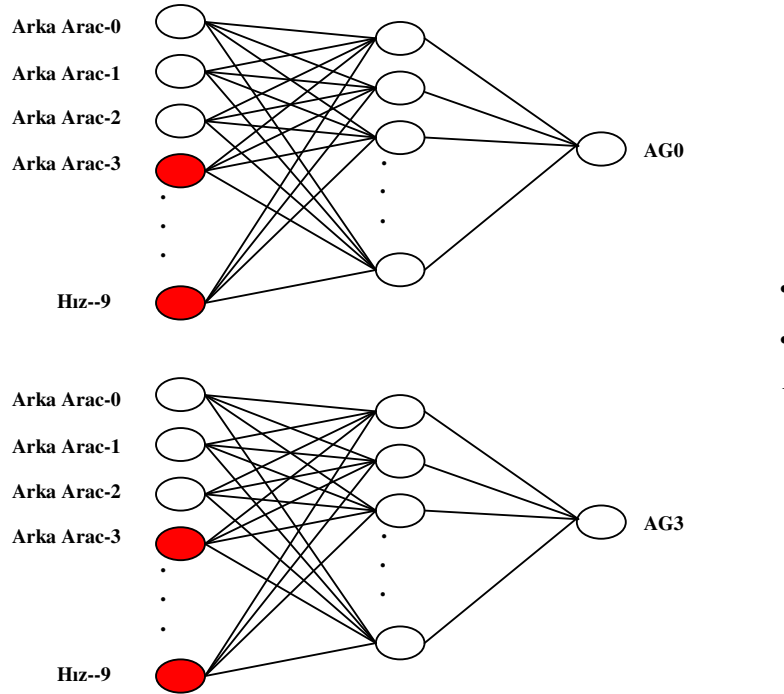
$$A.G. = \{ag0, ag1, ag2, ag3\}$$

şeklindedir.

3.7.5.2. Alt Görevin Seçimi

Model, her t anı için, global alt görev seçim ağlarının seçimi veya devamı için bir değerlendirme yapmak durumundadır. Bu global ağın 3 temel fonksiyonu söz konusudur.

- Bir alt görev seçebilir.
- Seçilen alt görevi devam ettirebilir.

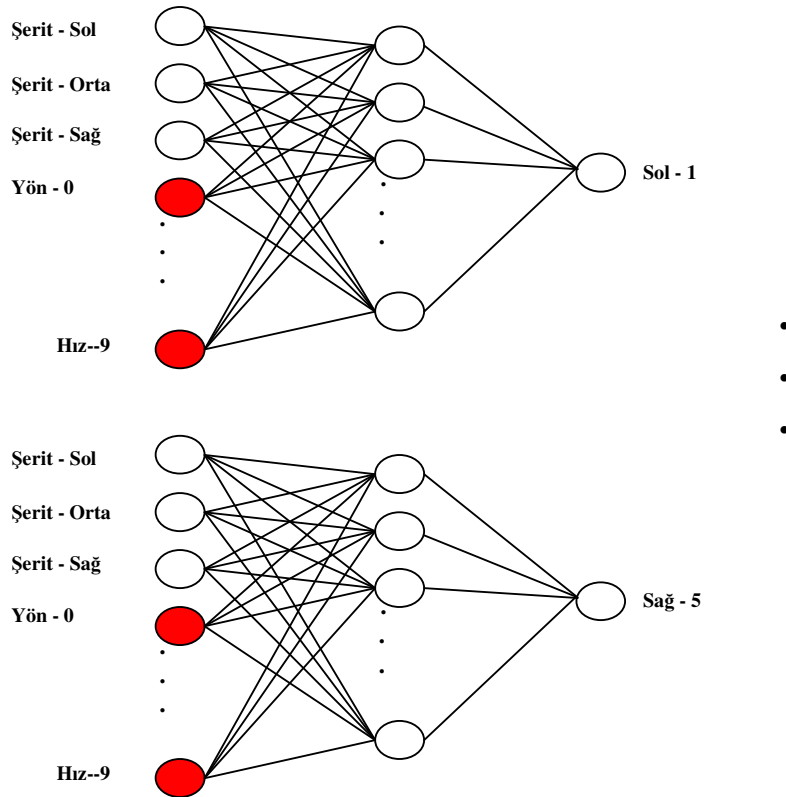


Şekil 3.50. Global geri yayılım ağı, alt görevlerin seçimi ve denetiminden sorumludur.

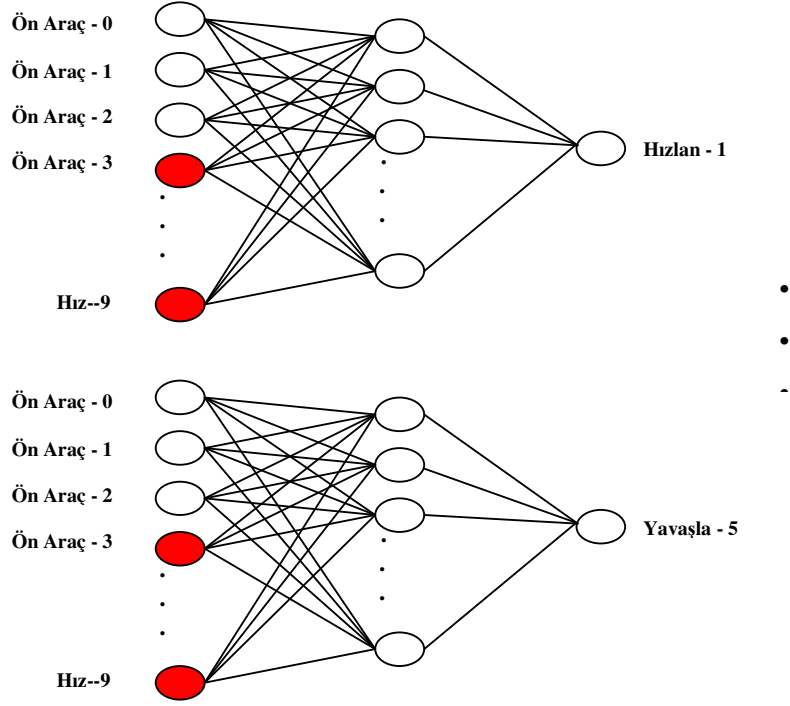
- Seçilen alt görevi sonlandırabilir

Şekil 3.50'deki global geri yayılım ağı, alt görev seçimini ve denetimini doğru bir şekilde sürdürebilmek için, tüm durum değişkenlerini giriş olarak kabul etmelidir. Alt görevlerin durum uzayları, kendi görevleri doğrultusunda sınırlandırılmış olduğu için, ortamda oluşan ve alt görev uzayında yer almayan diğer durum değişkenlerindeki değişimler doğrultusunda, yeni hareket stratejisini belirlemek global geri yayılım ağının görevidir. Alt görev ise yalnızca kendi durum ve eylem uzayında, başlangıç noktasından bitiş noktasına doğru hareket edecektir. Alt görev tamamlandığı zaman *alt-görev-tamamlandı* bayrağı aktif olacak ve yeni alt görev seçimi için global seviyede bir komut oluşturulacaktır.

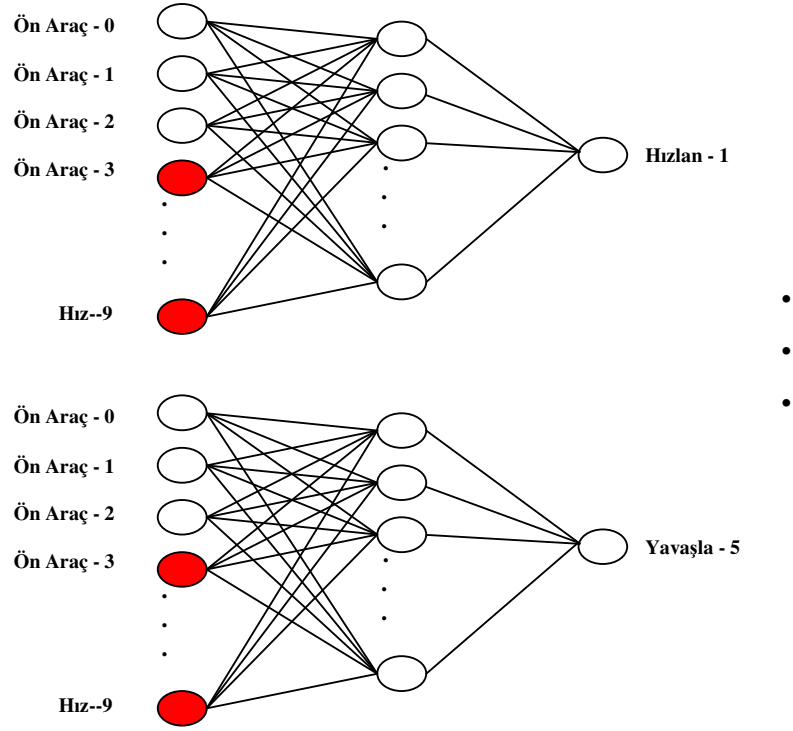
Çalışmada oluşturulan alt görev geri yayılım ağları ise şekil 3.51, 3.52 ve 3.53'deki gibi olacaktır:



Şekil 3.51. Şerit değiştirme alt görev ağı.



Şekil 3.52. Maksimum hıza erişme alt görev ağı.



Şekil 3.53. Takip Mesafesine ulaşma alt görev ağı.

3.7.5.3. Q-Değerlerinin Belirlenmesi

Alt Seviye

Bu aşamada elimizdeki durum dizilerinden, alt seviyede durumla ilişkili tüm olası alt görevlerin değerleri hesaplanır. Durum bilgisi şekil 3.48'deki global geri yayılım ağına uygulanarak her alt görev için bir çıkış değeri oluşturulur. Burada dikkat edilmesi gereken nokta her alt görev için ayrı bir ağın oluşturulduğudur, ayrıca her alt görev içinde farklı geri yayılım ağları oluşturularak alt görevlerin sınırlandırılmış durum-eylem uzayında çalışması sağlanmıştır. Alt görev seçildikten sonra o alt göreve ait geri yayılım ağı çalıştırılarak geçerli eylem seçilmektedir.

Geri yayılım ağı ağırlıkları, başlangıçta sıfır değerine çok yakın bir değere atanmıştır. Eğitim süresince bu ağ ağırlıkları en uygun değerlerine doğru yakınsayacaktır.

Üst Seviye

Üst seviyedeki eylem kurallarına dayalı olarak, tüm olası eylemler (b_1, b_2, \dots, b_m) bulunur.

Üst seviyedeki eylem değerlerinin belirlenmesi için anlık koşula uyan kurallara bakılmaktadır. Kuralın koşulu anlık duruma karşılık gelmektedir. Bu koşul için denklem 3.34'ten

$$Kazanç_j = \frac{c_7 + PM(j)}{c_8 + PM(j) + NM(j)}$$

ve denklem 3.35'ten

$$Maliyet_j = \frac{j - kuralı - uygulama - süresi}{kurallar - ortalama - uygulama - süresi}$$

ve denklem 3.36'dan fayda ölçümü belirlenir. Bu noktada maliyet hesabı yapılamayacağından maliyet hesabı sabit bir değer olarak alınmıştır. Böylece, fayda ölçümü doğrudan kazançla orantılı bir hale gelmiştir.

$$U_j^T = kazanç_j - v * maliyet_j$$

Bu ifadenin anlamı ise genel olarak şu şekilde açıklanabilir. Herhangi bir koşul için zaman içinde birden fazla eylem gerçekleştirilmiş olabilir. Bu kurallardan hangisinin çalıştırılacağına, daha doğrusu en uygun eylemi seçen kuralın hangisi olduğunun belirlenmesinde kazanç ölçümü kullanılmaktadır. Kazanç ölçümünün temelinde ise, o kuralın olumlu kullanılma istatistiği yer almaktadır. Örneğin, araç kullanılırken zaman zaman takip mesafesini geçecek eylemler de gerçekleştirilebilir. Bu kural deposuna bir kural olarak atılabilir, fakat olumsuz sonuçlar getirebildiği için olumsuz çakışma sayısı artırılır, böylece toplamda getirdiği kazanç düşürülmüş olur, bu da bize o kuralın faydalılığı konusunda bir bilgi sağlamaktadır.

3.7.5.4. Alt Görev - Eylem Seçimi

Üst Seviye

a_i (alt seviyeden) ve b_j (üst seviyeden) alt ve üst seviye eylemlerine bir birleştirme algoritması uygulayarak, **en uygun eylemi seç.**

Denklem 3.37'deki Luce'nin seçim aksiyonuyla üst seviyeden bir eylem seçilir.

$$p(i | x) = \frac{e^{u_i/\tau}}{\sum_j e^{u_j/\tau}}$$

Alt Seviye

Alt seviye eylem seçimi denklem 3.38'deki Softmax eylem seçimi yöntemiyle seçilir.

$$p(a | x) = \frac{e^{Q(x,a)/\tau}}{\sum_i e^{Q(x,a)/\tau}}$$

Denklem 3.38'deki Boltzmann dağılımına geri yayılım ağı çıkışları olan çizelge 3.11'deki değerler verilir.

Çizelge 3.12. Her bir alt görev için ağ çıkışları

ag0	ag1	ag2	ag3
0.1001	0.4456	0.0872	0.0449

ve Softmax Eylem seçimi şu şekilde gerçekleştirilir.

$$\text{Seçilecek eylem } a \rightarrow \max_a \frac{e^{Q_{t-1}(a)/\tau}}{\sum_b e^{Q_{t-1}(b)/\tau}}$$

burada sıcaklık faktörünün etkisi ise şu şekilde gerçekleşir. Farklı τ 'lar için deneyecek olursak,

$\tau = 0.9$ için

0.0043 0.0192 0.0038 0.0019 0.0001

$\tau = 0.4$ için

0.0101 0.0240 0.0098 0.0088 0.0079

Bu durumda ikisi içinde seçilen alt görev, ag0-şerit değiştirme alt görevi olacaktır. Sürücümüzün bu durum için gerçekleştirdiği eylem ise (Direksiyon Sağa – 1 kademe) olmuştur.

Sürücünün seçtiği alt görev burada seçilmiş eylem olarak alınacak ve global ağ eğitilecektir. Seçilen alt görevin Q' değeri ile ağın bir önceki Q değeri arasındaki fark geri yayılım ağının hatası olarak kabul edilecektir ve bu hata doğrultusunda ağ tekrar eğitilecektir. Burada dikkat edilmesi gereken 2 nokta vardır, birincisi yalnızca alt görevi seçilen ağ eğitilecektir (bölüm 3'de değinilen hususlardan dolayı) ve ikincisi ağın alt görev seçim davranışı sürücünün davranışına yakınsayana kadar, davranışların sürücü tarafından seçilmesidir.

