
GAZİ ÜNİVERSİTESİ
Gazi
1926

OTONOM ARAÇLAR İÇİN PEKİŞTİRMELİ ÖĞRENME TABANLI ŞERİT DEĞİŞTİRME ALGORİTMASI GELİŞTİRİLMESİ

Şeyma UYGUR

YÜKSEK LİSANS TEZİ
OTOMOTİV MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TEMMUZ 2025

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

.

Şeyma UYGUR

07/07/2025

OTONOM ARAÇLAR İÇİN PEKİŞTİRMELİ ÖĞRENME TABANLI ŞERİT DEĞİŞTİRME ALGORİTMASI GELİŞTİRİLMESİ

(Yüksek Lisans Tezi)

Şeyma UYGUR

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Temmuz 2025

ÖZET

Günümüzde otonom sürüş teknolojilerinin yaygınlaşması, bu alandaki araştırmaları artırmıştır. Trafik güvenliği, akıcılığı ve sürüş konforu gibi hedeflere ulaşmak için bu teknolojiler sürekli geliştirilmektedir. Otonom sürüş sistemlerinin gelişiminde yapay zekâ, özellikle pekiştirmeli öğrenme (Reinforcement Learning) algoritmaları önemli rol oynamaktadır. Bu çalışmada, otonom araçlar için şerit değiştirme kararını verebilen bir sistem geliştirilmiştir. Karar verme, rota planlama ve rota takibi aşamaları entegre bir şekilde ele alınmıştır. Karar verme süreci için DQN (Deep Q Network) tabanlı pekiştirmeli öğrenme algoritması kullanılmıştır. Şerit değiştirme rotası Sigmoid Fonksiyonu ile oluşturulmuş, rota takibi için Stanley denetleyici tercih edilmiştir. Modelleme, senaryo oluşturma ve test aşamaları MATLAB/Simulink yazılımında gerçekleştirilmiştir. Araç modeli olarak üç serbestlik dereceli bisiklet modeli kullanılmıştır. Senaryo, iki şeritli bir yolda ilerleyen otonom araç ve sabit hızda hareket eden dört çevre araçtan oluşmaktadır. Gözlem kümesi; araçların hız ve konum bilgilerini, aksiyon kümesi ise şerit değiştirme ve şeritte kalma eylemlerini içermektedir. Ödül fonksiyonu, çarpışmasız ve başarılı şerit değişimlerine göre yapılandırılmıştır. Farklı başlangıç koşullarında ve 10–30 m/s hız aralığında oluşturulan senaryolar ile DQN ajanı eğitilmiş; araç, uygun kararlar vererek güvenli şekilde ilerlemiştir.

Bilim Kodu	:	93018
Anahtar Kelimeler	:	Otonom taşıtlar, pekiştirmeli öğrenme, otonom şerit değiştirme, DQN, karar verme, rota planlama
Sayfa Adedi	:	85
Danışman	:	Prof. Dr. Fatih ŞAHİN

DEVELOPMENT OF A REINFORCEMENT LEARNING-BASED LANE CHANGE ALGORITHM FOR AUTONOMOUS VEHICLES

(M. Sc. Thesis)

Şeyma UYGUR

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

July 2025

ABSTRACT

With the growing prevalence of autonomous driving technologies, research interest in this field has significantly increased. These technologies are continuously improved to enhance traffic safety, ensure smooth flow, and maximize driving comfort. Artificial intelligence plays a crucial role in the development of autonomous systems, and among its methods, reinforcement learning offers effective solutions. In this study, a decision-making system for lane changes in autonomous vehicles is developed. The process integrates decision-making, path planning, and path tracking in a unified structure. A Deep Q-Network (DQN)-based reinforcement learning algorithm is used for decision-making. The lane change path is generated using a Sigmoid Function, and the Stanley Controller is employed for path tracking. The modeling, scenario creation, and testing phases are implemented in MATLAB/Simulink. A three degrees-of-freedom bicycle model is used to represent the vehicle. The scenario consists of an autonomous vehicle and four surrounding vehicles moving at constant speeds on a two-lane road. The observation set includes the positions and velocities of vehicles, while the action set consists of lane keeping and lane changing maneuvers. The reward function is designed to encourage successful, collision-free lane changes. The DQN agent is trained using scenarios with random initial positions and speeds ranging from 10 to 30 m/s. The trained agent successfully enables the vehicle to make safe and appropriate decisions under various conditions.

Science Code	:	93018
Key Words	:	Autonomous vehicles, reinforcement learning, autonomous lane change, DQN, decision-making, path planning
Page Number	:	85
Supervisor	:	Prof. Dr. Fatih ŞAHİN

TEŞEKKÜR

Bu yüksek lisans çalışmasının her aşamasında yanımda olan, koşulsuz sevgileri ve desteğiyle beni daima motive eden aileme sonsuz teşekkür ederim. Varlığınız, karşılaştığım zorluklarla baş etmemde en büyük güç kaynağım oldu. Aynı zamanda akademik gelişimime yön veren, bilgi ve tecrübesiyle yolumu aydınlatan, sabrı, anlayışı ve değerli katkılarıyla bu çalışmanın ortaya çıkmasında rolü büyük olan tez danışmanım Sayın Prof. Dr. Fatih ŞAHİN'e en içten şükranlarımı ve teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklamalar**
kg	Kilogram
m	Metre
rad	Radyan
s	Saniye

Kısaltmalar	**Açıklamalar**
CAV	Connected Automated Vehicles
CNN	Convolutional Neural Network
DDPG	Deep Deterministic Policy Gradient
DNN	Deep Neural Network
DOF	Degrees of Freedom
DQN	Deep-Q Network
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LIDAR	Light Detection and Ranging
MDP	Markov Decision Process
ML	Machine Learning
MPC	Model Predictive Control
RADAR	Radio Detection And Ranging
RL	Reinforcement Learning
SVM	Support Vector Machine

1. GİRİŞ

Son yıllarda otomotiv endüstrisi dijitalleşme, yapay zekâ ve sensör teknolojilerindeki hızlı gelişmelerle birlikte önemli bir dönüşüm ivmesi kazanmıştır. Bu dönüşümün öne çıkan alanlarından biri de sürücüsüz yani otonom taşıtların geliştirilmesidir. Otonom taşıtlar insan müdahalesi olmaksızın çevreyi algılayabilir, durum değerlendirmesi yapabilir ve güvenli sürüş kararları alabilir. Bu sayede sürüş güvenliğinin artırılması, yakıt tüketiminin azaltılması, çevresel etkilerin sürüş dinamiklerine etkisinin minimize edilmesi ve trafik akışının maksimum konfor ile sağlanabilmesi gibi önemli kazanımlar elde edilmektedir. Bu yönler göz önüne alındığında otonom taşıtlar geleceğin ulaşım sistemlerinin temeli olarak görülmektedir.

Otonom taşıt teknolojileri sürücüsüz araçların ne kadar bağımsız hareket edebileceğini ifade eden, SAE (Society of Automotive Engineers) tarafından tanımlanan 0'dan 5'e kadar olan otomasyon seviyelerine göre sınıflandırılmaktadır. Bu sınıflandırma kapsamında Seviye 0 gerçek sürücünün bütün sürüş sorumluluğuna sahip olduğu tamamen manuel sürüşü, Seviye 5 ise insan müdahalesi olmaksızın her koşulda tam otonom sürüşü ifade etmektedir. Günümüzde yaygın olarak kullanılan araçlar Seviye 2 ve Seviye 3 otomasyon özelliklerini taşımakta ve belirli koşullarda sürücüye destek olmaktadır. Daha ileri otomasyon seviyeleri ve tam otonom sistemler araştırmalara ve çalışmalara konu olarak geliştirilme aşamasındadır.

Otonom sürüş sistemlerinin en temel gereksinimlerinden biri, aracın çevresini ve trafik durumunu yüksek doğrulukta algılayabilmesidir. Bu amaçla kullanılan kamera, LIDAR, RADAR ve ultrasonik sensörler gibi komponentler bu çevre algılamasını sağlamaktadır. Bu sensörlerden elde edilen verilerin işlenmesi ve yapay zekâ algoritmaları kullanılarak anlamlandırılması ile üç boyutlu çevre modellemesi yapılmaktadır. Böylece otonom araç yol üzerindeki diğer araçlar, yayalar, trafik işaretleri ve yol yapısı gibi çevresel unsurları gerçek zamanlı ve yüksek hassasiyetle tanıyabilmektedir. Algılamada kullanılan komponentlerin kalitesi ve sistemlerin başarısı, otonom sürüşün güvenliği açısından kritik önem taşımaktadır.

Algılama verilerinin yanı sıra aracın kesin konumunu belirlemek amacıyla konum belirleme (localization) ve haritalandırma (mapping) sistemleri de kullanılmaktadır. GPS, atalet

ölçerler (IMU), kamera ve LIDAR verilerinin entegre bir şekilde işlenmesi ile yüksek doğruluklu konum belirleme gerçekleştirilmektedir. Bu sayede araç, harita üzerindeki konumunu gerçek zamanlı olarak takip edebilmekte ve hedef noktaya ulaşmak için gereken navigasyon bilgilerini kullanabilmektedir. Ayrıca dinamik haritalar kullanılarak çevresel değişiklikler ve trafik koşulları sürekli olarak güncellenmekte ve sisteme aktarılmaktadır.