İkinci adımda, seçilen alt görevin yerel geri yayılım ağı eğitilecektir. Örnek olarak, çizelge 3.13'teki şerit değiştirme dizisini ele alalım ve bu doğrultuda şerit değiştirme ağının eğitiminin nasıl gerçekleştirileceğine bakalım.

Çizelge 3.13. Şerit değiştirme alt görevi dizisi

Dizi No	Şerit Durumu	Kabul edilen şerit	Hız Kademesi	Yola göre yön	Eylem
23	Sağ	Sağ	6	0	Sol - 2
24	Ara	Sağ	6	-2	Sol - 2
25	Ara	Sağ	6	-4	Sol - 2
26	Ara	Sağ	6	-6	Sol - 3
27	Ara	Sağ	6	-9	-----
28	Ara	Orta	6	-9	Sağ - 3
29	Ara	Orta	6	-6	Sağ - 3
30	Ara	Orta	6	-3	Sağ - 1
31	Ara	Orta	6	-2	Sağ - 1
32	Ara	Orta	6	-1	Sağ - 1
33	Orta	Orta	6	0	-----

Burada bu çizelgeyi ,durum dizisi olarak ifade etmemiz gerekirse

$$S_{23}^{ag0} = \{(\text{Şerit Durumu} - \text{Sağ}), (\text{Kabul edilen şerit} - \text{Sağ}), (\text{Hız} - 6 \text{ kademe yüksek}), (\text{Yola göre yön işaret} - 0), (\text{Yola göre yön} - 0 \text{ kademe}), (\text{Eylem} - 2 \text{ kademe sol})\}$$

·
·
·

$$S_{33}^{ag0} = \{(\text{Şerit Durumu} - \text{Orta}), (\text{Kabul edilen şerit} - \text{Orta}), (\text{Hız} - 6 \text{ kademe yüksek}), (\text{Yola göre yön işaret} - 0), (\text{Yola göre yön} - 0 \text{ kademe}), (\text{Eylem} - \text{Hiç bir şey yapma})\}$$

olarak ifade edebiliriz.

Eylem seçimi, alt görev seçimiyle tamamen aynı mantıkta çalışmaktadır. Yine aynı şekilde alt ve üst seviyelerde seçimler ve güncellemeler gerçekleştirilmektedir. Bilişsel eleman sürücünün seçtiği davranışı kendisi seçmiş gibi davranacak ve ağırları bu doğrultuda güncelleyecektir.

İki Seviyeden Çıkış Alınması

Eğer üst seviyede hazır bir kural varsa çıkış oradan alınır, eğer yoksa alt seviyeden seçilir.

3.7.5.5. Güncelleme

Alt seviye güncellemesinde, örnek dizilerden elde edilen yeni alt görev – eylem ve pekiştirme fonksiyonundan elde edilen durum pekiştirmesi kullanılır.

Başarı Oranı

Seçilen alt görev ve uzman sürücü tarafından seçilen alt göreve, seçilen a eylemi ve uzman tarafından seçilen a eylemine göre başarı ölçütü güncellenir.

Alt Seviyenin Güncellenmesi

Q-öğrenme-geri yayılım algoritması güncelleme prosedürü bölüm 3.2.6.2.'den

$$Q' \leftarrow r + \gamma \cdot \text{Max}\{Q(s', k \mid k \in \text{aksiyonlar})\}$$

Yeni Q değeri hesaplanır. Bunun için gereken r değeri pekiştirmeli öğrenme fonksiyonundan elde edilir. $\gamma = 0.9$ olarak seçildi.

Fayda ağını geri yayılım hatası ΔQ_a ile s girişi doğrultusunda düzeltilir,

$$\Delta Q_a = \begin{cases} Q' - Q_a & a_i = a \\ 0 & \text{veya} \end{cases}$$

Hangi eylemin ağı güncelleniyorsa o ağ için geri yayılım hatası yukarıdaki ifade olarak alınır ve ilgili ağ geri yönde (hata düzeltme ve ağırlıkları ayarlama yönünde) çalıştırılarak ağırlıkları düzenlenir

Üst Seviyenin Güncellenmesi.

Bölüm 3.1.3.2'de değinilen kural-seçim-iyileştir algoritması çalıştırılır. Buna göre önce eldeki (s, s', r, a) kümesi (burada s , a eylemi gerçekleştirilmeden önceki durumu, ve r 'de a eylemi gerçekleştirildikten

sonraki pekiştirmeyi temsil eder) göz önünde bulundurularak başarı ölçütü belirlenir.

$$\gamma \max_b Q(s', b) + r - Q(x, a) > \text{eşik}$$

eşitsizliği a eyleminin (kurala göre seçilmiş) kabul edilebilir derecede iyi olup olmadığını belirler. Eğer sol taraftaki ifade eşik değerinden büyükse sonuç olumlu, değilse sonuç olumsuzdur.

Her üst seviye güncelleme adımında üç temel mekanizma çalıştırılır:

- Kural eklenebilir.
- Kural genelleştirilebilir.
- Kural özelleştirilebilir.

Kural Ekleme

Kural eklemenin bölüm 3.1.3.2'deki tanımına tekrar göz atalım. Eğer anlık adım olumlu ise (anlık olumluluk ölçütüne göre) ve eğer üst seviyede bu kuralı kapsayan bir başka kural yoksa (hem koşulu hem de eylemi), kural düzenlenir, $C \rightarrow a$, burada C tüm boyutların değerlerini tam olarak belirtir.

Alt seviyede geçerli olan alt görev mantığı, üst seviye için de geçerlidir. Bir şerit değiştirme alt görevi dizisi ele alalım, dizinin ham verileri çizelge 3.13'teki gibidir.

$$S_{23}^{ag0} = \{(\text{Şerit Durumu} - \text{Sağ}), (\text{Kabul edilen şerit} - \text{Sağ}), (\text{Hız} - 6 \text{ kademe yüksek}), (\text{Yola göre yön işaret} - 0), (\text{Yola göre yön} - 0 \text{ kademe}), (\text{Eylem} - 2 \text{ kademe sol})\}$$

•
•
•

$$S_{33}^{ag0} = \{(\text{Şerit Durumu} - \text{Orta}), (\text{Kabul edilen şerit} - \text{Orta}), (\text{Hız} - 6 \text{ kademe yüksek}), (\text{Yola göre yön işaret} - 0), (\text{Yola göre yön} - 0 \text{ kademe}), (\text{Eylem} - \text{Hiç bir şey yapma})\}$$

Kuralın koşulu üst seviyede, bu alt seviye durum ifadelerine bağlanacaktır, ilk gelen 23 numaralı durum, üst seviye kuralın koşulunu şu şekilde oluşturacaktır:

IF (*Şerit Durumu – Sağ*) **and** (*Kabul edilen şerit – Sağ*) **and** (*Yola göre yön işaret – 0*) **and** (*Yola göre yön - 0*) **and** (*Hız – 6 kademe yüksek*) **THEN** (*Sol - 2*)

Üst seviye kural deposu başlangıç aşamasında tamamen boş olduğu için, dolayısıyla bu kuralı karşılayabilecek bir kural olmadığı için bu yeni kural üst seviye eylem kural deposuna atılacaktır. Deponun 1 numaralı adresinde artık bu kural yer almaktadır. Üst seviyenin başlangıç aşamasında tamamen boş olduğunu düşünürsek, gelen ilk durumda oluşturulan ilk kural, kural deposuna çizelge 14'deki gibi atılacaktır:

Çizelge 3.14. Üst Seviye Kural Deposu

Kr. No	Kural Tanım
1	IF (<i>Şerit Durumu – Sağ</i>) and (<i>Kabul edilen şerit – Sağ</i>) and (<i>Yola göre yön işaret – 0</i>) and (<i>Yola göre yön - 0</i>) and (<i>Hız – 6 kademe yüksek</i>) THEN (<i>Sol - 2</i>)

Gelen her yeni durumda, yukarıda değindiğimiz ekleme, özelleştirme veya genelleştirme olarak üç temel işlem gerçekleştirilir. Kural deposunda en az bir kuralın oluşturulmasından itibaren gelen her yeni durum için, o durumla kural deposundaki kural koşulları karşılaştırılacaktır, eğer özelleştirme veya genelleştirme kriterlerine uyarsa, yeni kural eklenmeyecek ve varolan kurallar yeniden düzenlenecektir. Eğer özelleştirme ve genelleştirme kriterlerine uymuyorsa da yeni bir kural olarak depoya eklenecektir. Kural, eylem kural deposuna atıldıktan sonra, bu kurala ait istatistik güncellenir. Eğer kural denklem 3.1'de bahsedilen olumluluk ölçütüne göre başarılıysa, olumlu çakışma sayısı $PM_a(C)$, başarısızsa olumsuz çakışma sayısı $NM_a(C)$ artırılır. Buradaki olumluluk ve olumsuzluğu belirlenmesi için denklem 3.1'deki ifade kullanılır:

$$\gamma \max_b Q(s', b) + r - Q(s, a) > \text{eşik}$$

eşitsizliği a eyleminin (kurala göre seçilmiş) kabul edilebilir derecede iyi olup olmadığını belirler (Sun ve ark. 2001). Dikkat edilirse bu denklemin sol tarafındaki değerler çizelge 3.6'daki alt seviyede geri yayılım hatasını bulmada zaten kullanıldığı görülebilmektedir. Burada eşik değeri, literatürde varolan

çalışmalarda 0.08 olarak alındığı için, bu değer çalışmamızda da eşik değeri olarak kabul edilmiştir. İstatistiğin güncellenmesi ise şu şekilde gerçekleştirilir:

$$sta = sta + 1$$

burada sta PM veya NM 'yi temsil eder ve her döngü sonunda

$$sta = sta \times 0.9$$

ile indirgenir.

Burada kural deposunda henüz bir kural mevcut olduğu için yalnızca onun istatistiği güncellenecektir, yani

$$PM(C_1) = 0.9$$

olacaktır.

Kuralın özelleştirme ve genelleştirme kriterinin denetimi, bilgi kazanç ölçümü aracılığıyla gerçekleştirilmektedir. Bilgi kazanç ölçümünün hesabı ise şu şekilde gerçekleşecektir.

Bilgi kazanç Ölçümü

$$IG(A,B) = \log_2 \frac{PM_a(A)+1}{PM_a(A)+NM_a(A)+2} - \log_2 \frac{PM_a(B)+1}{PM_a(B)+NM_a(B)+2}$$

Burada A ve B aynı eylemi tetikleyen iki ayrı koşuldur. Ölçüm zorunlu olarak, A ve B değişik koşulları altında olumlu çakışmaların yüzdesini vermektedir. Eğer A , B 'ye göre, yüzdesini belirli bir değer üstüne taşıyabilirse, A 'nın B 'den daha doğru olduğu düşünülebilir. Algoritmada, eğer herhangi biri kural çakışma açısından diğer tümüne göre daha iyiye (yani, olası tüm giriş koşullarıyla çakışmaya göre), kural o durumda en başarılı kural olarak kabul edilir.

Genelleştirme

Genelleştirmenin bölüm 3.1.3.2'deki tanımına tekrar göz atalım. Eğer $IG(c,all) > eşik$ ve $\max_c IG(C',C) \geq 0$ ise, burada C kuralın anlık koşuluna, "all" hepsiyle-çakış kuralına (orijinal kural tarafından belirtilen aynı eyleme, fakat her duruma uyan koşula sahip) karşılık gelir ve C' , $C'=C$ artı bir

*değer**(yani C' giriş boyutu olarak bir fazlaya sahiptir) şeklinde oluşur, bunun anlamı, eğer anlık kural başarılı ise ve genelleştirilmiş koşul potansiyel olarak daha iyi ise o zaman genelleştirmenin gerçekleştirilmesinin doğru olacağıdır, bu durumda kuralın yeni (genelleştirilmiş) koşulu olarak $C^N = \arg \max_{C'} IG(C', C)$ seçilir. Burada '*all*' terimi, aslında giriş bağlamında gerçekleşebilecek tüm olası koşulları temsil etmektedir. Fakat, istatistiği güncellenen yalnızca kural deposunda varolanlar olduğu için, kural deposunda yer alan ve geçerli eyleme bağlanan tüm koşullar anlamına gelmektedir. Kural deposunun zaman içinde genişlediğini ve çizelge 3.15'deki duruma geldiğini düşünelim.

Çizelge 3.15. Üst Seviye Kural Deposu

Kr. No	Kural Tanım	PM(C)	NM(C)
1	IF (Şerit Durumu – Sağ) and (Kabul edilen şerit – Sağ) and (Yola göre yön işaret – 0) and (Yola göre yön - 0) and (Hız – 6 kademe yüksek) THEN (Sol - 2)	3.686	0
2	IF (Şerit Durumu – Ara) and (Kabul edilen şerit – Sağ) and (Yola göre yön işaret – 1) and (Yola göre yön - -2) and (Hız – 6 kademe yüksek) THEN (Sol - 2)	3.686	0
3	IF (Şerit Durumu – Ara) and (Kabul edilen şerit – Sağ) and (Yola göre yön işaret – 1) and (Yola göre yön - -4) and (Hız – 6 kademe yüksek) THEN (Sol - 2)	3.686	0
4	IF (Şerit Durumu – Ara) and (Kabul edilen şerit – Sağ) and (Yola göre yön işaret – 1) and (Yola göre yön - -6) and (Hız – 6 kademe yüksek) THEN (Sol - 3)	3.095	1.71
5	IF (Şerit Durumu – Ara) and (Kabul edilen şerit – Sağ) and (Yola göre yön işaret – 1) and (Yola göre yön - -9) and (Hız – 6 kademe yüksek) THEN (Hiçbirşey yapma)	3.095	1.71
6	IF (Şerit Durumu – Ara) and (Kabul edilen şerit – Sağ) and (Yola göre yön işaret – 1) and (Yola göre	3.686	0

	<i>yön - 9) and (Hız – 6 kademe yüksek) THEN (Sağ - 3)</i>		
.			
.	...		
.			
22	IF (<i>Şerit Durumu – Ara</i>) and (<i>Kabul edilen şerit – Orta</i>) and (<i>Yola göre yön işaret – 1</i>) and (<i>Yola göre yön - -2</i>) and (<i>Hız – 6 kademe yüksek</i>) THEN (<i>Sol - 2</i>)	3.095	1.71
.			
.	...		