Otonom sürüşün en karmaşık ve kritik aşamalarından biri karar verme ve rota planlama süreçleridir. Bu süreçler aracın içerisinde bulunduğu mevcut trafik durumu, yol koşulları, yerel kurallar ve güvenlik kriterleri göz önünde bulundurularak hız kontrolü, şerit değiştirme, kavşak geçişi gibi manevraları gerçekleştirmesini sağlar. Geleneksel kural tabanlı algoritmalar, belirli ve daha az değişken durumlar için etkili çözümler sunsa da karmaşık ve belirsiz gerçek dünya koşullarında yetersiz kalabilmektedir. Bu nedenle son yıllarda yapay zekanın bir alt dalı olan makine öğrenmesi ve özellikle pekiştirmeli öğrenme tabanlı yöntemler ön plana çıkmıştır. Pekiştirmeli öğrenme aracın çevresel etkileşimlerinden elde ettiği geri bildirimlerle davranışlarını optimize etmesini sağlar. Böylece otonom sistemler, farklı trafik senaryolarında daha esnek, adaptif ve güvenilir kararlar alabilmektedir.

Otonom taşıtların gerçekleştirmesi gereken en önemli sorumluluklardan biri, dinamik ve karmaşık trafik ortamlarında güvenli sürüşü sağlamaktır. Şehir içi trafik, kavşaklar, yoğun otoyol koşulları, yaya yoğunluğu gibi faktörler aracın karar alma mekanizmalarını zorlamaktadır. Bu karmaşık ve dinamik yapı; sensör verilerindeki gürültü ve dalgalanmalar, algılama hataları ve çevresel belirsizliklerle birleştiğinde sistemin güvenilirliğini tehlikeye atan durumlara neden olabilmektedir. Bu nedenle otonom sürüş algoritmalarının gerçekçi senaryolar ve simülasyon ortamlarında kapsamlı test edilmesi gerekmektedir. Simülasyonlar, yüksek maliyet ve risk barındıran gerçek araç testlerinin önünde önemli bir adım olmakla beraber, algoritmaların farklı koşullardaki performansını analiz etmekte ve optimize etmekte kullanılmaktadır.

Otonom taşıt teknolojilerinin geliştirilmesi yalnızca teknik ve mühendislik açılarından değil, hukuki, etik ve sosyal boyutlar açısından da ele alınmalıdır. Otonom araçların yaygınlaşması trafik düzenlemeleri, sorumluluk dağılımı, kişisel veri güvenliği ve etik karar problemleri gibi pek çok yeni konuyu gündeme getirmiştir. Özellikle otonom sistemlerin karşılaşabileceği etik belirsizlikler ve kaza senaryolarında sorumluluğun nasıl paylaşılacağı

gibi hususlar, teknolojinin toplumsal kabulü ve gündelik yaşantıya entegrasyonu için kritik önem taşımaktadır. Bu nedenle otonom taşıt araştırmaları disiplinler arası bir yaklaşımla mühendislik alanına ek olarak ekonomik, psikolojik ve sosyolojik olarak da incelenmektedir.

Otonom taşıt teknolojileri, otomotiv mühendisliği başta olmak üzere yapay zekâ, robotik, bilgisayar mühendisliği, yazılım ve sosyal bilimlerin kesişiminde yer alan multidisipliner ve dinamik bir araştırma konusudur. Bu alanda yapılan yenilikler ve gerçekleştirilen çalışmalar yalnızca teknolojik ilerlemeleri değil, aynı zamanda toplumsal faydayı ve sürdürülebilir ulaşımı da desteklemektedir.

Bu çalışma otonom taşıtların yalnızca karar verme değil, otonom şerit değiştirme hareketinde kritik öneme sahip rota planlama ve planlanan rotanın takibi konularına da yenilikçi bir yaklaşım geliştirmeyi amaçlamaktadır. Çalışmada geliştirilen algoritmalar MATLAB ve Simulink yazılımı ile modellenmiş, çeşitli trafik senaryolarında test edilmiştir. Bu sayede önerilen yöntemlerin güvenlik, konfor ve dinamik ortamlarda kullanılabilmesi açılarından etkinliği ve kullanılabilirliği değerlendirilmiş ve mevcut yöntemlerle karşılaştırılmıştır. Tezin amacı, otonom sürüş sistemlerinin farklı trafik koşullarında güvenilirliğini artırmak, daha etkin karar alma yeteneği kazandırmak ve aracın şerit değiştirme hareketini doğru bir şekilde tamamlamasını sağlamaktır.

Bu tez çalışmasında öncelikle kullanılan yöntemler detayları ile açıklanmış ve tercih edilen yöntemlerin tercih edilme sebepleri açıklanmıştır. Simülasyonlarda kullanılan taşıt modeli açıklanarak belirlenen senaryo koşulları parametreleri belirtilmiştir. Karar verme algoritmasında kullanılan DQN yöntemi içerisindeki gözlem kümesi, aksiyon kümesi, ödül fonksiyonu gibi parametreler detaylandırılmıştır. Bir sonraki aşama olan rota planlama aşamasında kullanılan Sigmoid fonksiyonunun uygulaması açıklanmış, rotaya ait eğrilik hesaplamaları yapılmıştır. Rota takibi için kullanılan Stanley denetleyici parametreleri açıklanarak DQN ajanının eğitimi ile ilgili detaylar tanımlanmıştır. Eğitim süreci sonucunda taşıtın davranışları ve sistemin doğruluğu test edilerek grafik ve görsellerle ifade edilmiştir.

2. OTONOM TAŞITLARDA ŞERİT DEĞİŞTİRME MANEVRASI

Otonom taşıtlar otomotiv ve ulaşım teknolojisinde köklü bir dönüşümü temsil etmektedir. İnsan müdahalesi olmadan taşıtın hareketini sağlayan otonom sistemler birden fazla teknolojinin ve disiplinin bir araya gelmesi ile kompleks sayılabilecek ancak tamamen entegre bir sistemi ifade eder. Temel olarak araç üzerine yerleştirilen çeşitli kamera, sensör vb. ekipmanlarla çevre tespiti yapılır. Bu tespitler yol genişliği, şerit ayrımları, çevre taşıtlar, yayalar, tabelalar gibi otonom olmayan sürüş esnasında bir sürücünün dikkat etmesi gereken unsurları kapsar. Otonom araçların gelişiminin desteklenmesi farklı bakış açılarından çeşitli avantajlara sahiptir. Sürüş esnasında sürücü hatalarından kaynaklanabilecek kaza veya hatalı sürüş durumlarının azaltılarak şehir içi ya da şehirler arası trafiğin daha güvenli hale getirilmesi amaçlanır. Sürücü dikkatsizliği, zayıf refleksler, olumsuz yol şartları ve yavaş tepki verme gibi sebeplerden meydana gelen kazalarda taşıt hızına bağlı olarak maddi hasarlar ve ciddi yaralanmalar meydana gelebilmektedir. Otonom sürüş sayesinde taşıt hareketi ve hızı yol şartlarına göre optimize edilerek kazaların ve dolayısıyla yaralanmaların önüne geçilebilir. Çeşitli fiziksel engellere sahip olması nedeniyle araç kullanamayan bireylerin günlük yaşantılarını kolaylaştırmak ve başka kişilere bağımlılığını azaltarak topluma kazandırmak da otonom araç geliştirmelerinin amaçlarından bir tanesidir. Kararsız sürüş dinamikleri ya da sıkışık trafik durumlarında taşıtların harcadığı yakıt, ortalama yakıt tüketimleri ve emisyon salınımları artmaktadır. Otonom taşıtlarda bu durum da göz önüne alınarak taşıt hareketi kolaylaştırılır. Bununla beraber daha kararlı bir sürüş karakteristiğinin bulunduğu bir çevrede trafik olgusu da azalacağı için verimlilik ve yakıt ekonomisi iyileştirilir, çevre kirliliğinin azaltılmasına yardımcı olunur. Otonom taşıt tarafından idealize edilen sürüş karakteristiği sayesinde trafiğin azalacağı, bu sayede trafikte geçirilen zamanın azalması ile beraber zaman tasarrufu sağlanabileceği; bununla beraber doğru park konumlandırması sayesinde de park alanlarından tasarruf edilebileceği ön görülmektedir. Otonom taşıtlar teknolojik etkiler bakımından çeşitli entegrasyonlar ile güvenliğin artışı ve konforun yükseltilmesinde önemli bir adım olup farklı disiplinleri bünyesinde barındırarak maksimum konforu ve güvenliği sağlamayı amaçlamaktadır (Gherardini ve Cabri, 2024).

Otonom taşıtlar sürücüye ve trafiğe konfor ve güvenlik bakımından önemli katkılar sağlasa da çeşitli dezavantajları da bulunmaktadır. Bunlardan ilki veri güvenliği olarak söylenebilir. Otonom taşıtlar hareketini sağlamak için bütün verileri sensör ve kameralardan elde eder. Bu komponentlerde meydana gelebilecek herhangi bir aksaklık ya da eksiklik taşıt ve

içerisindeki kişiler için tehlike yaratma ihtimalini ortaya çıkarmaktadır. Bu nedenle bu gibi parçaların sağlamlık ve güvenilirliğinin oldukça hassas şekilde doğrulanması gerekir (Yeong ve diğerleri, 2021).