Yeni gelen durumun koşulunun şu şekilde olduğunu düşünelim:

$C \rightarrow$ (*Şerit Durumu – Ara*) **and** (*Kabul edilen şerit – Orta*) **and** (*Yola göre yön işaret – 1*) **and** (*Yola göre yön - -2*) **and** (*Hız – 6 kademe yüksek*)

C koşulu için gerçekleştirilen eylem ise *Sol - 2* eylemidir.

Bu durumda *Sol - 2* eylemine yol açan varolan bütün koşullarla, anlık koşulumuz C' nin arasındaki bilgi kazanç oranı karşılaştırılır. Çizelge 3.16'dan görüldüğü gibi, 1,2 ve 3 numaralı kurallar *sol - 2* eylemiyle sonuçlanmaktadır. Eğer elde edilen oran belirli bir eşikten büyükse, yani

$$IG(C, all) > eşik$$

ise, genelleştirme gerçekleştirilebilir. Genelleştirmede dikkate alınan ikinci ölçüt

$$\max_C IG(C', C) \geq 0$$

eşitsizliğidir. Burada C' , C koşulunun boyutlarından birinin +/- bir birim değiştirilmesiyle oluşturulur. Çizelge 16'da olası C' ifadeleri görülebilir:

Çizelge 3.16. Durum değişkeni varyasyonları

C'	Koşul
1	(<i>Şerit Durumu – Ara</i>) and (<i>Kabul edilen şerit – Orta</i>) and (<i>Yola göre yön işaret – 1</i>) and (<i>Yola göre yön - -2</i>) and (<i>Hız – 6 kademe yüksek</i>)

2	(Şerit Durumu – Sağ) and (Kabul edilen şerit – Orta) and (Yola göre yön işaret – 1) and (Yola göre yön - -2) and (Hız – 6 kademe yüksek)
3	(Şerit Durumu – Orta) and (Kabul edilen şerit – Orta) and (Yola göre yön işaret – 1) and (Yola göre yön - -2) and (Hız – 6 kademe yüksek)
4	(Şerit Durumu – Sol) and (Kabul edilen şerit – Orta) and (Yola göre yön işaret – 1) and (Yola göre yön - -2) and (Hız – 6 kademe yüksek)
5	(Şerit Durumu – Ara) and (Kabul edilen şerit – Sağ) and (Yola göre yön işaret – 1) and (Yola göre yön - -2) and (Hız – 6 kademe yüksek)
6	(Şerit Durumu – Ara) and (Kabul edilen şerit – Sol) and (Yola göre yön işaret – 1) an <i>(Yola göre yön - -2) and (Hız – 6 kademe yüksek)</i>
7	(Şerit Durumu – Ara) and (Kabul edilen şerit – Orta) and (Yola göre yön işaret – 0) and (Yola göre yön - -2) and (Hız – 6 kademe yüksek)
8	(Şerit Durumu – Ara) and (Kabul edilen şerit – Orta) and (Yola göre yön işaret – 0) and (Yola göre yön - -3) and (Hız – 6 kademe yüksek)
.	...
.	

Burada tüm olasılıklar denenerek

$$\max_{C'} IG(C', C) \geq 0$$

eşitsizliğinin sağlanması gerekir. Bu iki eşitsizliğin sağlandığı durumda genelleştirme gerçekleştirilebilir, bu durumda kuralın yeni (genelleştirilmiş) koşulu olarak $C^N = \arg \max_{C'} IG(C', C)$ seçilir.

Yeni kural şu şekilde oluşur:

$C^N \rightarrow$ **(Şerit Durumu – Ara) and (Kabul edilen şerit – Orta veya Sağ) and (Yola göre yön işaret – 1) and (Yola göre yön - -2) and (Hız – 6 kademe yüksek)**

Bu genelleştirilmiş kural, aslında sağdan orta şeride geçmekle, orta şeritten sol şeride geçmek arasında eylemsel açıdan bir fark olmadığını göstermektedir. İki kuralla kapsanacak bir koşulu tek kuralla ifade ederek hem depo alanından hem de işlem zamanından tasarruf etmektedir. Bu örneğimizde, 2 ve 22 numaralı kurallar depodan çıkarılarak, yeni kural depoya eklenecektir.

Özelleştirme

Genelleştirmenin olduğu bir durumda özelleştirmenin de olması gereklidir. Özelleştirmenin bölüm 3.1.3.2'deki tanımına tekrar göz atalım. Eğer $IG(C, all) < eşik$ ve $\max_C IG(C', C) > 0$ ise, burada C kuralın anlık koşuluna (anlık duruma ve koşula uyan), "*all*" hepsiyle çakış (orijinal kuralla aynı fakat her duruma uyan koşuluyla) kuralına karşılık gelir, ve $C' = C$ eksi bir değer (yani, C' giriş boyutlarında bir eksik boyuta sahiptir) şeklinde oluşur. Bunun anlamı, eğer anlık kural başarısızsa ve özelleştirilen kural ondan daha iyiye, kuralın yeni(özelleştirilmiş) koşulu olarak $C^N \arg \max_C IG(C', C)$ seçilir.

Yeni gelen durumun koşulunun şu şekilde olduğunu düşünelim:

$C \rightarrow (\text{Şerit Durumu} - \text{Ara}) \text{ and } (\text{Kabul edilen şerit} - \text{Orta veya Sağ}) \text{ and } (\text{Yola göre yön işaret} - 1) \text{ and } (\text{Yola göre yön} - -2) \text{ and } (\text{Hız} - 6 \text{ kademe yüksek})$

C koşulu için gerçekleştirilen eylem ise *Sol - 2* eylemidir. Bu durumda *Sol - 2* eylemine yol açan varolan bütün koşullarla, anlık koşulumuz C' nin arasındaki bilgi kazanç oranı karşılaştırılır. Eğer elde edilen oran belirli bir eşikten küçükse, yani

$$IG(C, all) < eşik$$

ise, özelleştirme gerçekleştirilebilir. Özelleştirmede dikkate alınan ikinci ölçüt ise

$$\max_C IG(C', C) > 0$$

eşitsizliğidir. Burada C' , C koşulunun "veya" içeren boyutlarından birinden bir değer eksiltmesiyle oluşturulur. Örneğin yeni gelen durumun çizelge 3.17'deki gibi iki adet olasılığı olacaktır:

Çizelge 3.17. Eksiltilmiş durum değişkenli koşul olasılıkları

C'	Koşul
1	(Şerit Durumu – Ara) and (Kabul edilen şerit – Orta) and (Yola göre yön işaret – 1) and (Yola göre yön - -2) and (Hız – 6 kademe yüksek)
2	(Şerit Durumu – Ara) and (Kabul edilen şerit – Sağ) and (Yola göre yön işaret – 1) and (Yola göre yön - -2) and (Hız – 6 kademe yüksek)

Burada tüm olasılıklar denenerek

$$\max_{C'} IG(C', C) > 0$$

eşitsizliğin sağlanması gerekir. Bu iki eşitsizliğin sağlandığı durumda özelleştirme gerçekleştirilebilir, bu durumda kuralın yeni (özelleştirilmiş) koşulu olarak $C^N = \arg \max_{C'} IG(C', C)$ seçilir, yani

$$C^N \rightarrow (\text{Şerit Durumu – Ara}) \mathbf{and} (\text{Kabul edilen şerit – Orta}) \mathbf{and} (\text{Yola göre yön işaret – 1}) \mathbf{and} (\text{Yola göre yön - -2}) \mathbf{and} (\text{Hız – 6 kademe yüksek})$$

Özelleştirmenin anlamı, genel ifadenin artık koşulu ifade etmekte yetersiz kalmasıdır. Genel ifade fazla eyleme sebep olmakta ve daha spesifik hale getirilmesi gerekmektedir.

3.7.5.6. Sonlandırma

Eğitimin sonlaması için başarı oranının eşik değerini geçmesi gerekmektedir. Eğer dizi sonlanmadıysa tekrar bölüm 3.7.4.3'ten devam edilir, aksi takdirde başarı oranı güncellenerek 3.7.4.1'e geçilir. Eğitimimiz için başarı oranı %80 olarak kabul edilmiş ve %80'i geçtiği durum için eğitim sonlandırılmıştır.

3.7.6. Deneysel Çalışma Sonuçları

Kurulan model, yaklaşık 4 saatlik araç sürmeye karşılık gelen 4892 örnekten oluşan bir durum dizisiyle çalıştırılmıştır. Bu durum dizisi üzerinde iki farklı deney gerçekleştirilmiştir. Birinci deneyde alt görev yapısı kullanılmamış, ikinci deneyde ise alt görev yapısı kullanılarak sonuçlar alınmış, algoritmanın sisteme olan katkısı ortaya çıkarılmıştır.

3.7.6.1. Deney I

Alt görev yapısı kullanılmadığı durum için, araç sürme problemi tek bir problem olarak düşünülmüş ve çözüm bu yapıya göre oluşturulmuştur. Problemi bütün olarak düşünmek, modelin içinde bulunduğu ortamda varolan tüm algılayıcı bilgilerinin sistemin girişi veya çıkışı olarak değerlendirilmesini gerektirir. Bu durum için kurulan alt seviye geri yayılım ağının temelde 121 girişi ve 29 çıkışı olacaktır. Fakat bölüm 3.4.1'de değinildiği üzere Markov olmayan durumun da çözümlenebilmesinde her algılayıcı girişi için belirli uzunlukta geçmişe ait bilginin de anlık duruma dahil edilmesi gerekmektedir. Model kurulumunda bölüm 3.7.4.1'de bahsettiğimiz şekilde her bir algılayıcı girişi için 7 adet geçmiş bilgisi anlık duruma dahil edilmiş ve böylece geri yayılım ağı giriş düğümü sayısı $121 \times 7 = 847$ olarak belirlenmiştir. Her bir eylem için 29 ayrı geri yayılım ağı kurulmuş, bu ağlar geçerli durum ve eylem için güncellenmiştir. Ağlar öncelikle ileri yönde çalıştırılarak bir eylem tavsiyesi üretmişlerdir. Üretilen eylem tavsiyesi ile sürücünün gerçekleştirdiği eylem karşılaştırılarak modelin başarı oranı ortaya çıkarılmıştır. Daha sonra sürücünün gerçekleştirdiği eylem doğru eylem olarak kabul edilerek, bu eyleme ait geri yayılım ağı güncellenmiştir. Geri yayılım ağı iç ağırlıklarının güncellenmesinde kullanılan geri yayılım hatası bölüm 3.2.6.2.'den

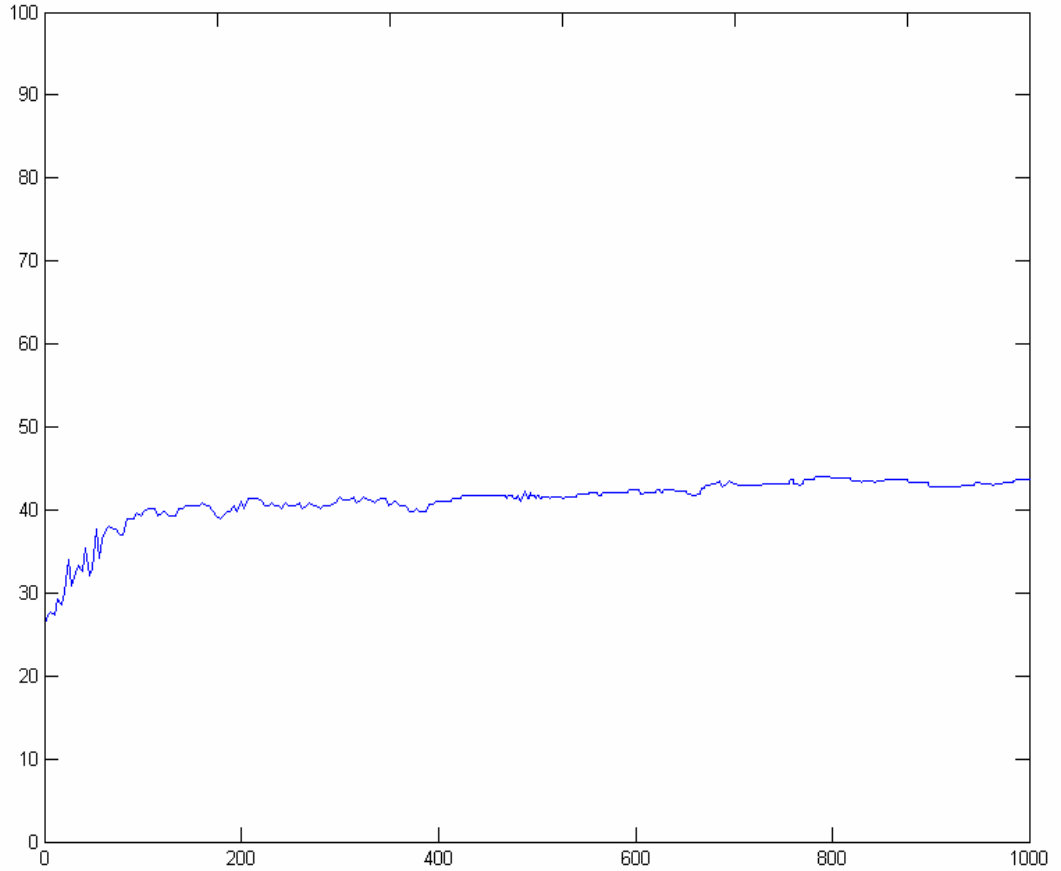
$$Q' \leftarrow r + \gamma \cdot \text{Max}\{Q(s',k \mid k \in \text{aksiyonlar})\}$$

ifadesiyle belirlenmektedir. Bu hata geri yayılım ağına verilerek geri yayılım ağı iç ağırlıkları eğitilmektedir.

Problemin bir bütün olarak düşünülmesiyle $847 \times 847 \times 1$ 'lik 29 adet geri yayılım ağı oluşturulmuştur. Ağların ileri yönde çalıştırılmasıyla giriş katmanı ile

gizli katman arasında, 'n' in düğüm sayısı olduğu durum için, $n*(2n-1)$ adet işlem gerçekleştirilmektedir. Gizli katman ile çıkış katmanı arasında ise $(2n-1)$ adet işlem söz konusudur. Bu durumda, her saniye için $n*(2n-1) + (2n-1)$ adet işlem gerçekleştirilmektedir. $847 \times 847 \times 1$ 'lik geri yayılım ağına $n= 847$ olduğundan dolayı toplamda 1.432.258 işlem gerçekleştirilmektedir. Bu işlemler 29 geri yayılım ağı için de ayrı ayrı gerçekleştirilmiştir, sonuçta her saniyede yaklaşık 41×10^6 işlem gerçekleşecektir. Deneyimizde 4982 örnek için geri yayılım ağları baştan sona çalıştırılmış ve bu prosedür 1000 iterasyonla tekrarlanmıştır.

Deneyde iki ayrı durum incelenmiştir. İlk durumda, geri yayılım ağları tarafından üretilen eylem tavsiyesi ile sürücünün gerçekleştirdiği eylemin bire bir çakıştığı durum doğru kabul edilmiştir. Bu durumla ilgili grafik şekil 3.54'deki gibidir.

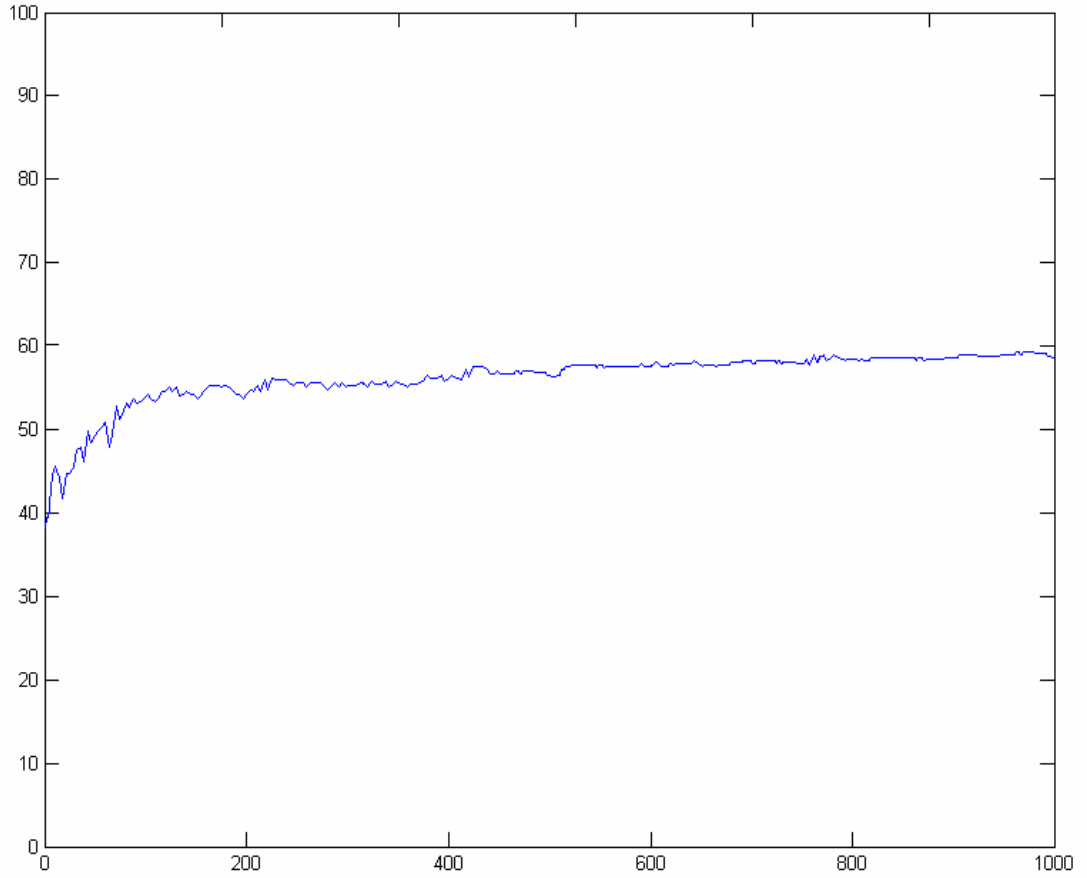


Şekil 3.54. Eylem tavsiyesi ile gerçekleşen eylem arasındaki doğruluk oranı

Şekil 3.52' ye göre ilk iterasyon sonunda model %27'lik bir doğruluk oranı sağlamıştır. 1000 iterasyon sonunda ise yaklaşık %44'lük bir doğruluk oranı sağlanmıştır. Doğru değerlere yakınsama eğiminin oldukça düşük olduğu, ya yakınsamanın çok uzun bir zaman alacağı veya hiç yakınsayamacağı da çıkarılabilecek sonuçlar arasındadır.

Deneyde ele alınan ikinci durum, doğruluk oranlarının kademesiz incelenmesidir. Bunun anlamı modelin dört temel eylemi doğru olarak tespit edip etmediğinin belirlenmesidir. Bu dört temel eylem hızlanma, yavaşlama, sağa ve sola direksiyon çevirme eylemleridir. Burada kademeler göz ardı edilmiştir. Yani, üretilen eylem tavsiyesi hızlan-3 ise ve gerçekleşen eylem hızlan-1 (bölüm 3.7.1) ise bu doğru olarak kabul edilmiştir. Bu durumla ilgili ortaya çıkan doğruluk oranları ise şekil 3.55'te gösterilmektedir.

Burada doğruluk oranının başlangıçta %39 seviyesinde iken 1000 iterasyon sonunda %59 civarında bir doğruluk oranına erişmektedir. Bu önceki duruma göre daha kötü bir tablo ortaya koymaktadır, yani bu sonuca göre 1000 iterasyon sonunda model halen hangi eylem tipini seçeceğine dahi karar verememektedir.



Şekil 3.55. Kademesiz inceleme.

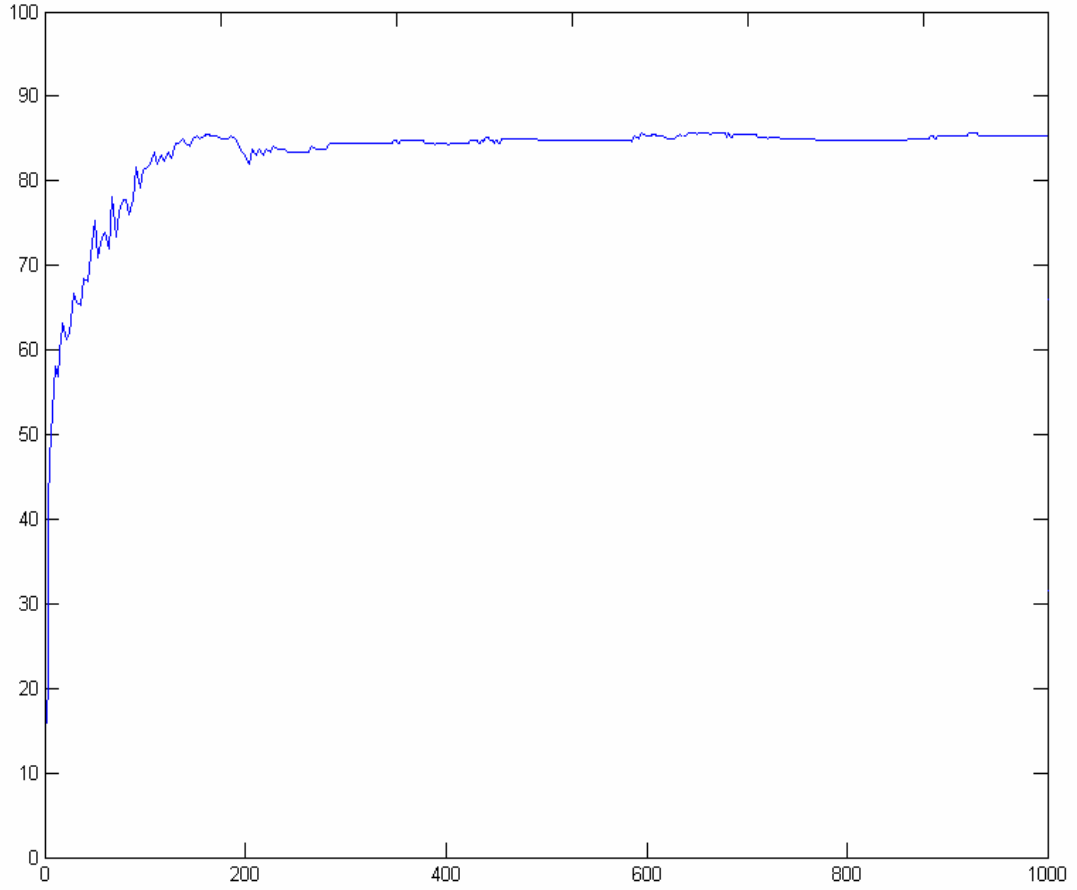
3.7.6.2. Deney II

Bu deneyde alt görev algoritması devreye alınarak, bu algoritmanın kullanımının sisteme olan katkısı ortaya çıkarılmaya çalışılmıştır. Alt görev algoritmasının kullanıldığı durumda artık problem tek bir problem olmaktan çıkıp, birbirine paralel alt problemler haline dönüşmektedir. Bu deneyde, alt görevin seçimi, takip mesafesi koruma alt görevi ve şerit değiştirme alt göreviyle ilgili geri yayılım ağları üzerinde çalışılmıştır.

I. Alt görevin seçimi

Sistemin girişleri her an için değerlendirilerek modelin anlık alt görevi tahmin etmesi istenmektedir. 3 adet farklı alt görev tanımı vardır. Sistem her an için bunların arasından bir seçim yapar ve daha sonra seçilen alt görevin alt

seviyedeki geri yayılım ağı çalıştırılarak görev devam ettirilir. Alt görev seçimi ile ilgili doğruluk oranları şekil 3.56'da görülebilir.

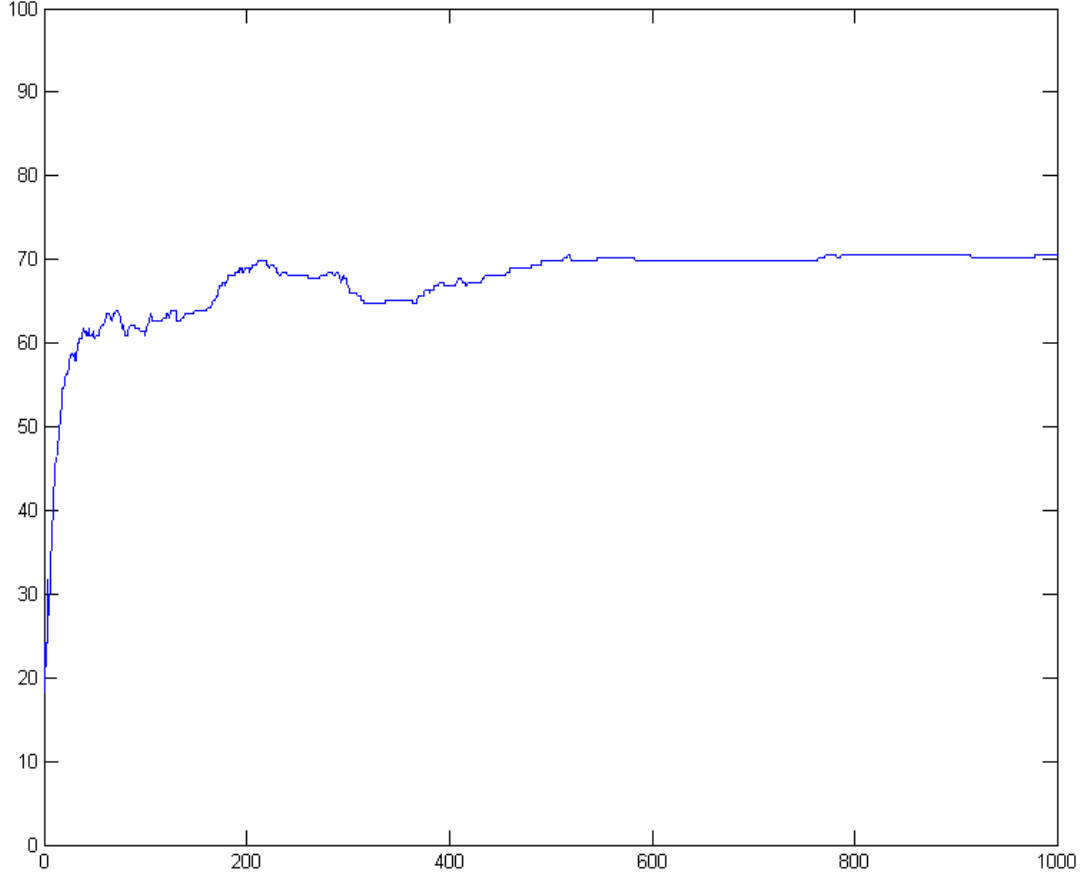


Şekil 3.56. Alt görev seçimi doğruluk oranları.