Otonom taşıtların bir diğer negatif yönü ise bünyesinde kullanılan komponentlerin pahalılığıdır. Taşıt en doğru sonuca varabilmek ve taşıtı en güvenli şekilde doğru yönlendirebilmek için pek çok parça (GPS, LIDAR, RADAR…) kullanır. Bu parçaların çoğunlukla elektronik temelli ve pahalı parçalar olması taşıt maliyetinin de yükselmesi anlamına gelir. Diğer yandan bu parçalarda meydana gelebilecek herhangi bir hasar ya da arıza durumunda bakım ve onarım maliyetleri de ekonomik sayılabilecek seviyenin oldukça üzerinde olacağı tespit edilebilmektedir. Günümüzde etkileri yoğun bir şekilde görülmese de ilerleyen yıllarda otonom taşıt teknolojisinin de gelişmesi ile beraber taşıt kullanımına bağlı iş sektörlerinde işçi ihtiyacının azalacağı ön görülmektedir. Bu durum işsizlik adına toplumsal olarak bir tehdit olarak görülebilmektedir. Otonom taşıtların günlük kullanımının artışı büyük oranda sürücülerin bu teknolojiye uyum sağlamasıyla doğru orantılıdır. Kullanıcıların otonom sürüşü tercih etmeleri, güven kazanabilmeleri ve alışabilmeleri için belli bir süre gereksinimi mevcuttur. Teknolojinin ve çalışmaların gelişimi ve sistem güvenilirliğinin artması bu süreyi kısaltabilecek faktörlerdendir denebilir (Yeong ve diğerleri, 2021).

2.1. Otonom Taşıtların Sınıflandırılması

Otonom taşıtlar sürücüye bağımlılık oranına göre 6 farklı seviyede sınıflandırılmıştır. Bu seviyeler ve sınıflandırmalar SAE (Society of Automobile Engineers) tarafından belirlenmiştir. Bu sınıflandırma aşağıdaki gibi açıklanabilir:

Seviye 0 – Otomasyon Yok: Bu seviyede taşıt kontrolü tamamen sürücüye bağlıdır. Sistem sürücünün kontrolü dışında herhangi bir hareket ya da manevra girişiminde bulunmaz. Bununla beraber bazı aktif güvenlik sistemleri sürücüyü uyarmak adına devreye girebilir. Bu sistemler şerit takip uyarısı, kör nokta uyarısı, ESC (Electronic Stability Control) gibi sistemlerdir (Technologies, 2022).

Seviye 1 – Sürüş Asistanı: Bu seviyede sistem direksiyon, gaz pedalı, fren pedalı gibi kısımlara müdahale ederek taşıt hızını ve hareket yönünü değiştirebilir. Örneğin ACC

(Adaptive Cruise Control) sistemi öndeki araçla mesafe ve hız takibi yaparak taşıt hızına etki eder. Şerit takip sistemi ile de aracın bulunduğu şeridin dışına kontrolsüz şekilde çıktığı tespit edilirse sistem aracı mevcut şeritte tutmak için direksiyona müdahale eder. Sürücü daima aracın kontrolüne sahip olmalıdır. Her iki fonksiyonun da aktifliği sürücü tarafından iptal edilebilir ya da devreye alınabilir (Zhao ve diğerleri, 2024).

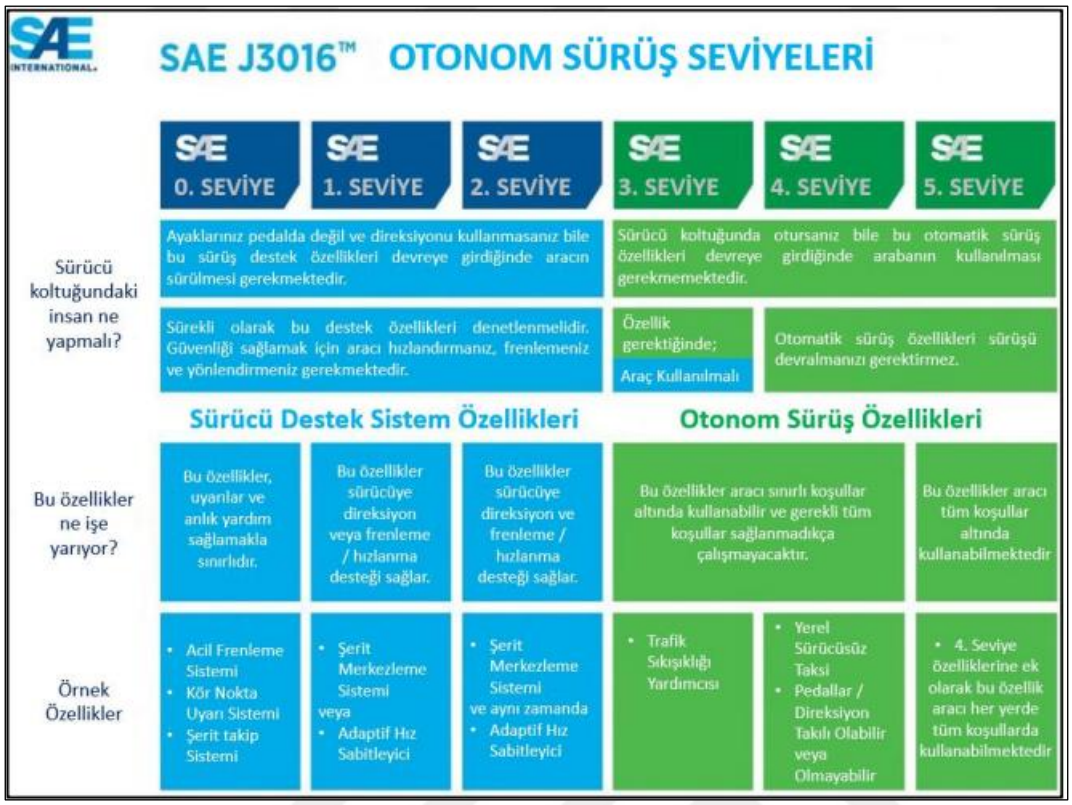
Seviye 2 – Kısmi Otomasyon: Bu seviye bir önceki ile benzerlik gösterse de daha kapsamlı bir sistem olduğu söylenebilir. Taşıt bu seviyede gaz ya da fren pedalı ile direksiyona aynı anda müdahale edebilir. Başka bir deyişle ACC (Adaptive Cruise Control) ve şerit takip sistemi eş zamanlı olarak kullanılabilmektedir. Bu seviyede de sürücü aracın tam kontrolüne sahip olmalıdır (Wienrich, 2022; Zhao ve diğerleri, 2024).

Seviye 3 – Koşullu Otomasyon: Bu seviyede taşıt sürüş görevini sürücüden tamamen alabilir. Sürücü belirli bir seviyeye kadar aracın kontrolünü bırakıp farklı şeylerle ilgilenebilir. Ancak sistem kontrol limitlerine ulaştığında sürücüyü sesle veya titreşimle uyararak kontrolü ele almasını talep eder. Bu noktada sürücünün bir süre aracın kontrolünü sağlaması ya da sadece kontrolü ele alabileceğini bildirecek şekilde direksiyonu belli bir süre ve şiddette tutması sistem tarafından beklenebilir (Inagaki ve Sheridan, 2019; Wienrich, 2022; Zhao ve diğerleri, 2024).

Seviye 4 – Yüksek Otomasyon: Bu seviyede sistem aracı kendi kendine kontrol edebilir. Sürücüden herhangi bir kontrol ya da müdahale beklemez. Buna ek olarak bu seviyede araçlarını piyasaya süren üreticiler seyir esnasında sürücülerin yemek yiyebileceklerini, film izleyebileceklerini ve hatta uyuyabileceklerini söylemişlerdir. Bu seviyedeki taşıtlar için acil durumlar dışında sürücüye ihtiyacı yoktur (Zhao ve diğerleri, 2024).

Seviye 5 – Tam Otomasyon: Bu seviyede herhangi bir sürücüye, dolayısıyla gaz ve fren pedalı ya da direksiyona ihtiyaç duyulmaz. Bütün koşullarda belirlenen istikamette sistem taşıt hareketini sağlayabilir. Çeşitli ülkelerde şehir içi toplu ulaşım ya da kargo/lojistik gibi alanlarda bu tip araçlar kullanılmaktadır (Inagaki ve Sheridan, 2019; Wienrich, 2022; Zhao ve diğerleri, 2024).

Taşıtlar için SAE J3016 ile belirlenen otomasyon seviyeleri Şekil 2.1.'de gösterilmiştir.



Şekil 2.1. Taşıt otomasyon seviyeleri (Günay, 2021)

Otonom taşıtlar; algılama, karar verme ve kontrol gibi temel işlevleri insan müdahalesi olmaksızın yerine getirebilen oldukça gelişmiş sistemlerdir. Bu sistemlerin sınıflandırılması, avantajları ve mevcut sınırlılıkları ele alındığında tam anlamıyla güvenli ve etkili bir otonom sürüş deneyimi sağlamak için yalnızca gelişmiş donanımların yeterli olmadığı görülmektedir. Araçta kullanılan lidar, kamera, radar gibi sensörler çevreyi fiziksel olarak algılayabilse de bu verilerin anlamlı bilgilere dönüştürülmesi, işlenmesi, yorumlanması ve doğru zamanda doğru kararların alınması için daha ileri düzeyde ve dinamik geçişlere uyum sağlayabilecek sistemlere ihtiyaç duyulmaktadır. Özellikle gerçek dünyadaki trafik ortamının belirsiz ve çoğu zaman öngörülemez yapısı göz önüne alındığında sistemlerin belirli senaryolara özel kural tabanlı çözümlerle yönetilmesi çoğu durumda yetersiz kalmaktadır.