II. Takip Mesafesi Alt görevi

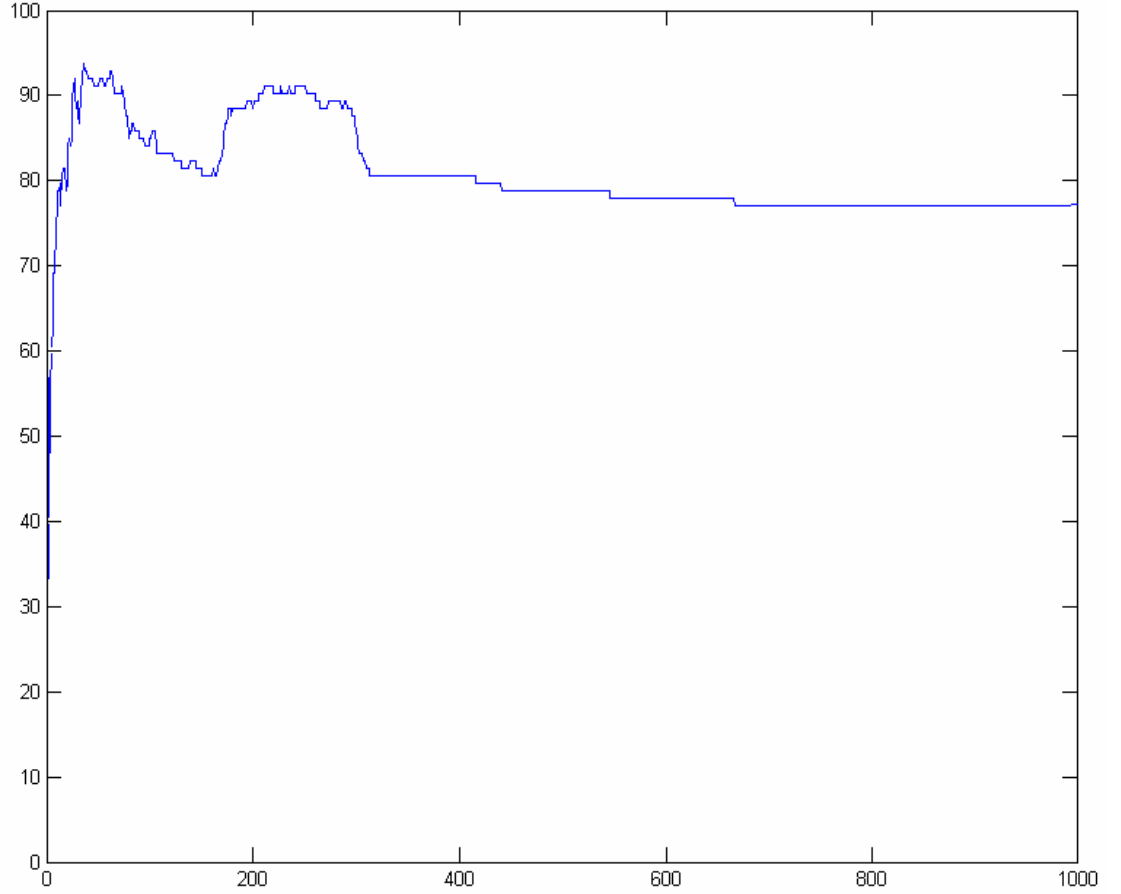
Takip mesafesi alt görevi, hedefi öndeki araçla belirli bir mesafeyi korumak olan, bu amaçla yalnızca hızı azaltıp arttıran, ve tek girdisi öndeki araçla arasındaki mesafe olan bir pekiştirmeli öğrenme problemidir. Yani bu problemin alt seviyesinde, giriş düğümü sayısı temelde 18, geçmiş bilgisiyle birlikte 124 olan, 124x124x1'lik 18 adet geri yayılım ağı mevcuttur. Burada gerçekleşen eylemler hızlan – 1, ..., -9, ve yavaşla – 1, ..., -9 şeklindedir. Ağ eğitime başlandıktan sonra üç farklı inceleme gerçekleştirilmiştir.

İlk incelemede ağın ürettiği eylem tavsiyesi ile gerçekleşen eylemin birebir çakıştığı durum incelenmiştir. Şekil 3.57’de bu incelemeyle ilgili doğruluk oranları görülmektedir.



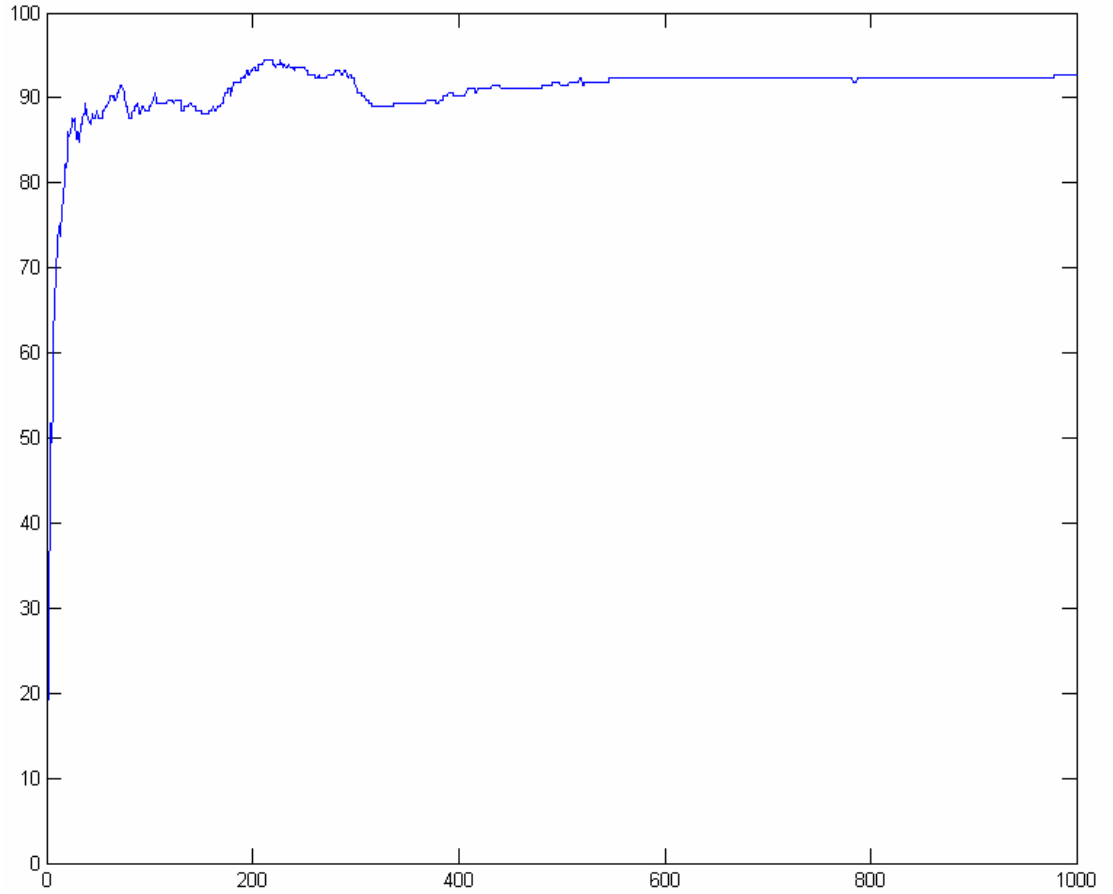
Şekil 3.57. Eylem tavsiyesi ile gerçekleşen eylemin birebir çakıştığı durum için doğruluk oranları.

İkinci incelemede gerçekleşen eylemle ± 1 fark kümesi içinde bulunan eylem tavsiyeleri doğru olarak kabul edilmiştir. Yani gerçekleşen eylem hızlan-2 ise, eylem tavsiyesinin hızlan-1, hızlan-2 veya hızlan-3 olduğu durumlar doğru olarak kabul edilmiştir. Bu durumda ortaya çıkan doğruluk oranları şekil 3.58’deki gibidir.



Şekil 3.58. Gerçekleşen eylemle ± 1 fark kümesi içinde bulunan eylem tavsiyelerinin doğru olarak kabul edildiği durumda doğruluk oranları.

Üçüncü inceleme kademesiz incelemedir. Yani eylem tavsiyesi hızlanmak ve gerçekleşen eylem hızlanmaksa, kademe önemli değildir ve sonuç doğru olarak kabul edilmiştir. Bununla ilgili doğruluk oranları şekil 3.59'daki gibidir.

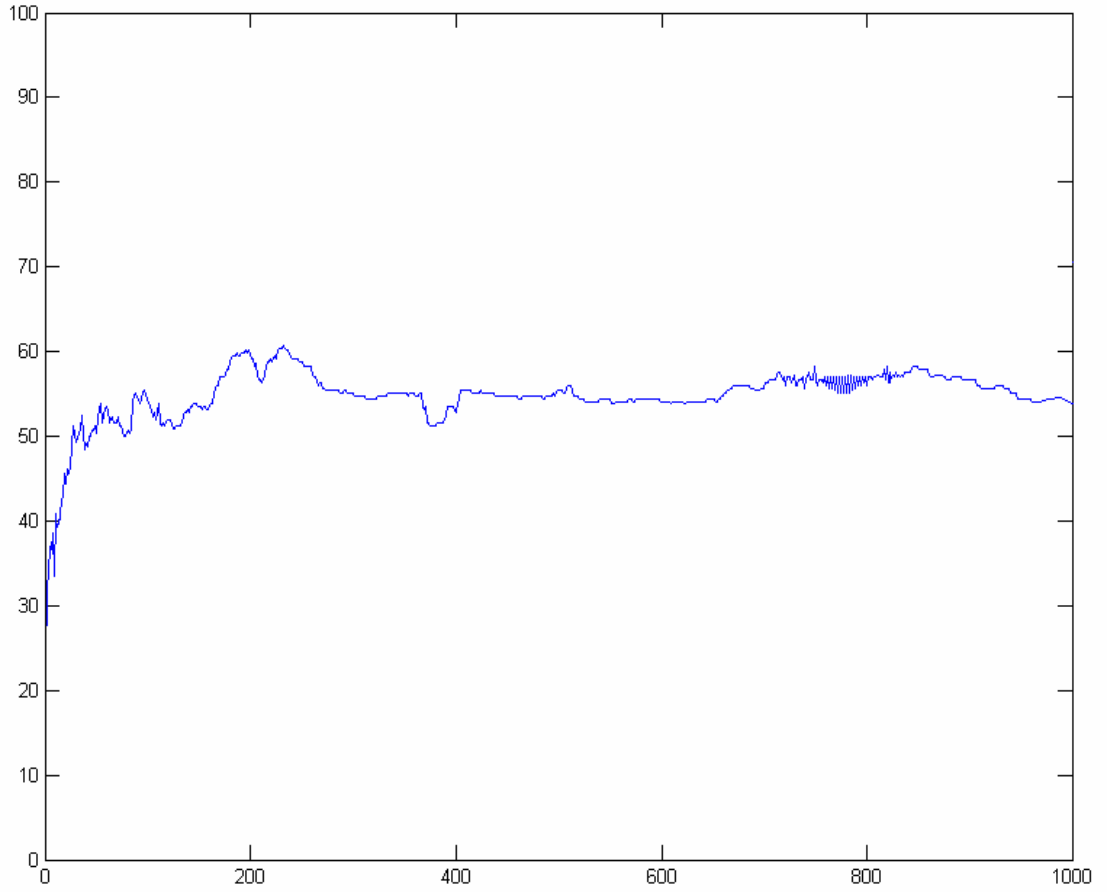


Şekil 59. Kademesiz inceleme.

III. Şerit Değişirme Alt görevi

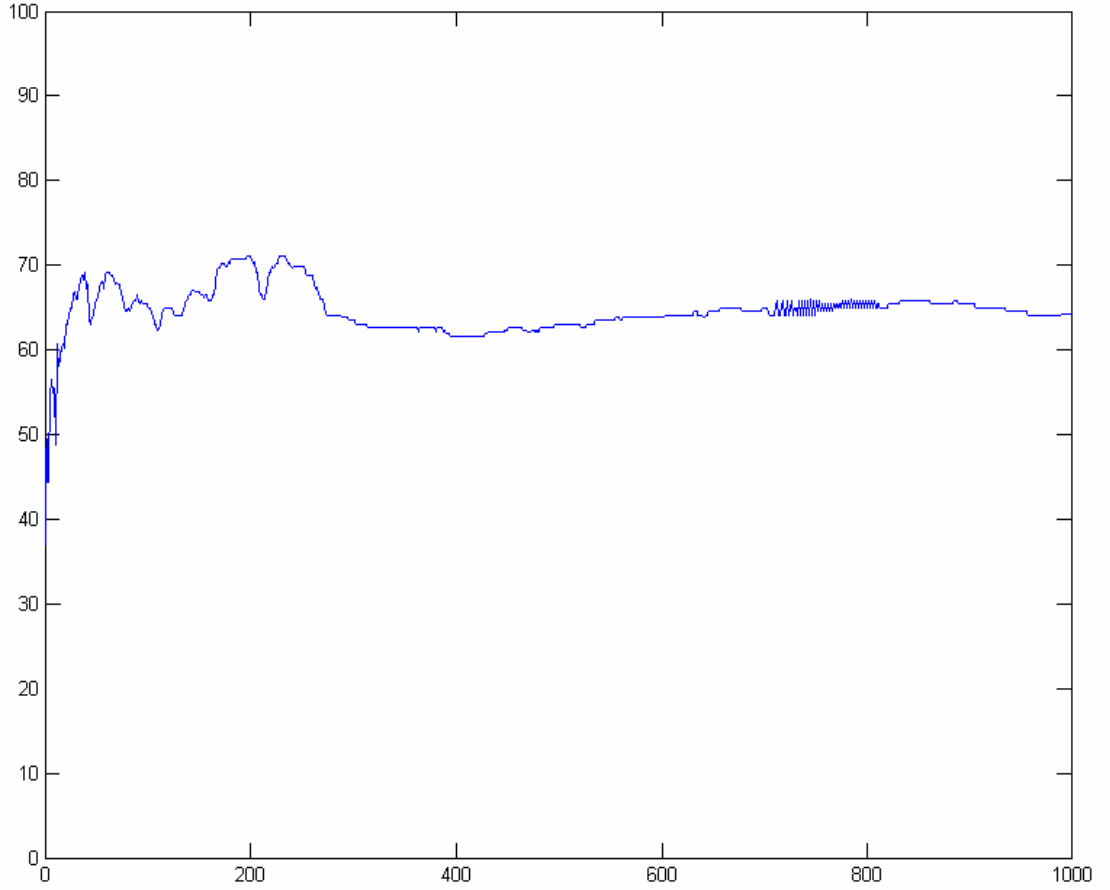
Şerit deęişirme alt görevi, hedefi saę veya sol şeride geęmek olan, bu amaçla yalnızca direksiyon aracılıęıyla aracın yönünü saęa veya sola doęru deęiştiren, ve girişleri aracın hızı, yol üzerindeki yönü, bulunduğu şerit ve şeride göre konum çıkışı ise yön eylemi olan pekiştirmeli öğrenme problemidir. Yani bu problemin alt seviyesinde, giriş düęümü sayısı temelde 35, geęmiş bilgisiyle birlikte 280 olan, 280x280x1'lik 10 adet geri yayılım aęı mevcuttur. Burada geręekleşen eylemler saę – 1, ..., -5, ve sol – 1, ..., -5 şeklindedir. Aę eğitilmeye başlandıktan sonra üç farklı inceleme geręekleştirilmiştir.

İlk incelemede aęın ürettięi eylem tavsiyesi ile geręekleşen eylemin birebir çakıştıęı durum incelenmiştir. Şekil 3.60'ta bu incelemeyle ilgili doęruluk oranları görölmektedir.