Bu noktada devreye giren yapay zekâ teknolojileri, otonom taşıtların çevresel verileri sadece algılamasını değil, aynı zamanda bu verilerden öğrenmesini, yorum yapmasını ve karmaşık sürüş senaryolarına uyum sağlayarak doğru kararlar alabilmesini mümkün kılar. Yapay zekâ sayesinde araçlar, yaya, araç gibi trafik ve çevreye ait unsurları tanımlamakla kalmaz, bu unsurların hızını ve yönünü analiz ederek olası hareketlerini öngörebilir. Böylece, önceden

programlanmış senaryolara bağlı kalmaksızın, anlık durumlara uygun otonom kararlar üretilebilir. Bu da hem sürüş güvenliği hem de konfor açısından sistemlerin daha güçlü, bağımsız ve insan benzeri davranışlar sergileyebilmesini sağlar.

2.2. Otonom Taşıtlarda Kullanılan Yapay Zekâ Yöntemleri

Yapay zekâ günümüzde çok çeşitli disiplinlerde devrim yaratacak nitelikte yenilikler sağlamış, yalnızca bilgi teknolojileriyle sınırlı kalmayıp otomotiv, sağlık, finans ve üretim gibi alanlarda da etkili olmuştur. Bu disiplinin temelini insan zekâsını taklit eden sistemlerin geliştirilmesi oluşturur. Yapay zekanın temel dallarından biri olan makine öğrenmesi (Machine Learning- ML), veriden öğrenmeyi mümkün kılar. Bu yaklaşımda sistemler, açıkça programlanmaksızın deneyimlerden öğrenerek karar verebilirler. Makine öğrenmesi, denetimli öğrenme (supervised learning), denetimsiz öğrenme (unsupervised learning) ve pekiştirmeli öğrenme (reinforcement learning) gibi alt başlıklara ayrılır (Goodfellow ve diğerleri, 2016).

Denetimli öğrenme algoritmalarında sistem giriş ve çıkış verileriyle eğitilir. Sınıflandırma (classification) ve regresyon (regression) bu kategoriye dâhildir. Örneğin lojistik regresyon (logistic regression) ve destek vektör makineleri (support vector machines- SVM) sınıflandırma problemlerinde yaygın kullanılırken, doğrusal regresyon (linear regression) ve karar ağaçları (decision trees) gibi yöntemler sayısal tahminler için uygundur. Bununla beraber rastgele orman (random forest) gibi topluluk yöntemleri, birden fazla modelin çıktısını birleştirerek doğruluğu artırmayı amaçlar (Hastie ve diğerleri, 2009).

Denetimsiz öğrenme ise veri üzerinde herhangi bir etiket olmaksızın yapısal ilişkileri keşfetmeyi hedefler. Kümeleme (clustering) bu yaklaşımın önde gelen tekniklerinden biridir ve K-ortalama kümeleme (K-means clustering) ya da hiyerarşik kümeleme (hierarchical clustering) gibi algoritmalarla gerçekleştirilir. Boyut indirgeme (dimensionality reduction) teknikleri ise verilerin daha düşük boyutlu temsilini elde etmeyi amaçlar. Bu amaçla kullanılan temel yöntemler arasında temel bileşen analizi (principal component analysis - PCA) ve tekil değer ayrıştırması (singular value decomposition - SVD) yer alır (Bro ve Smilde, 2014) .

Derin öğrenme (deep learning), makine öğrenmesinin bir alt kümesidir. Makine öğrenmesine kıyasla daha karmaşık ve çok katmanlı yapay sinir ağlarını kullanarak yüksek seviyeli sistemlerin ve ajanların öğrenimini sağlar. Evrişimli sinir ağları (convolutional neural networks- CNNs) özellikle görüntü işleme alanında başarılı sonuçlar verirken, yinelemeli sinir ağları (recurrent neural networks- RNNs) zaman serisi verilerinde ya da dil modellemede tercih edilir. Uzun- kısa dönem hafıza (LSTM - long short-term memory) ağları ise RNN'lerin uzun etkileşimlere sahip sistemlerde öğrenme konusundaki sınırlamalarını aşmak için geliştirilmiştir (LeCun ve diğerleri, 2015).

Son yıllarda popülerleşen transformer mimarisi, özellikle doğal dil işleme (natural language processing - NLP) alanında devrim niteliğinde ilerlemeler sağlamıştır. Bu mimari sayesinde metin üretimi, özetleme, çeviri gibi işlemler daha verimli ve mevcut ortama duyarlı biçimde gerçekleştirilebilmektedir. Doğal dil işleme modellerinin en yaygın örneklerinden biri olan BERT (Bidirectional Encoder Representations from Transformers), çok katmanlı mimarisiyle bağlamsal anlamı etkin biçimde modelleyebilir (Vaswani ve diğerleri, 2017).

Çekişmeli üretici ağlar (Generative Adversarial Networks GANs) gibi üretici yapılar da özellikle görüntü sentezi, restorasyon ve veri genişletme konularında yaygın olarak kullanılır. GAN'ler bir üretici (generator) ve ayırt edici (discriminator) ağdan oluşur; bu iki yapı birbirine karşı çalışarak yüksek kaliteli veriler üretir. GAN'lerin sanat, sağlık görüntüleme ve sahte veri üretimi gibi alanlarda birçok yenilikçi uygulaması bulunmaktadır (Goodfellow ve diğerleri, 2016).

Yapay zekanın alt dallarından biri olan pekiştirmeli öğrenme (reinforcement learning - RL), ajanların çevreyle etkileşime girerek ödül sinyali üzerinden strateji öğrenmesine dayanır. Bu alanda model-tabanlı ve modelden-bağımsız yaklaşımlar olmak üzere iki temel yöntem öne çıkar. Model tabanlı yöntemler, çevrenin matematiksel veya istatistiksel bir modelini oluşturarak karar verme süreçlerini yürütür. Bu yöntemlerde genellikle ortamın dinamikleri yani eylem gerçekleştirildiğinde ortamın nasıl değişeceği bilinir veya öğrenilir. Modelden bağımsız yöntemler, çevrenin nasıl çalıştığını bilmeden veya öğrenmeden yalnızca eylemler ve ödüller arasındaki ilişkiye odaklanır. Ajan, deneme-yanılma yoluyla en iyi eylemleri öğrenir. Ortamın dinamiğini bilmeye gerek yoktur. Modelden bağımsız yöntemler değer tabanlı, politika tabanlı olarak iki ayrı grupta incelenir. Değer tabanlı (value-based) yöntemlerde ajan, her durumda en yüksek toplam ödülü sağlayacak eylemleri öğrenirken;

politika tabanlı (policy-based) yaklaşımlarda doğrudan bir politika fonksiyonu öğrenilir. Derin pekiştirmeli öğrenme (deep reinforcement learning - DRL), sinir ağlarını kullanarak bu süreçleri daha karmaşık ortamlar için uygular (Sutton ve Barto, 1998). Çoklu ajanlı pekiştirmeli öğrenme (multi-agent RL) ise birden fazla ajanın aynı ortamda eşzamanlı öğrenmesini kapsar.

Doğal dil işleme (NLP), yapay zekanın dilsel veriyle çalışmasını sağlar. Bu alanda ön işleme (text preprocessing), sözcük analizi (lexical analysis), sözdizimsel çözümleme (syntactic analysis) ve anlamsal analiz (semantic analysis) gibi aşamalar bulunur. Sözcüklerin köklerini ayıklamak için kök bulma ve lemmatizasyon teknikleri, anlamlandırma içinse anlamsal rol etiketleme (semantic role labeling) ve ad-öbek tanıma (named entity recognition - NER) gibi yöntemler kullanılır (Jurafsky ve Martin, 2020).

Görüntü işleme alanındaki bilgisayarlı görü (computer vision) uygulamaları, nesne algılama (object detection), nesne takibi (object tracking), görüntü sınıflandırma (image classification), görüntü bölütleme (segmentation) ve restorasyon (restoration) gibi görevleri içerir. Bu görevler genellikle derin sinir ağları ve CNN'ler aracılığıyla gerçekleştirilir. Özellikle otonom taşıtlarda, çevresel algılama görevlerinde bilgisayarlı görü uygulamaları kritik rol oynar (Szegedy ve diğerleri, 2015).

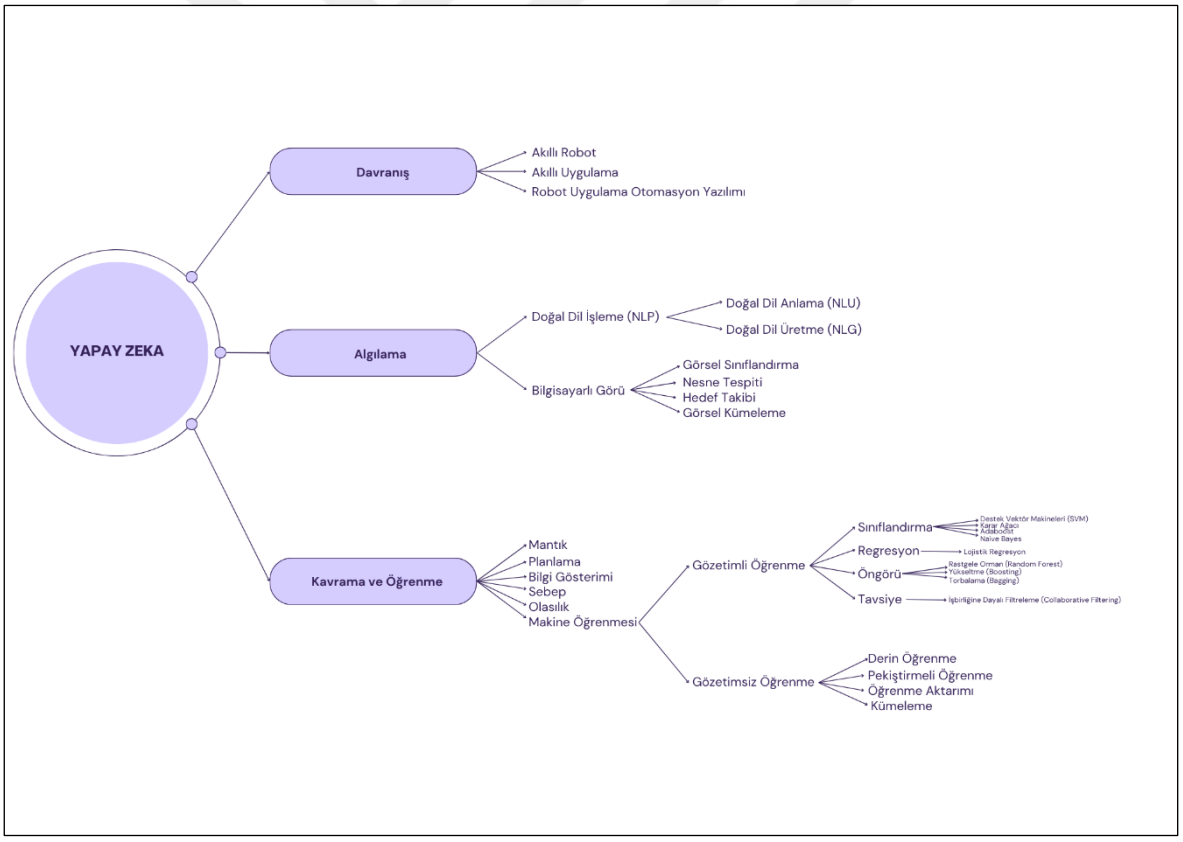
Yapay zekanın otonom sistemlerdeki etkisi, özellikle robotik sistemlerde açıkça görülmektedir. Robotik alanında hem denetimli hem de denetimsiz öğrenme yöntemleri kullanılmaktadır. Otonom taşıtlar, sürücü müdahalesi olmaksızın çevresel algı, karar verme ve hareket kontrolü işlevlerini yerine getirebilmek için yapay zekanın bu alt alanlarından faydalanmaktadır. Bu sistemlerde, görüntü işleme yoluyla algılama, pekiştirmeli öğrenme yoluyla karar verme ve kontrol mekanizmaları birlikte çalışır. Özellikle derin öğrenme temelli yaklaşımlar, otonom taşıtlarda şerit takip, engelden kaçınma, hız optimizasyonu gibi görevlerde yüksek başarı sağlar (Bojarski ve diğerleri, 2016).

Ayrıca açıklanabilir yapay zekâ (explainable AI- XAI), nöro-sembolik yapay zekâ (neurosymbolic AI) ve üretken yapay zekâ (generative AI) gibi yükselen alanlar, yapay zekanın daha anlaşılır, etik ve yaratıcı biçimde kullanılmasını hedeflemektedir. XAI, kararların şeffaflığını artırmayı; nöro-sembolik yapay zeka, sembolik akıl yürütmeyle sinir

ağlarının birleşimini; üretken yapay zeka ise özgün içerik üretimini amaçlamaktadır (Gunning ve Aha, 2019).

Bütün yönleri ile ele alındığında yapay zekâ günümüzde hem kuramsal hem de uygulamalı yönleriyle hızla gelişmekte, birçok sektör ve teknolojiye yön vermektedir. Özellikle otonom taşıt sistemlerinde, tüm bu yöntemlerin bir araya gelerek oluşturduğu sinerji sayesinde insan müdahalesi olmaksızın güvenli ve etkili karar alma mekanizmaları tasarlanabilmektedir. Bu doğrultuda yapay zekanın gelecekteki uygulama alanlarının daha da genişlemesi kaçınılmaz görünmektedir (Russell ve Norvig, 2021).

Yapay zekâ teknolojilerinin kullanım alanları ve alt dallarının görselleştirildiği şema Şekil 2.2.'de verilmiştir.



Şekil 2.2. Yapay zekâ yöntemlerinin sınıflandırılması

2.2.1. Pekiştirmeli öğrenme

Pekiştirmeli öğrenme (Reinforcement Learning - RL) yapay zekânın makine öğrenmesi alt dallarından en özgün ve karmaşık yöntemlerinden biridir (Sutton ve Barto, 1998). Bu makine öğrenmesi algoritması, temelde bir ajanın çevreyle etkileşim kurarak deneyim kazanması ve bu deneyimlerden yola çıkarak uzun vadeli ödülü maksimize edecek optimal davranış stratejilerini geliştirmesi prensibine dayanmaktadır. Geleneksel denetimli öğrenme yöntemlerinin aksine, pekiştirmeli öğrenmede açıkça tanımlanmış bir eğitim veri seti bulunmamakta ve sistem doğru davranışın ne olduğunu bilmemektedir. Bunun yerine ajan, aldığı ödül sinyallerine göre kademeli olarak davranışlarını iyileştirmektedir (Kaelbling ve diğerleri, 1996). Bu öğrenme insan öğrenme mekanizmasına oldukça benzemekte ve deneme-yanılma (trial-and-error) yaklaşımı üzerine kuruludur. Pekiştirmeli öğrenmenin bu özgün yapısı özellikle dinamik ve belirsiz ortamlarda oldukça etkili bir çözüm olarak karşımıza çıkmaktadır. Örneğin, bir robotun yürümeyi öğrenmesi veya bir yapay zekanın satranç oynamayı öğrenmesi gibi kompleks görevlerde pekiştirmeli öğrenme algoritmaları sıklıkla kullanılmaktadır (Silver ve diğerleri, 2016) .

Pekiştirmeli öğrenmenin matematiksel temeli, Markov Karar Süreçleri (Markov Decision Processes - MDP) olarak adlandırılan bir çerçeve ile modellenmektedir (Puterman, 2014). Bu süreçte ajan, içinde bulunduğu durumdan (state) bir aksiyon (action) seçerek, ortamın durumunu değiştiren bir geçiş (transition) oluşturur ve bir ödül (reward) alır. Zaman içinde elde edilen toplam ödülü maksimize edecek bir stratejiyi, yani politikayı (policy) öğrenmeyi amaçlar. Matematiksel olarak MDP=(S,A,P,R,γ) şeklinde tanımlanır. Bu matematik modeli durum kümesi (S), eylem kümesi (A), geçiş olasılıkları (P), ödül fonksiyonu (R) ve indirim oranı (γ) olmak üzere beş temel bileşenden oluşur (Sutton ve Barto, 1998).

Pekiştirmeli öğrenmede ajanın amacı, bir politika (π) doğrultusunda gelecekte elde edeceği ödüllerin toplamını maksimize etmektir (Sutton ve Barto, 1998). Beklenen toplam ödül, durum-eylem çiftlerine atanmış bir değer fonksiyonu olan Q-fonksiyonu ile ifade edilir. Bu fonksiyon Bellman Denklemi ile tanımlanır (Bellman, 1966). Bellman denklemi 1 numaralı eşitlikte verilmiştir.

$$Q^{\pi}(s,a) = \mathrm{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} R(s_t, a_t) | s_0 = 0, a_0 = 0\right] \quad (2.1)$$

Optimal politika, her adımda en yüksek beklenen ödülü sağlayacak aksiyonu seçer. Bu durumda optimal Q-değerleri Bellman Optimalite Denklemi olarak bilinir ve 2 numaralı eşitlik ile ifade edilir (Bellman, 1966):

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s'} P(s' \,|\, s,a) \max_{a'} Q^*(s',a') \tag{2.2}$$

Bu eşitlik, dinamik programlama yaklaşımına dayalı algoritmalarla iteratif olarak çözülebilir. Ancak pek çok gerçek dünya problemi, geçiş olasılıkları bilinmeyen ya da sonlu olmayan durum/eylem uzayları içerdiğinden dolayı doğrudan çözüm mümkün değildir.

Pekiştirmeli öğrenme algoritmaları, ortamın dinamiklerini bilip bilmemelerine göre model tabanlı (model-based) ve modelden bağımsız (model-free) yöntemler olarak ikiye ayrılır (Sutton ve Barto, 1998). Model tabanlı yöntemlerde, ajan çevrenin dinamiklerini (geçiş olasılıklarını ve ödül fonksiyonlarını) karar verme sürecinden önce bilir ya da öğrenir. Bu bilgilerle simülasyon yaparak ya da planlama yöntemleri kullanarak en uygun eylemi seçer. Örneğin, Dyna-Q algoritması (Sutton ve Barto, 1998) hem deneyimden öğrenmeyi hem de modelden simülasyon üretip o deneyimi kullanarak öğrenmeyi birleştiren entegre bir algoritmadır. Bununla beraber Model Predictive Control (MPC) ve Monte Carlo Tree Search (MCTS) gibi planlama algoritmaları da model tabanlı pekiştirmeli öğrenme yöntemleri ile yüksek benzerlik gösteren ve genellikle pekiştirmeli öğrenme algoritmaları ile entegre çalışabilen yöntemlerdir.