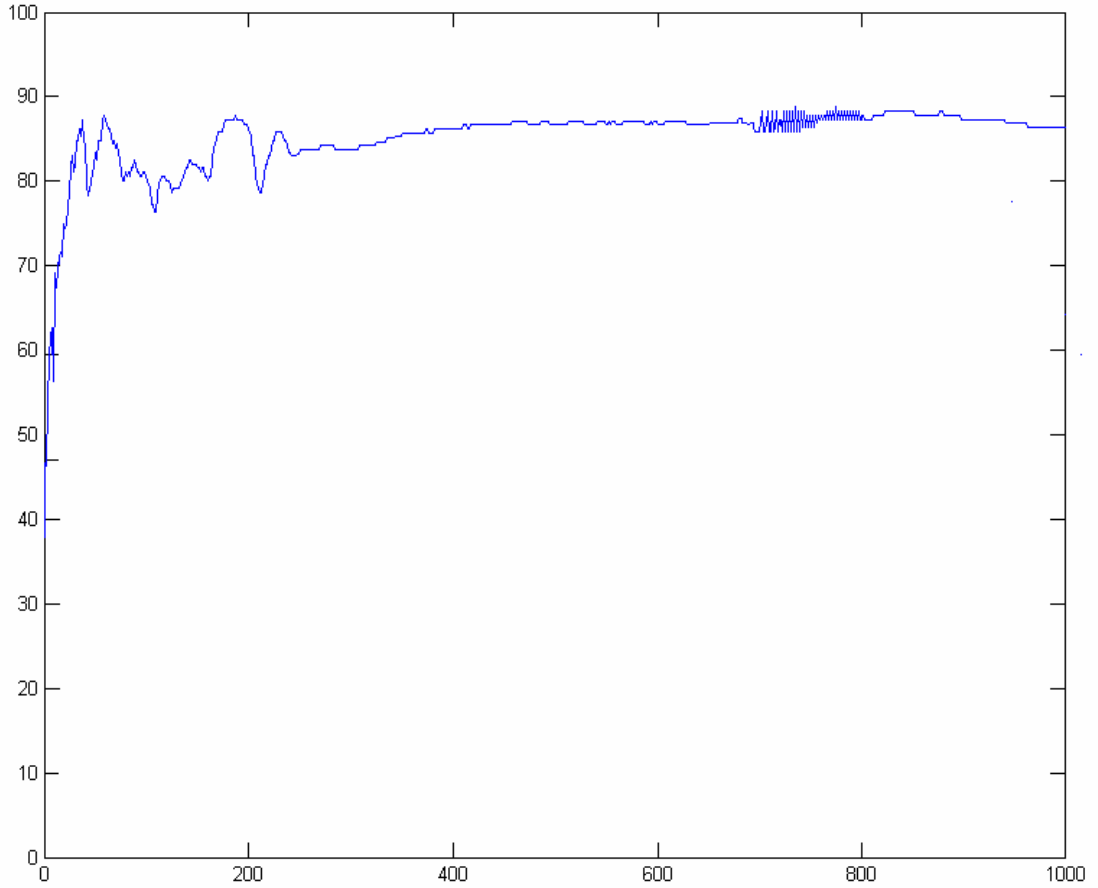
Şekil 3.60. Eylem tavsiyesi ile gerçekleşen eylemin birebir çakıştığı durum için doğruluk oranları.

İkinci incelemede gerçekleşen eylemle ± 1 fark kümesi içinde bulunan eylem tavsiyeleri doğru olarak kabul edilmiştir. Yani gerçekleşen eylem sağ-2 ise, eylem tavsiyesinin sağ-1, sağ-2 veya sağ-3 olduğu durumlar doğru olarak kabul edilmiştir. Bu durumda ortaya çıkan doğruluk oranları şekil 3.61'deki gibidir.



Şekil 3.61. Gerçekleşen eylemle ± 1 fark kümesi içinde bulunan eylem tavsiyelerinin doğru olarak kabul edildiği durumda doğruluk oranları.

Üçüncü inceleme kademesiz incelemedir. Yani eylem tavsiyesi hızlanmak ve gerçekleşen eylem hızlanmaksa, kademe önemli değildir ve sonuç doğru olarak kabul edilmiştir. Bununla ilgili doğruluk oranları şekil 62'deki gibidir.



Şekil 62. Kademesiz inceleme.

3.7.6.3. Deney Sonuçlarının Değerlendirilmesi

Şekil 3.54 ve şekil 3.55'e bakıldığı zaman, alt görev yapısı yerleştirilmeden, doğru değerlere yakınsamanın çok uzun süreceği görülebilmektedir. İşlem sayıları da göz önünde bulundurulursa, bu yöntemin çok uygun olmadığı sonucu çıkarılabilir.

Şekil 3.56'de alt görev seçimiyle ilgili doğruluk grafiği yer almaktadır. Burada gelen herhangi bir yeni durum için, geri yayılım ağının 3 alt görev içinden birisini seçmesi gerekmektedir. Doğruluk grafiğine bakıldığı zaman, 1000. iterasyonun sonunda geri yayılım ağı %85'in üzerinde bir doğrulukla herhangi bir an için doğru kararı verebilmektedir.

Şekil 3.57'de takip mesafesi alt görevinin doğruluk grafiği yer almaktadır. Burada birebir çakışma şartı aranmaktadır. Herhangi bir durum için

model 18 olası eylemden birisini seçecektir. 1000. iterasyon sonunda ulaşılan doğruluk derecesi %70'in üzerindedir.

Şekil 3.58'de modelin seçtiği eylemin, gerçek eylemin ± 1 kademe uzağında olması, doğruluk şartı olarak kabul edilmiştir. Bu durum için doğruluk grafiğine bakıldığı zaman, 1000. iterasyon sonunda doğru karar verme oranının %80'in biraz olduğu görülebilmektedir.

Şekil 3.59, model açısından oldukça umut verici bir doğruluk oranını göstermektedir. Model yapılan eylemi kademesiz olarak incelemekte, yani ne yapması gerektiğine kademesiz olarak karar vermektedir. Şekil 3.59'dan görülebileceği gibi model yaklaşık %93 oranında bir doğrulukla, hangi anda hızlanması, hangi anda yavaşlaması gerektiğine karar verebilmektedir.

Şekil 3.60, Şekil 3.61 ve Şekil 3.62, şerit değiştirme alt görevi için ortaya çıkarılan doğruluk oranlarıdır. 1000. iterasyon sonunda birebir çakışma oranı %55 civarında kalırken, ± 1 kademe komşuluğu doğruluk oranı %64 civarında görülmektedir. Kademesiz inceleme ise yine umut vericidir, çünkü herhangi bir anda, modelin sağa mı yoksa sola mı dönülmesi gerektiğine dair verdiği kararların doğruluk oranı 1000. iterasyon sonunda yaklaşık %87 civarındadır.

Araç sürme gibi bir problemin çözümü neredeyse hatasız çalışmayı şart koşturmaktadır. Verilecek hatalı kararlar araç sürme ortamında telafisi mümkün olmayabilecek hasarlara yol açabilir. Bu açıdan bakılırsa, kurulan model uygulanabilir olmaya adaydır fakat geliştirilmeye ihtiyacı vardır.

Modelin şu an için yetersiz görünmesinde birkaç faktör etkili olabilir. Modelin verdiği kararların hatalı olmasındaki en büyük faktör, düşüncemize göre uzman sürücülerden veri toplama prosedürünün çok da ideal şartlar altında gerçekleştirilememiş olmasıdır. Bilgilerin büyük kısmı el yordamıyla düzenlenmiştir. Bu kısım hataya çok açıktır. Tabii ki anlık durumun ve buna karşılık gerçekleşen eylemin hatalı ifadesi, modelin eğitimi konusunda büyük sorunlar oluşturabilir. Bunun yanında, durum ifadesi mükemmelere yakın olsa dahi, bu sefer de, örnek sayısının çok daha fazla olması gerektiği de söylenebilir. Örnek sayısının sonsuza doğru gitmesiyle birlikte, Q-

değerlerimizin, bunun yanında pekiştirme fonksiyonumuzun doğru değerlerine daha da yakınsayacağı bir gerçektir. Bunun yanı sıra, ele alınan durum ifadesinin durum değişkenleri açısından daha da zenginleştirilmesi, ayırıklaştırmanın daha büyük boyutlarda yapılması, modelin doğru bir modele doğru yakınsamasına yol açacak faktörlerdendir.

4. ARAŞTIRMA SONUÇLARI ve TARTIŞMA

Bu tez çalışmasında, araç sürücü davranışını modellemek için bilişsel mimari temelli bir model önerilmiştir. Bilişsel mimari insan davranışını modelleme konusunda gelecek vaat eden bir konudur ve geniş bir yelpazede kendisine uygulama alanı bulmuştur. Fakat araç sürücü davranışını modelleme konusunda bilişsel mimari tabanlı uygulamalar konusunda oldukça kısıtlı bir literatür bilgisi mevcuttur. Literatürde bulunan çalışmalarda ise araç sürme bütünsel bir problem şeklinde ele alınmamış, araç sürme probleminin parçaları incelenerek modellenmeye çalışılmıştır. Örneğin Aasman araç sürme probleminin bir parçasını oluşturan kavşağa girme problemini ele almış, Salvucci ve arkadaşları ise şerit içinde güvenli bir seyir probleminin modellenmesi üzerine çalışmışlardır.

Bu tez çalışmasında önerilen mimarinin, araç sürme problemini bütünsel olarak modellemesi hedeflenmiştir. Mimari, bilgi tipine dayalı alt ve üst seviyede birbirine paralel olarak çalışan modüllerden oluşmaktadır. Alt seviye eylem seçme ve öğrenme işlemi geri yayılım ağlarıyla, üst seviye eylem seçme ve öğrenme işlemi kural tabanlı yapılarla gerçekleştirilmiştir. Anlık durum için alt seviyede geri yayılım ağları, üst seviyede kural tabanlı sistemler, bir eylem çıkışı üretirler. İki seviyeden gelen çıkışlar arasından bir seçim gerçekleştirilerek, eylem uygulanır. Öğrenme işlemi ise eylem uygulandıktan sonra başlar, eylemin uygulanmasıyla oluşan yeni durum ve yeni duruma ortamın verdiği tepki, öğrenmenin anahtarını oluşturur. Ortamın verdiği tepki, Q-Öğrenme mekanizmasının pekiştirmesini oluşturur ve anlık duruma bu pekiştirmeyle birlikte bir Q-değeri atanır. Yeni atanan Q-değeri ile durumun eski Q-değeri arasındaki fark mevcut geri yayılım ağlarının geri yayılım hatası olarak kabul edilir ve bu hata doğrultusunda ağlar eğitilir. Bu yeni ve eski Q-değerleri arasındaki fark üst seviyede de kural seçimi, kural genelleştirme ve kural özelleştirme gibi işlemlerin gerçekleştirilmesinde kullanılır ve üst seviyede bu bilgi aracılığıyla kendini günceller.

Q-Öğrenme hem alt hem de üst seviye için kullanılan en önemli öğrenme mekanizmasıdır. Aslında alttan üste öğrenme yöntemi seçildiği için tek

öğrenme yöntemidir de diyebiliriz. Bunun sebebi taklit öğrenme, bağımsız kural öğrenme, öğreticili öğrenme veya öğreticisiz öğrenme gibi diğer yöntemlerin kullanılması da mümkündür fakat bu mimari dahilinde bu yöntemlerin tümü öğrenme işleminde Q-değerlerini kullanmaktadır. Bu yüzden geri yayılım ağları ve Q-Öğrenmenin doğru bir şekilde kurulup çalıştırılması bu tez çalışması açısından çok önemli bir konuyu oluşturmaktadır.

Bu tez çalışmasının katkısını iki ana başlık altında inceleyebiliriz, birincisi sürücü davranışını modelleme konusunda bilişsel mimari tabanlı yöntemler yeni geliştirilmektedir ve şu anda bu problemi sağlıklı bir şekilde çözebilen bir yapı kurulabilmiş değildir. Bu tez çalışmasında bu konuda sağlıklı bir yapı oluşturabilmek için bir model önerilmektedir. İkincisi, bu tez çalışmasında insan davranışını modellemede, mimarinin çekirdeğini teşkil eden Q-Öğrenmenin yaşadığı bazı problemlere çözüm önerileri getirilmektedir. Q-öğrenme tekniklerinde bu problemlerinin kaynağını teşkil eden standart Q-Öğrenmenin varsayımlarıdır. Q-Öğrenmede en önemli hedef, durumlara atanan Q-değerlerinin olabilecek en doğru değerlerine doğru yakınsamasıdır. Ancak en doğru değerlerine yakınsadığı zaman Q-Öğrenme doğru davranış biçimleri geliştirebilir. Aksi takdirde oluşan davranış biçimi anlık durum için doğru karar vermeyi garanti etmemektedir.

Öncelikle, standart Q-Öğrenme yöntemi doğru davranış biçimine yakınsamak için ele aldığı problem uzayının sonlu veya belirli oranda büyük olmasını şart koşar, çok büyük veya sonsuz durum uzayının varlığı durumunda yakınsamayı garanti etmemektedir. Aslında durum uzayı ne kadar büyürse, standart Q-öğrenmenin doğru davranışa doğru yakınsama oranı o kadar düşmektedir. Fakat insan davranışı gibi sürekliliğin olduğu, dolayısıyla durum ifadelerinin çok zengin olduğu bir koşulda nasıl bir yapı kurulabilir. Araç sürme davranışının algılayıcı kümesi oldukça kısıtlı olmakla birlikte, algılayıcılardan akan bilgilerin ve oluşturulan eylemlerin sürekli olmasından dolayı durum uzayı oldukça büyümektedir. Büyük durum uzayı probleminin çözümü için ortaya atılan yaklaşımlardan biri durum uzayının parçalanarak problemin bir çok küçük pekiştirmeli öğrenme problemine dönüştürülmesi ve her birinin birbirine paralel olarak çalıştırılmasıdır. Literatürde bunu gerçekleştirebilmek için önerilmiş pek

çok yöntem vardır ve bu yöntemler genel olarak hiyerarşik bölümlenme, geçici soyutlama ve alt görevlere ayırma gibi temel başlıklar altında toplanmıştır. Bu yöntemler kullanılarak Q-öğrenmenin performansının önemli oranda artırıldığı, literatürde varolan çeşitli yayınlarda çeşitli deneylerle gösterilmiştir. Önerilen yöntemlerin en önemli ve en çok kabul gören yaklaşımlarından birisi seçenekler yapısıdır.