Bir diğer yöntem olan modelden bağımsız yöntemlerde ajan, çevrenin dinamiklerine dair herhangi bir bilgiye sahip değildir. Öğrenme tamamen deneyime dayanır. Modelden bağımsız pekiştirmeli öğrenme algoritmaları genellikle çevre modelinin bilinmediği veya öğrenilmediği durumlarda kullanılmaktadır (Mnih ve diğerleri, 2015). Modelden bağımsız yöntemler değer tabanlı (value-based), politika tabanlı (policy-based) ve aktör-kritik (actor-critic) yöntemler olarak üçe ayrılır. Q-learning gibi değer tabanlı algoritmalar, her durum-eylem çiftinin değerini tahmin ederek en iyi eylemi seçmeye çalışır.

Değer tabanlı yöntemler, optimal bir değer fonksiyonu (value function) veya Q-fonksiyonu (action-value function) öğrenmeye odaklanmaktadır (Dayan ve Watkins, 1992). Q-öğrenme (Q-Learning), bu kategorinin en sık kullanılan algoritması olmakla beraber zaman farkı (temporal difference) yöntemine dayanmaktadır. Algoritma, Q-değerlerini güncellemek için

mevcut Q-değeri ile hedef Q-değeri (bir sonraki durumda alınabilecek maksimum Q-değeri) arasındaki hatayı kullanmaktadır. Matematiksel olarak 3 numaralı eşitlikteki gibi ifade edilir.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + R(s,a) + \gamma \max_a Q(s', a') - Q(s_t, a_t)] \quad (2.3)$$

Burada α öğrenme oranı ($0 < \alpha \leq 1$), γ indirim faktörü, s′ yeni durum r_{t+1} ise (t+1) zaman adımında alınan ödülü ifade etmektedir. Q-learning algoritması, çevreden gelen geri bildirimlerle taban değerlerini iteratif olarak günceller ve optimal politikayı dolaylı olarak öğrenir. Bir diğer değer tabanlı algoritma olan SARSA algoritması ise güncel politikaya bağlı olarak Q-değerlerini günceller. Q-learnnig algoritmasının derin öğrenme ile birleştirildiği DQN (Deep Q-Network) metodu da sıklıkla kullanılmakta ve başarılı sonuçlar elde edilmektedir. Bu algoritmalar genellikle giriş seviyesinde pekiştirmeli öğrenme uygulamaları için etkilidir (Dayan ve Watkins, 1992). Ancak ayrık eylem alanlarında etkili olan bu yöntem, sürekli eylem alanlarında yetersiz kalır.

Politika tabanlı yöntemlerde, ajan direkt politikayı yani eylem seçme stratejisini öğrenmekte ve doğrudan bir politika fonksiyonu $\pi_\theta(a|s)$ öğrenmeye odaklanmaktadır (Williams, 1992). Bu yaklaşım, özellikle sürekli eylem uzaylarında (örneğin robotik kontrol problemleri) ve stokastik politikalar gerektiren durumlarda avantaj sağlamaktadır. Politika Gradyan (Policy Gradient) teoremi, bu tür algoritmaların matematiksel temelini oluşturmaktadır. Bu yöntemler sürekli eylem uzaylarında ve karmaşık problemler için daha uygundur. REINFORCE algoritması politikanın gradyana göre optimize edilmesine dayanır. Bu algoritmanın matematiksel ifadesi 4 numaralı eşitlikte verilmiştir.

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi(a|s\,;\,\theta)\, G_t \quad (2.4)$$

Eşitlikte yer alan θ ajanın öğrenmeye çalıştığı politika parametrelerini ifade eder. α ifadesi öğrenme oranını belirler ve ajanın ilerleyeceği adımların büyüklüğünü kontrol eder. $\nabla_\theta \log \pi(a|s;\theta)$ ifadesi, ajanın belirli bir durumda (s) belirli bir eylemi (a) seçme olasılığının logaritmasının gradyanıdır ve ajanın yaptığı eylemin ne kadar iyi olduğunu değerlendirmesine yardımcı olur. G_t ise ajanın gerçekleştirdiği eylemden sonra gelecekte beklediği toplam ödülü ifade eder. Bu yapı sayesinde ajan, başarılı eylemleri daha sık tercih etmeyi, başarısız olanları ise azaltmayı öğrenir. Politika tabanlı derin pekiştirmeli öğrenme yöntemleri arasında en sık kullanılan diğer algoritmalar Proximal Policy Optimization (PPO)

ve Trust Region Policy Optimization (TRPO) algoritmalarıdır. PPO, politikanın her güncellemede fazla değişmesini engelleyerek öğrenme sürecini daha kararlı hale getirir (Schulman ve diğerleri, 2015). Bununla beraber hem sürekli hem ayrık eylem uzaylarında yüksek performans göstermektedir. Ancak politika tabanlı yöntemler yüksek varyans sorununa sahiptir ve bu durum öğrenmenin dengesiz olmasına yol açabilir.

Bu problemi azaltmak için aktör-kritik yöntemleri geliştirilmiştir. Aktör-kritik yöntemler hem değer fonksiyonlarını hem de politikayı birlikte öğrenerek bu iki yaklaşımın avantajlarını birleştirmektedir (Konda ve Tsitsiklis, 1999). Bu yöntemlerde aktör politika geliştirirken, kritik (eleştirmen) (critic) bu politikanın performansını değerlendirmektedir. Aktör-kritik yöntemlerinin literatürde de sıkça karşılaşılan ve kullanılan bir kolu DDPG (Deep Deterministic Policy Gradient) yaklaşımıdır. DDPG, klasik Q-learning'in sürekli eylem alanlarına uyarlanmış versiyonu olarak da düşünülebilir. Q-learning ayrık (discrete) eylem alanlarında uygulanırken, DDPG'de bu mümkün olmadığından, maksimum değeri doğrudan çıkaran bir politika fonksiyonu (aktör) öğrenilir. Kritik ağı ise bu politikanın çıktığı eylemin ne kadar iyi ve kullanılabilir olduğunu değerlendirir. DDPG bu yönüyle DQN ile politika gradyan yöntemlerinin bir sentezidir (Lillicrap ve diğerleri, 2015; Sutton ve Barto, 1998).

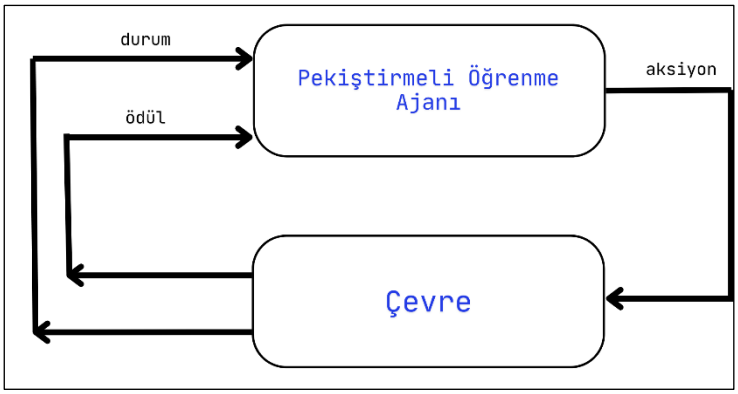
Pekiştirmeli öğrenme, son yıllarda çeşitli alanlarda başarılı şekilde uygulanmıştır (Arulkumaran ve diğerleri, 2017). Oyunlar ve strateji geliştirme alanında, DeepMind'ın geliştirdiği AlphaGo ve AlphaZero sistemleri gibi örneklerde pekiştirmeli öğrenme algoritmalarının karmaşık strateji oyunlarında insan seviyesinin ötesine geçebileceğini göstermiştir (Silver ve diğerleri, 2016). AlphaGo geleneksel yapay zeka yöntemlerinin aksine, hamle değerlendirmek için derin sinir ağları kullanmış ve kendi kendine denemeler yaparak yani oyun oynayarak milyonlarca oyun deneyimi kazanmıştır.

Robotik ve otonom alanlarında da pekiştirmeli öğrenme algoritmaları sistemlerin karmaşık görevleri öğrenmesinde ve doğru hareketin geliştirilmesinde yaygın olarak kullanılmaktadır (Kober ve diğerleri, 2013). Pekiştirmeli öğrenme yaklaşımı dört ayaklı robotların zorlu arazilerde yürümesi, robot kolların nesnelerle etkileşimde bulunması veya drone gibi cihazların engellerden kaçınarak uçması gibi görevlerde başarıyla uygulanmıştır. Boston Dynamics gibi şirketler pekiştirmeli öğrenme tabanlı yöntemler kullanarak robotların doğal ve çevik hareketler kazanmasını sağlamaktadır. Otonom araçlar alanında ise pekiştirmeli

öğrenme metodları araçların karmaşık trafik senaryolarında güvenli şekilde hareket etmesini sağlamak için kullanılmaktadır (Kendall ve diğerleri, 2019). Araçlar, simüle edilmiş ortamlarda milyonlarca kilometre sürüş deneyimi kazanarak, gerçek dünya koşullarına uyum sağlayabilmektedir.

Finans ve ticaret gibi alanlarda da pekiştirmeli öğrenme metodları, portföy yönetimi ve yüksek öneme sahip ticaret stratejilerinin optimizasyonunda kullanılmaktadır (Dixon ve diğerleri, 2017). Pekiştirmeli öğrenme tabanlı sistemler, piyasa koşullarına dinamik olarak uyum sağlayabilmekte ve geleneksel istatistiksel yöntemlere göre daha esnek çözümler sunabilmektedir. Pekiştirmeli öğrenme metodları sağlık sektöründe de kullanıcıların yaptıkları işlerin otomatik hale getirilmesine katkı sağlamaktadır. Örneğin kişiselleştirilmiş tedavi planlarının oluşturulması ve tıbbi teşhis sistemlerinin geliştirilmesinde pekiştirmeli öğrenme algoritmaları kullanılmıştır (Shortreed ve diğerleri, 2011). Bunlara ek olarak pekiştirmeli öğrenme metodları enerji yönetimi sistemlerinde binaların enerji tüketimini optimize etmek ve şebeke yönetimini iyileştirmek için başarıyla uygulanmıştır (Vázquez-Canteli ve Nagy, 2019).

Pekiştirmeli öğrenme yapısı genel anlamıyla çevreden aldığı bilgileri birtakım değerlendirmelerden geçirerek bir karar verir. Bu kararın doğruluğunu ölçer ve bir sonraki adımda kullanılan metodun dinamiklerine bağlı olarak yeni hareketine karar verir. Söz konusu sürekli döngü Şekil 2.3.'te verilmiştir.



Şekil 2.3. Pekiştirmeli öğrenme akışı

2.2.2. Derin pekiştirmeli öğrenme

Karmaşık ve yüksek boyutlu durum uzaylarında klasik değer tablosu tabanlı yaklaşımlar yetersiz kalmaktadır. Bu nedenle derin öğrenme ile pekiştirmeli öğrenmenin birleştiği Derin Pekiştirmeli Öğrenme (Deep Reinforcement Learning) yöntemleri geliştirilmiştir. Son yıllarda özellikle derin öğrenme tekniklerinin pekiştirmeli öğrenme yöntemi ile entegre edilmesi, daha yüksek hacimli ve sürekli uzaylarda da öğrenmenin mümkün hale gelmesini sağlamıştır. Bu sayede otonom araçlar, robot kontrol sistemleri, strateji oyunları ve finansal sistemler gibi birçok karmaşık uygulama alanında pekiştirmeli öğrenme algoritmaları başarılı sonuçlar vermeye başlamıştır (Kober ve diğerleri, 2013; Mnih ve diğerleri, 2015). Özellikle görsel veri işleme gerektiren uygulamalarda, derin sinir ağlarının özellik çıkarımı yeteneği sayesinde DRL sistemleri ham piksel verisinden doğrudan öğrenme yapabilmektedir (Mnih ve diğerleri, 2015).

Derin pekiştirmeli öğrenmenin matematiksel temeli, klasik Markov Karar Süreçleri (MDP) çerçevesine derin öğrenme bileşenlerinin eklenmesiyle genişletilmiştir. Matematik modeli 5 numaralı eşitlikte verilmiştir.

$$MDP_{DRL}=(S,A,P,R,\gamma,\phi_\theta) \tag{2.5}$$

Bu modelde kullanılan ϕ_θ derin sinir ağı tarafından sembolize edilen özellik dönüşüm fonksiyonunu temsil etmektedir (Sutton ve Barto, 1998).

Bunların en önemlilerinden biri Derin Q-Ağları (Deep Q-Networks – DQN), algoritmasıdır. DQN, durumları doğrudan sinir ağına vererek Q-değerlerini tahmin eder. Ayrıca deneyim tekrarı (experience replay) ve hedef ağlar (target networks) gibi teknikler, öğrenmenin istikrarlı bir biçimde gerçekleşmesini sağlar (Mnih ve diğerleri, 2015). Derin pekiştirmeli öğrenme, pekiştirmeli öğrenme ile derin sinir ağlarının birleşiminden oluşan ve son yıllarda büyük ilgi gören bir makine öğrenmesi alanıdır. Geleneksel pekiştirmeli öğrenme yöntemleri, düşük boyutlu durum uzaylarında etkili sonuçlar verebilirken, yüksek boyutlu ve yapılandırılmamış verilerle (örneğin görüntüler veya sensör verileri) başa çıkmakta zorlanır. Derin pekiştirmeli öğrenme, bu sorunu derin sinir ağlarının güçlü özellik çıkarım yetenekleriyle çözerek, karmaşık ortamlarda ölçeklenebilir ve etkili çözümler sunar.

Derin pekiştirmeli öğrenmenin en önemli avantajı, saf veriden otomatik olarak anlamlı özellikler çıkarabilmesidir. Geleneksel yöntemlerde, mühendislerin manuel olarak özellik mühendisliği yapması gerekirken, derin öğrenme modelleri bu süreci otomatikleştirir. Örneğin, bir otonom araç kamerasından gelen ham görüntüleri işlerken, derin bir evrişimli sinir ağı (CNN), görüntülerdeki şerit çizgilerini, diğer araçları ve yaya geçitlerini otomatik olarak tanıyabilir. Bu özellikler daha sonra pekiştirmeli öğrenme algoritmasına girdi olarak verilerek, aracın şerit değiştirme kararlarını optimize etmesi sağlanır.

Derin pekiştirmeli öğrenmenin en bilinen uygulamalarından biri, DeepMind tarafından geliştirilen ve Atari oyunlarında insan seviyesinde performans gösteren Deep Q-Network (DQN) algoritmasıdır. Bu çalışma, derin öğrenme ve pekiştirmeli öğrenmenin birlikte kullanılabileceğini göstermiş ve alanda bir dönüm noktası olmuştur. DQN, deneyim hafızası (experience replay) ve hedef ağ (target network) gibi tekniklerle öğrenme sürecini stabilize ederek, derin pekiştirmeli öğrenmenin pratikte başarılı olmasını sağlamıştır (Mnih ve diğerleri, 2015; Silver ve diğerleri, 2016).

Ancak, derin pekiştirmeli öğrenmenin de bazı zorlukları vardır. Öncelikle, bu yöntemlerin eğitimi için büyük miktarda veri ve hesaplama gücü gereklidir. Ayrıca, hiperparametre optimizasyonu ve öğrenme sürecinin kararlılığı gibi teknik zorluklar da mevcuttur. Örneğin, ödül fonksiyonunun yanlış tasarlanması, ajanın istenmeyen davranışlar öğrenmesine yol açabilir. Bu nedenle, derin pekiştirmeli öğrenme sistemlerinin tasarımında dikkatli bir mühendislik yaklaşımı gereklidir.

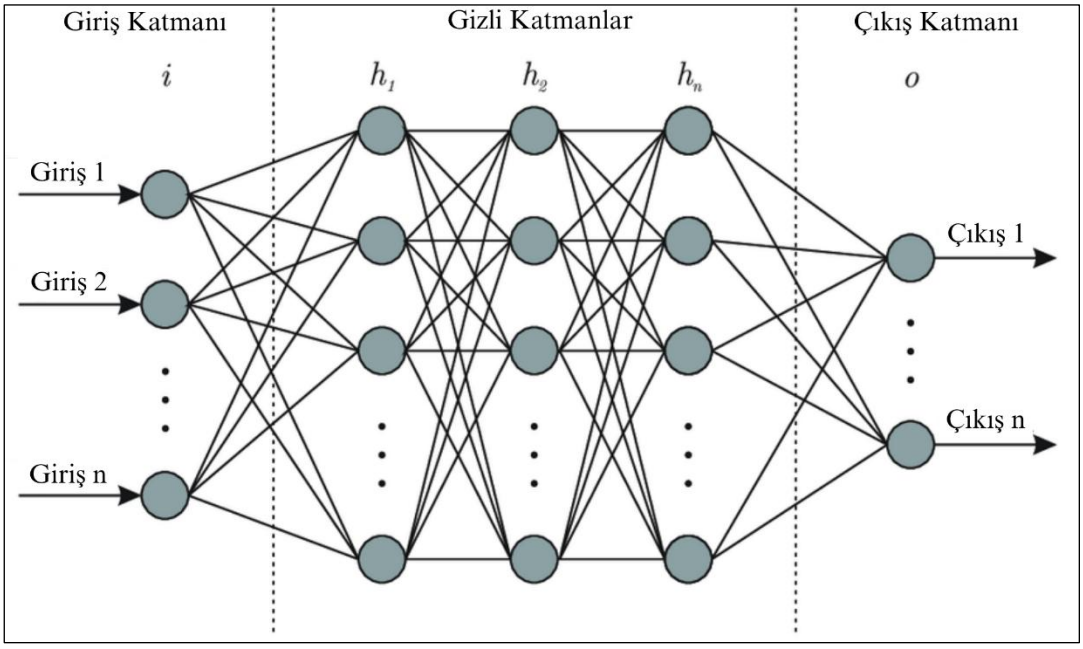
Derin pekiştirmeli öğrenme, otonom sürüş, robotik kontrol, doğal dil işleme ve finansal algoritma ticareti gibi birçok alanda başarıyla uygulanmaktadır. Özellikle otonom taşıtlarda şerit değiştirme, trafikte uyum sağlama ve park etme gibi karmaşık görevlerin çözümünde etkili sonuçlar vermektedir. Gelecekte, daha verimli ve kararlı algoritmaların geliştirilmesiyle, derin pekiştirmeli öğrenmenin uygulama alanlarının daha da genişlemesi beklenmektedir.

2.2.3. Yapay sinir ağları

Yapay sinir ağları son yıllarda otonom araç teknolojisinde kullanılan yöntemlerin en önemli bileşenlerinden biri haline gelmiştir. Yapay sinir ağı sistemleri insan beyninin yapısı ve

işleyişinden esinlenerek geliştirilen ve verilerden karmaşık davranışları öğrenebilen yapay zekâ modelleridir. Otonom araçlarda kullanılan derin öğrenme sistemleri, geleneksel tabanlı öğrenme ve tahmin yaklaşımlarının aksine, büyük miktarda veriden doğrudan öğrenme yeteneğine sahiptir. Bu özelliği ile trafiğin karmaşık ve öngörülemeyen doğasında etkili kararlar alabilmek bakımından kritik önem taşımaktadır (Bojarski ve diğerleri, 2016).