Seçenekler yapısı araç kullanma problemine uygulandığı zaman, araç kullanma problemi, sol şeride geçme, sağ şeride geçme, maksimum hıza çıkma veya takip mesafesini koruma benzeri alt problemlere bölünecektir. Bu alt problemlerin her biri kendi bölünmüş durum uzayına sahiptir ve o uzay yöntemin çalışma alanını oluşturmaktadır. Seçenekler yapısına göre bu alt görevleri birbirine bağlayan geçiş noktaları vardır, ve bu noktalar belirlenirse durum uzayı bu alt görevlerin bir bileşimi olarak yeniden oluşturulabilir, yani bunun anlamı durum uzayının eksiksiz şekilde alt görevlere bölünmesidir. Seçenekler yapısının en büyük eksikliği ise bu geçiş noktalarının otomatik olarak belirlenememesidir. Geçiş noktalarının belirlenmesiyle ilgili literatürde çalışmalar mevcuttur fakat bu çalışmaların tamamı gridworld ortamı gibi son derece basit bir domeni kendilerine problem uzayı olarak seçmişlerdir. Bu çalışmalarda bize yön vermesi açısından en büyük eksiklik, aynı yöntemlerin insan davranışı gibi çok daha karmaşık bir ortama uygulanırsa nasıl performans gösterebileceği ile ilgili bilginin mevcut olmamasıdır.

Bu tez çalışmasında gerçekleştirilen en önemli katkı, insan davranışı gibi oldukça karmaşık bir yapıda geçiş noktalarının otomatik olarak belirlenmesinin sağlanmasıdır. Alt görev geçiş noktalarının belirlenmesinde son derece doğal bir yolun kullanılmasını önerdik. Bu yöntemin oluşturulmasındaki temel düşünce, seyir esnasında araç sürücüsünün bulunduğu durumu her zaman adımında değiştirmesinin mantıklı olmadığı, hiçbir sürücünün durduk yerde içinde bulunduğu durumu değiştirmeyeceği varsayımı üzerine kurulmuştur. Yani hiçbir sürücü, ortada bir sebep yokken, yani önünde arkasında bir araç yokken hızını her adımda düşürmez veya arttırmaz, veya hiçbir sebep yokken her zaman adımında aracın yönünü değiştirmez. Genelde sürücü, belirli bir duruma gelir ve bu durumda, durumunu değiştirmesini

gerektirecek bir sebep oluşana kadar devam eder. Örneğin, önüne hiçbir araç geçmediği takdirde veya buna benzer başka bir sebep oluşmadığı takdirde sürücü belirli bir hızla ve şeridini de değiştirmeye gerek duymaksızın yoluna devam eder. Ancak önüne bir araç çıktığı takdirde iki seçenek arasından seçim yapmalıdır. Ya en kötü ihtimalle güvenli takip mesafesini koruyacak kadar hızını düşürecektir veya şeridini değiştirerek önündeki aracı geçecektir. Psikoloji literatüründe, insanın belirli bir sebep olmadan değiştirmedığı durum dizilerine kararlı durum dizileri adı verilmektedir. Belirli bir pekiştirme oluşmadığı takdirde insan kararlı durumunu koruyacaktır. Kararlı durumların bu çalışmadaki tanımı sürücünün üzerinde ısrarla durduğu durumlar diğer bir deyişle ısrarcı durumlardır. Bir pekiştirme oluştuğu zaman ise önüne gelen seçeneklerden birisini seçerek yoluna devam edecektir oluşmadığı durumda ise bulunduğu durumda ısrar edecektir. Bu çalışmada önerilen yöntemde, bu ısrarcı durumlar alt görevler olarak kabul edilmiş ve bu ısrarcı durumların ortaya çıkarılması için bir sürücünden gözlenen durum dizileri kullanılmıştır. Durum dizileri bir zaman serisi olarak ele alınmış ve zaman serisinde bulunan ısrarcı durumların belirlenmesinde, veri madenciliği literatüründe mevcut tekniklerden birisi olan zaman serileri içinde ısrarcı aralıkların belirlenmesi yöntemi kullanılmıştır. Bu teknikle ısrarcı aralıklar ortaya çıkarılmış ve bu ısrarcı aralıklar, yöntemimizde aynı zamanda alt görevlerin geçiş noktaları olarak kabul edilmiştir. Algoritmanın çalıştırılmasıyla ortaya çıkan sonuçların, literatürde kabul edilen araç sürme alt görevleriyle uyduğu görülmüştür.

Yine bir varsayım olarak standart Q-öğrenme yöntemleri, problem domeninin Markov ortamı olduğunu varsaymaktadır. Markov özelliği, anlık durumun geçmişle ilgili tüm bilgileri taşıdığını kabul eder. Fakat araç sürme olayı oldukça dinamik bir ortamda gerçekleşmektedir ve bu ortamda elde edilen anlık durumun geçmişle ilgili doğru bilgiyi taşıdığı garanti edilemez. Örneğin sürücüye göre sol şeritte ve arkada olan bir araç olduğunu ve sürücünün de o araçla aynı şeride geçmek için karar vermek durumunda olduğunu düşünelim. İlgili aracın anlık pozisyonu sürücünün karar vermesinde yeterli olacak mıdır? Olmayacaktır çünkü araç o şeritte hızla ilerlemekte de olabilir, bir süredir aynı hızda devam ediyor da olabilir veya sürücü bir süre önce o aracı geçmişte

olabilir. Buradan çıkarılacak sonuç, arkada bulunan aracın anlık durum bilgisinin aslında o aracın geçmişiyile ilgili bilgiyi taşımadığıdır. Bu da araç sürme ortamının Markov özelliğine sahip olmadığını göstermektedir. Bu probleme literatürde algısal örtüşme problemi adı verilmektedir ve yöntemin buna benzer Markov özelliği taşımayan ortamlarda da çalışabilmesi için pek çok çözüm önerisi mevcuttur. En basit yöntem, geçmiş bilgisinin durum bilgisi ifadesine dahil edilmesidir. Fakat yarı-Markov seçeneklerinin kullanımı da bu sorunun çözümünü sağlamaktadır. Bu çalışmada geçmiş bilgisinin durum bilgisi dahilinde değerlendirilmesi için bir geçici bellek kullanılmış ve geçmiş bilgisi de anlık durumla birlikte sistemin girişi olarak değerlendirmeye tabi tutulmuştur.

Standart Q-Öğrenme tekniklerinin yakınsamasının garanti edilebilmesi için bir üçüncü varsayım, Q-Öğrenme fonksiyonun en uygun değerlere doğru yakınsamasında en önemli fonksiyonlardan birisi olan pekiştirme fonksiyonunun doğru oluşturulmasıdır. Fakat insan davranışı gibi karmaşık bir yapıda insana ait pekiştirme fonksiyonunu ortaya çıkarmak o kadar kolay değildir. Durumlarla pekiştirmeler arasında kolayca ifade edilebilecek bir ilişki olmadığı gibi, varolan ilişki de insandan insana değişmektedir. O halde pekiştirme fonksiyonu önceden belirleyip sisteme vermektense, sistemin sürücüyü gözlemleyerek sürücünün pekiştirme fonksiyonunu elde etmesi makul bir çözüm gibi görünmektedir. Yapılan literatür taramasında bu konuyla ilgili de çözüm önerileri incelenmiş ve doğru pekiştirme fonksiyonunu elde etme amacıyla bir geri pekiştirmeli öğrenme algoritması modele dahil edilmiştir. Q-öğrenme yöntemi elde edilen bu pekiştirme fonksiyonu aracılığıyla çalıştırılmıştır.

Modele bir bütün halinde bakıldığı zaman eğitimin iki şekilde gerçekleştirildiği görülmektedir. Aslında bu yöntem insan sürücülerin eğitimini kendisine temel olarak almaktadır. İnsan sürücü araç kullanmayı bilir halde doğmaz, ancak gözlem yeteneği sayesinde araç kullanma hakkında zaman içinde bilgi sahibi olur. Modelimizin birinci aşamasını gözleyerek öğrenme yöntemi oluşturmaktadır. Bu yöneme göre, bilişsel eleman, bir insan sürücüyü gözleyerek hangi durumlarda hangi kararları ürettiğini inceleyecek, ve bu incelemeden sonra kendini bu verilerle eğitecektir. Her adım sonunda bilişsel elemandan bir karar vermesi beklenecek, daha sonra verdiği karar insan sürücü

ile karşılaştırılarak verilen doğru karar oranı hesaplanacaktır. Belirli bir eğitim sonunda doğru karar oranının belirli bir eşik seviyesini geçmesiyle eğitimin birinci safhası tamamlanmış olacaktır. Bu durumda bilişsel eleman, trafiğe yeni çıkmış acemi bir sürücü konumunda olacaktır. Eğitimin ikinci kısmı ise bizzat işi gerçekleştirerek devam edecektir. Bilişsel eleman kendi kararlarını kendisi verecek, verdiği kararlar doğrultusunda ortamdaki pekiştirmeler alacaktır. Q-değerlerinin güncellenmesi, oluşan yeni durumlar, yeni pekiştirmeler ve yeni eylemler zinciriyle devam edecektir. Bilişsel elemandan beklenen her gün deneyimler yoluyla bir önceki güne göre daha iyi bir performans göstermesidir.

Gelecek Çalışma

Bilişsel mimari temelli insan davranışı modelleme konusunda daha yapılacak çok fazla iş vardır ve yapılan literatür taramaları, bu konunun ilerleyen yıllarda çok yoğun çalışmalara sahne olacağını düşündürmektedir. Araç sürme problemi, insan davranışını modelleme ile ilgili problemlerden yalnızca biridir ancak sahip olduğu özellikler araç sürme problemini insan davranışı modelleme alanında yapay zeka teknikleri açısından çok çekici kılmaktadır.

Bu tez çalışmasında önerilen model, daha yolun başını temsil etmektedir. Bu tez çalışmasında ele alınan bilişsel mimarinin dört alt merkezinden yalnızca biri olan eylem merkezli alt sistemdir. Diğer alt merkezlerle ilgili yapılacak çalışmalar ve ondan da önemlisi bu alt merkezler arasındaki etkileşim, gelecek aşısından en önemli çalışma konularından birini teşkil edecektir.

Geri yayılım ağları önerdiğimiz yöntemde kullanılabilecek tek çözüm olarak görülmemelidir. Geri yayılım ağlarına alternatif olarak diğer yöntemler de denenebilir ve sisteme katkısı ortaya çıkarılabilir. Bu konuda yapılabilecek çalışmalar, bilişsel elemanın öğrenme hızına katkıda bulunabilir.

'Gerçek' hayatın sürekli doğasından dolayı, sonsuz sayıda anlamlı durum söz konusudur. Yalnızca durumlar değil eylemler de sürekli ve onları basitçe ayırklaştırmak yeterli değildir. Bir şeritte devam ederken, dönüş açılarının herhangi bir ayrık seti, şeridin merkezinde ilerlemeye çalışırken salınımına sebep olabilir. Bu durum bizi ya sürekli durumu kabul etmeye veya

ayrık kümemizi çok genişletmeye zorlar. Ayrık kümeyi genişletmek, durum uzayını da büyütme demektir. Durum uzayını büyütme performansı düşürecektir. Buradaki alışverişi ya en uygun şekilde gerçekleştirecek bir yöntemin, veya bu konuda yeni çözümlerin oluşturulabilmesi gelecek sistemleri açısından önemlidir.

Önerilen ısrarcılık aralıklarını belirleme tekniği, kararlı aralıkları belirlemektedir. Fakat şu an için hangi alt görevlerin hangi algılayıcı kümeleriyle çalışacağı, yani alt görevlere ait durum uzaylarının belirlenmesi konusu açık bırakılmıştır. Bu belirlemeleri gerçekleştirebilecek tekniklerin belirlenmesiyle yöntemin çok daha etkili bir yöntem haline geleceğini düşünmekteyiz.

Önerdiğimiz modelde, pekiştirme fonksiyonunun ve alt görevlerin belirlenmesi çevrimdışı bir şekilde gerçekleştirilmektedir. Yani ağırlar eğitilmeden önce, diziler pekiştirme fonksiyonu ve alt görevlerin belirlenmesinde kullanılmaktadır. Bu yöntemin çevrimiçi hale gelmesi için yapılacak çalışmalar, modelin geleceği açısından önemlidir.

Yapılan deneysel çalışmada algılayıcı olarak dört adet kamera kullanılmış ve bu kameralardan gelen bilgiler durum ifadesi şekline dönüştürülmüştür. Burada iki problem karşımıza çıkmaktadır, birinci problem veri toplama sıklığının yüksek kalması, ikinci problem ise kamera görüntülerinin durum ifadelerine dönüştürülmesi sürecinin hataya çok açık olmasıdır. Bu şartlarda dahi oluşturulan model, öğrenme işlemini gerçekleştirmede belirlenen eşiğe göre başarılı sayılabilir. Ancak daha yüksek performans sağlayacak deney setlerinin oluşturulmasıyla, önerilen modelin çok daha yüksek performans göstereceğini düşünmekteyiz.

5. KAYNAKLAR

- ABBEEL, P. NG, A.Y. 2004. Apprenticeship learning via inverse reinforcement learning. ACM International Conference Proceeding Series, Proceedings of the twenty-first international conference on Machine, Banff, Alberta, Canada 69 : 1-9.
- AN, P. E. ve HARRIS, C.J. 1996. An Intelligent Driver warning System for Vehicle Collision Avoidance. IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans, 26 (2) : 254-261.
- ANDERSON, J.R. 1983. The Architecture of Cognition. Harvard University Press, Cambridge, MA, 345 s.
- AMDERSON, J.R. 1993. Rules of the Mind. Lawrence Erlbaum Associates, Hillsdale, NJ, 320 s.
- ANDERSON J.R. ve LEBIERE, C. 1998. The atomic components of Thought. Lawrence Erlbaum Associates, Mahwah, NJ, 504 s.
- ANDERSON, C.W. 1987. Strategy learning with multilayer connectionist representations. In: Proceedings of the Fourth International Workshop on Machine Learning, p. 103-114.
- BARTO, A. SUTTON, R. ve ANDERSON, C. 1983. Neuron-like adaptive elements that can solve difficult learning control problems. IEEE Transactions on Systems, Man and Cybernetics, 13 : 834-846.
- BARTO, H. ve MAHADEVAN, S. 2003. Recent Advances in hierachical reinforcement learning. Discrete Event Systems Journal, p. 41-77.
- BAXTER, J. ve BARTLETT, P.L. 2001. Infinite horizon policy gradient estimation. Journal of Artificial Intelligence Research, 15 : 319-350.
- BEHNKE, S. ve BENNEWITZ, M. 2005. Learning to Play Soccer using Imitative Reinforcement. Proceedings of ICRA 2005 Workshop on Social Aspects of Robot Programming through Demonstration, Barcelona, Spain.

- BERRY, D. ve BROADBENT, D. 1984. On the relationship between task performance and associated verbalizable knowledge. *Quarterly Journal of Experimental Psychology*, 79 : 251-272.
- BERTOZZI, M. BROGGI, A. ve FASCIOLI, A. 2000. Vision-based intelligent vehicles: state of the art and perspectives. *Journal of robotics and autonomous systems*. 32 (1) : 1-16.
- BERTSEKAS, D.D. ve TSITSIKLIS, J. N. 1996. *Neuro-Dynamic Programming*. Nashua, Athena Scientific, 491 s.
- BISHOP, C.M. 1997. Vehicle-Highway automation activities in the United States. In *International AHS Workshop*. US Department of Transportation.
- BLACKBURN, M.R. ve NGUYEN, H.G. 1994. Autonomous visual control of a mobile robot. In *ARPA94*, 2 : 1143-1150.
- BOER, E.R. HILDRETH, E.C. ve GOODRICH, M.A. 1998. A driver model of attention management and task scheduling: Satisficing decision making with dynamic mental models. *Proc. 17th Euro. Annu. Conf. Human Decision Making Manual Control*, Valenciennes, France, 14–16 December 1998, 325-336.
- BRACHMAN, R.J. 2002. Systems that know what they are doing. *Intelligent Systems*, 17 (6) : 67-71.
- BROOKS, R.A. 1991. New Approaches to robotics. *Science*, 253 : 1227-1232.
- CHAPMAN, D. ve KAELBLING, L.P. 1991. Input generalization in delayed reinforcement learning: An algorithm and performance comparisons. *Proceedings of IJCAI-91*, p. 726-731.
- CLARK, A. ve KARMILOFF-SMITH, A. 1993. The cognizer's innards. A psychological and philosophical perspective on the development of thought. *Mind and Language*, 8 (4) : 487-519.
- CLEERAMANS, A. DESTREBECQZ, A. ve BOYER, M. 1998. Implicit Learning: Views from the front. *Trends in cognitive Sciences*, 2 (10) : 406-416.

COWARD, L. A. and SUN, R. 2004. Criteria for an effective theory of consciousness and some preliminary attempts. *Consciousness and Cognition*, 13 : 268-301.

DAYAN, D. ve HINTON, G.E. 1993. Feudal Reinforcement learning. *Advances in Neural Information Processing Systems*, 5 : 271-278.

DIETTERICH, T.G. 2000. Hierarchical reinforcement learning with MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13 : 227-303.

DIGNEY, B. 1998. Learning hierarchical control structure for multiple tasks and changing environments. In R. Pfeifer, B. Blumberg, J. Meyer, & S. Wilson (Eds.), *From Animals to Animats 5: Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior*, MIT Press, p. 321-330.

DREYFUS, H. ve DREYFUS, S. 1987. *Mind over Machine. The Power of Human Intuition*. The Free Press, New York, 252 s.

FENTON, R.E. 1994. IVHS/AHS: Driving into the future. *IEEE Control Systems Magazine*, 14 (6) : 13-20.

FLETCHER, L. PETERSON L. ve ZELINSKY, A. 2003. Driver Assistance Systems based on vision in and out of Vehicles, *IEEE Intelligent Vehicles Symposium*, 9-11 June 2003 p. 322 – 327.

FORBES, J.R.N. 2002. Reinforcement Learning for Autonomous vehicles. Ph.D. Thesis (unpublished), University of California at Berkeley, 110 p.

GALLET, A. SPIGAI, M. HAMIDI, M. ve SEGIME, S.A. 2000. Use of vehicle navigation in driver assistance systems. *Intelligent Vehicles Symposium, 2000. IV. IEEE Proceedings of the IEEE*, p. 492-497.

GREFENSTETTE, J.J. RAMSEY ve C.L. SCHULTZ, A.C. 1990. Learning sequential decision rules using simulation models and competition. *Machine Learning*, 5 : 355-382.

HADLEY, R. 1995. The explicit-implicit distinction. *Minds and Machines*. 5 : 219-242.

- KIRSH, D. 1992. When is Information Explicitly Represented? The Vancouver Studies in Cognitive Science, Re-issued Oxford University Press, p. 340-365.
- KRETCHMAR, M. FEIL, T. ve BANSAL, R. 2003. Improved Automatic Discovery of Subgoals for Options in Hierarchical Reinforcement Learning. Journal of Computer Science and Technology, p. 9-14.
- KRUGER, W. ENKELMAN, W. ve ROSSLE, S. 1995. Real Time estimation and tracking of optical flow vectors for obstacle detection. In IEEE Intelligent Vehicles Symposium, Detroit, MI, p. 341-346.
- KULLBACK, S. ve LEIBLER, R.A. 1951. On information and sufficiency, Annals of Mathematical Statistics 22 : 79-86.
- LEE, J. W. 2002. A Machine-vision System for Lane Departure Detection. Computer Vision and Image understanding, 86 : 52-78.
- LIN, L.J. 1991a. Self-improving reactive agents: Case studies of reinforcement learning frameworks. Proceedings of the First International Conference on Simulation of Adaptive Behaviour: From Animals to Animats, p. 297-305.
- LIN, L.J. 1991b. Self-improvement based on reinforcement learning and teaching. Proceedings of the Eighth International Workshop on Machine Learning, p. 323-327.
- LIN, L.J. 1991c. Programming robots using reinforcement learning and teaching. Proceedings of AAAI-91, p. 781-786.
- LIN, L.J. MITCHELL, T.M. 1992. Memory Approaches to Reinforcement Learning in Non-Markovian domains. Technical Report CMU-CS-92-138, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.
- LIN, L.J. 1992. Self-improving reactive agents based on reinforcement learning, planning and teaching, Machine Learning, Kluwer Academic Publishers, Boston. Manufactured in the Netherlands, 8 : 293-321.
- LUCE, R.D. 2000. The Utility of Gains and Losses. Lawrence Erlbaum Associates, Mahwah, NJ, 336 s.

LUCE, R. D. 2005. Individual Choice Behavior: A Theoretical Analysis. Wiley, New York, 176 s.

LÜTZELER, M. ve DICKMANN, E.D. 1998. Road recognition in marveye. In IEEE Intelligent Vehicle Symposium, Stuttgart, Germany, p. 341-346.

LYGEROS, J. GODBOLE D.N. ve SASTRY S.S. 1997. A verified hybrid controller for automated vehicles. Technical report UCB-ITS-PRR-97-9, California PATH.

MACE, C. F. 1996. In Pursuit of general behavioral relations. Journal of Applied Behavior Analysis, 29 : 557-563

MAHADEVAN, S. ve CONNELL, J. 1991. Scaling reinforcement learning to robotics by exploiting the subsumption architecture. Proceedins of the Eighth International Workshop on Machine Learning, p. 328-332.

MALIK, J. TAYLOR, C.J. MCLAUGHLAN P. ve KOSECKA, J. 1997. Development of Binocular stereopsis for vehicle lateral control, longitudinal control and obstacle detection. Technical report UCB Path, Final report MOU257.

MANDIRACIOĞLU, A. GÖVSA F. ve HANCI, H. 1997. Dünyada ve Türkiye’de kentlerde otomobil bağımlılığı. Ulaşım ve Trafik Kongresi Bildiriler Kitabı. Ankara, 2-3 Mayıs 1997, s. 102-109.

MANNOR, S. MENACHE, I. HOZE, A. ve KLEIN, M. 2004. Dynamic Abstraction in Reinforcement Learning via Clustering. Proceedings of the 21st International Conference on machine Learning, Banff, Canada, 4-8 July 2004.

MCGOVERN, A. ve BARTO, A.G. 2001. Automatic discovery of subgoals in reinforcement learning using diverse density. In Proceedings of the 18th International Conference on Machine Learning, Morgan Kaufman p. 361-368

MOORE, A.W. 1991. Variable resolution dynamic programming: Efficiently learning action maps in multivariate real-valued state-spaces. Proceedings of the Eighth International Workshop on Machine Learning, p. 323-327.

MORIARTY, D. SCHULTZ, A. ve GREFFENSTETTE, J. 1999. Evolutionary algorithms for reinforcement learning. *Journal of Artificial Intelligence Research* 11 : 199-299

MOZER, M.C. 1986. RAMBOT: A Connectionist expert system that learns by example. Institute for Cognitive Science Report 8610, University of California at San Diego.

MÖRCHEN, F. and ULTSCH, A. 2005. Finding persisting states for knowledge discovery in time series, In *From Data and Information to Knowledge Engineering – Proceedings of the 29th Annual Conference of the German Classification Society, Germany*, p. 660-665.

NAHSC 1995. Current AVCS Deployment. Technical report, National Automated Highway System Consortium.

NG, A.Y. HARADA, D. ve RUSSELL, S. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. *Proc. 16th International Conf. on Machine Learning*, p. 278-287.

NG, A.Y. RUSSELL, S. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, P. 663-670.

REBER, A. 1989. Implicit Learning and tacit knowledge. *Journal of experimental Psychology: General*. 118 (3) : 219-235.

PENTLAND, A. ve LIU, A. 1995. Toward augmented control systems. In *Proceedings of IEEE Intelligent Vehicles*, p. 350-355.

POMERLAU, D.A. 1989. ALVINN: An autonomous land vehicle in a neural network. Technical Report CMU-CS-89-107. Carnegie Mellon University.

RUMELHART, D. MCCLELLAND J. ve PDP Research Group, 1986. *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, MIT Press, Cambridge, MA, 550 s.

SHELL, R. ve DICKMANN E.D. 1994. Autonomous landing of airplanes by dynamic machine vision. *Machine Vision and Applications*, 7 : 127-134.

- SEGER, C. 1994. Implicit Learning. *Psychological Bulletin*. 115 (2) : 163-196.
- SHLADOVER, S.E. 1993. Research and development needs for advanced vehicle control systems. *Micro, IEEE*, 13 (1) : 11-19.
- SLUSARZ P. ve SUN, R. 2001. The interaction of explicit and implicit learning: An integrated model. *Proceedings of Cognitive Science Society Conference*, Edinburgh, p. 952-957.
- SMOLENSKY, P. 1988. On the proper treatment of connectionism. *Behavioral and Brain Sciences*, 11 (1) : 1-74.
- Sun, R. 1992. On variable binding in connectionist Networks. *Connection Science*, 4 (2) : 93-124.
- SUN, R. 1993. An efficient feature-based connectionist inheritance scheme. *IEEE Transactions on System, Man and Cybernetics*, 23 (2) : 512-522.
- SUN, R. 1994. *Integrating Rules and Connectionism for Robust Commonsense Reasoning*. John Wiley and Sons, New York, NY, 273 s.
- SUN, R. 1995. Robust Reasoning: Integrating rule-based and similarity-based reasoning. *Artificial Intelligence*. 75 (2) : 241-296.
- SUN, R. 1999. Accounting for the computational basis of consciousness: A Connectionist Approach. *Consciousness and Cognition*, 8 : 529-565.
- SUN R., 2002. *Duality of Mind*. Lawrence Erlbaum Associates, Mahwah, NJ, 346 s.
- SUN, R. 2003. A Tutorial on CLARION 5.0. Retrieved on October 15, 2003 from Rensselaer Polytechnic Institute, Department of Cognitive Science, Available : <http://www.cogsci.rpi.edu/~rsun/sun.tutorial>.
- SUN, R. ve ZHANG, X. 2002. Top-Down versus Bottom-Up in skill acquisition, *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, Fairfax, VA. Lawrence Erlbaum Associates, Mahwah, NJ, p. 63-89.
- SUTTON, R.S. ve BARTO, A.G. 1998. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 322 s.

- SUTTON, R.S. PRECUP, D. ve SINGH, S. 1999. Between MDPs and semi-MDPs: A Framework for temporal abstraction in reinforcement learning. *Artificial Intelligence Research*, 112 : 181-211.
- TAKADA, Y. ve SHIMOYAMA, O. 2001. Evaluation of driving-assistance systems based on drivers' workload. *International Driving Symposium on Human Factors in Driving Assessment, Training and vehicle design*, Aspen, Colorado, August 14-17.
- TAYLOR, C.J. KOSECKA, J. BLASI R. ve MALIK, J. 1999. A comparative study of vision-based lateral control strategies for autonomous highway driving. *International Journal of Robotics research*, 18 (5) : 442-453.
- TOATES, F. 1986. *Motivational Systems*. Cambridge University Press, Cambridge, UK, 200 s.
- TOUZET, C.F. 1999. *Neural Networks and Q-Learning for Robotics*. 1999 International Joint Conference on Neural Networks. Washington, DC, USA.
- TRIVEDI, M.M. 1989. Designing vision systems for robotic applications. *SPIE*, p. 106-115
- TYRELL, T. 1993. *Computational Mechanisms of Action Selection*. Ph.D Thesis (unpublished), Oxford University, Oxford, UK, 212 s.
- VARAIYA, P. 1991. *Smart Cars on Smart Roads: Problems of Control*. Technical report, California PATH/UC Berkeley.
- WEINER, B. 1992. *Human Motivation: Metaphors, Theories, and research*. Sage Publications. Newbury Park, CA, 408 s.
- WATKINS, C. 1989. *Learning with Delayed Rewards*. Ph.D. Thesis(unpublished), Cambridge University, Cambridge, UK, 234 s.
- WHITEHEAD, S. D. ve BALLARD, D. H. 1991. Learning to perceive and act by trial and error. In: *Machine Learning*, 7 : 45-83.
- WHO 2004. *The Fundamentals*. World report on road traffic injury prevention, p. 1- 9.

6. TEŞEKKÜR

Doktora tezimin hazırlanmasında emeđi geen danıřman hocam Yrd. Do. Dr. Halil YEŐİLİMEN'e, tez alıřması sırasında yardımını esirgemeyen deđerli hocam Do. Dr. İbrahim GÜNEY'e, tezin gelişimindeki katkılarından dolayı Tez İzleme Komitesinin deđerli üyesi Yrd. Do. Dr. H. Osman KOAL'a, alıřmalarım esnasında sabrını ve yardımını benden esirgemeyen sevgili eřim Sibel YENİKAYA'ya ve diđer tüm arkadaşlarıma sonsuz teřekkür ederim.

7. ÖZGEÇMİŞ

Gökhan YENİKAYA, 1975 yılında Bursa'da doğdu. 1993 yılında Bursa Anadolu Lisesini bitirdi. 1997 yılında Dokuz Eylül Üniversitesi Elektronik Mühendisliği bölümünü, 2001 yılında Uludağ Üniversitesi Fen Bilimleri Enstitüsünden yüksek lisansını tamamladı. 1997 yılından beri Uludağ Üniversitesi Elektronik Mühendisliği bölümünde Araştırma görevlisi olarak çalışmalarını sürdürmektedir.