Bir yapay sinir ağı temel olarak üç katmandan oluşur. Bu katmanlar giriş katmanı, gizli katmanlar ve çıkış katmanıdır. Giriş katmanı, verinin en saf hali ile sisteme alındığı ilk katmandır. Başka bir deyişle sistemin sahip olduğu çevrenin istenen parametrelerini direkt olarak alır. Gizli katmanlar ise verinin işlendiği ve özelliklerin çıkarıldığı ara katmanlardır. Derin öğrenme olarak adlandırılan yöntemlerde, bu gizli katmanların sayısı oldukça fazla olabilir. Derin sinir ağı adı ile bilinen yapılarda minimum iki gizli katman bulunurken bu katman sayısı yüzlerce katmana varabilir (Goodfellow ve diğerleri, 2016). Çıkış katmanı ise ağın son tahminlerini veya eylemlerini ürettiği son katmandır. Her bir katmanda bulunan nöronlar, bir önceki katmandaki nöronlardan gelen bilgileri alır, belli bir ağırlığa sahip çarpan ile bir toplamını hesaplar ve bir aktivasyon fonksiyonundan geçirerek bir sonraki katmana iletir (Goodfellow ve diğerleri, 2016; Silver ve diğerleri, 2016). Yapay sinir ağı yapısı Şekil 2.4.'te görülmektedir.



Şekil 2.4. Yapay sinir ağı yapısı (Bre ve diğerleri, 2018)

Aktivasyon fonksiyonları yapay sinir ağlarının doğrusal olmayan parametreleri modelleyebilmesini sağlayan kritik bileşenlerdir. En yaygın kullanılan aktivasyon fonksiyonları arasında ReLU (Rectified Linear Unit), sigmoid ve tanh fonksiyonları örnek verilebilir. ReLU, özellikle derin sinir ağlarında yaygın olarak kullanılır çünkü hesaplama açısından verimli olmasının yanı sıra ölü nöron problemini minimize eder. Bu bağlamda en yaygın doğrusal olmayan aktivasyon fonksiyonlarından biri de tanh (tanjant hiperbolik) fonksiyonudur. Bu fonksiyon, giriş değerlerini -1 ile +1 arasında ölçeklendirir ve bu özelliği sayesinde simetrik çıktı dağılımına ihtiyaç duyulan derin sinir ağlarında kullanımı avantajlıdır (LeCun ve diğerleri, 2015).

Derin pekiştirmeli öğrenmede, yapay sinir ağları hem politikanın hem de değer fonksiyonlarının opitimizasayonu için kullanılır. Örneğin, bir otonom araç için şerit değiştirme kararı verirken, araç kamerasından alınan görüntüler bir evrişimli sinir ağı (Convolutional Neural Network- CNN) ile işlenerek anlamlı veriler ortaya çıkarılır. Bu veriler daha sonra politika ağına girdi olarak verilerek aracın hangi eylemi gerçekleştireceğine karar vermesi sağlanır. Buna benzer olarak değer ağı da bu özellikleri kullanarak belirli bir durumda seçilen belirli bir eylem sonucunda alınabilecek beklenen ödülü tahmin eder (Silver ve diğerleri, 2016).

Yapay sinir ağlarının öğrenme kapasitesi, çoğu zaman mimarinin derinliğiyle doğrudan ilişkilidir. Bu noktada, derin sinir ağları (Deep Neural Networks – DNN) kavramı ortaya çıkmaktadır. Derin sinir ağları, iki ya da daha fazla gizli katmandan oluşan çok katmanlı yapay sinir ağı mimarileridir. Derin öğrenme (deep learning) adı verilen yaklaşımın temelini oluşturan bu yapılar, özellikle büyük veri kümeleri ile çalışıldığında yüksek düzeyde modelleme yeteneği kazanmakta ve görüntü işleme, doğal dil işleme, otonom sistemler gibi karmaşık uygulama alanlarında oldukça başarılı performans göstermektedir (Goodfellow ve diğerleri, 2016).

Derin sinir ağları, klasik yapay sinir ağlarına göre daha fazla parametre içerdiği ve öğrenme sürecinde daha karmaşık örüntüleri keşfedebildiği için pekiştirmeli öğrenme yöntemleriyle entegre edildiğinde önemli avantajlar sağlamaktadır. Bu birleşim, derin pekiştirmeli öğrenme olarak adlandırılmakta ve son yıllarda otonom sürüşten robotik kontrol sistemlerine kadar geniş bir uygulama alanı bulmaktadır. Derin pekiştirmeli öğrenme algoritmalarının önemli bir örneği olan Deep Q-Network (DQN), Q-değer fonksiyonunu iyileştirmek için bir

derin sinir ağı kullanmakta ve klasik Q-learning algoritmasının büyük durum ve eylem uzaylarına uygulanabilirliğini mümkün kılmaktadır (Mnih ve diğerleri, 2015).

2.2.4. DQN algoritması

Pekiştirmeli öğrenme, yapay zekânın karar alma gerektiren pek çok alanında başarıyla uygulanmaktadır. Otonom araçlar en önde gelen uygulama alanlarından biridir. Şerit değiştirme, takip mesafesi ayarlama ve çarpışma önleme gibi görevlerde pekiştirmeli öğrenme tabanlı sistemler insan benzeri ve gerçekçi kararlar verebilmektedir (Kiran ve diğerleri, 2021).

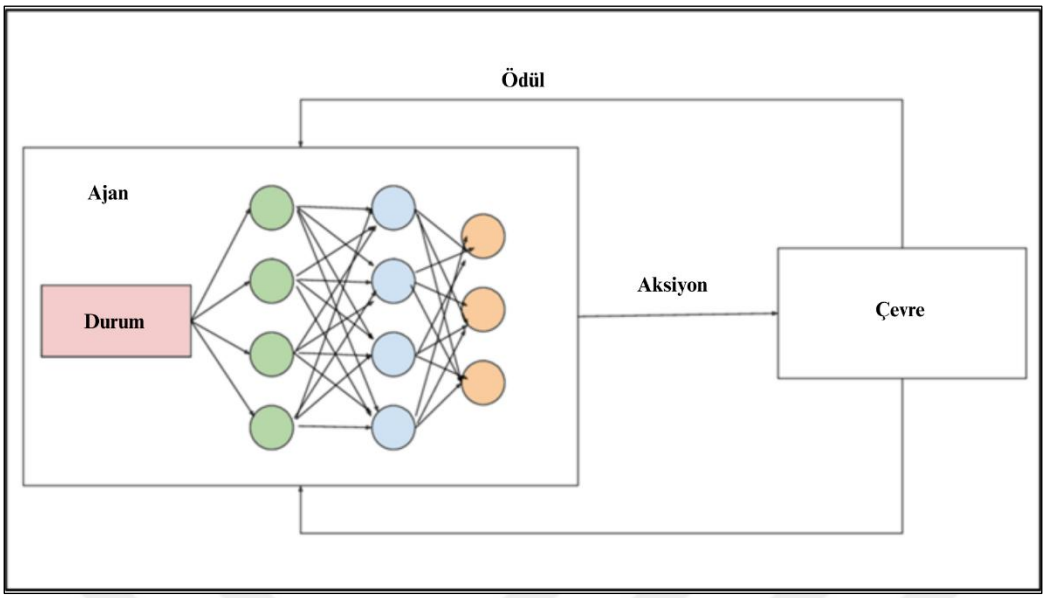
Derin Q-Ağları (DQN), pekiştirmeli öğrenme ile derin öğrenmenin başarılı bir şekilde birleştirildiği ilk yöntemlerden biridir ve derin pekiştirmeli öğrenme alanında önemli bir dönüm noktası olarak kabul edilir. DQN, 2015 yılında DeepMind araştırmacıları tarafından geliştirilmiş ve Atari 2600 oyunlarında insan seviyesinde performans göstermiştir. Bu başarı pekiştirmeli öğrenmenin yalnızca düşük boyutlu durum uzaylarında değil, yüksek boyutlu ve karmaşık girdilerle de başarılı olabileceğini göstermiştir. DQN'ın temel prensibi, geleneksel Q-öğrenme algoritmasını derin bir sinir ağı ile birleştirerek Q-değerlerinin daha etkili bir şekilde tahmin edilmesini sağlamaktır (Mnih ve diğerleri, 2015).

DQN'ın geleneksel Q-öğrenmeden en önemli farkı Q-değerlerinin tablo şeklinde saklanmak yerine bir sinir ağı tarafından saklanması ve işlenmesidir. Geleneksel Q-öğrenmede her durum-eylem çifti için bir Q-değeri saklanır ve bu değerler deneyimler sonucunda güncellenir. Ancak DQN yaklaşımı, durum uzayının büyük olduğu problemlerde (otonom araçlar için kamera görüntüleri gibi) uygulanabilir değildir. DQN ise durumu yani gözlem kümesini girdi olarak alan ve her bir eylem için Q-değerlerini tahmin eden bir sinir ağı kullanarak bu sorunu çözer. Bu sayede ağ daha önce hiç karşılaşmadığı durumlar için bile uygun Q-değerleri tahmin edebilir. Bu yaklaşımın matematiksel temeli, Bellman optimalite denklemine dayanmaktadır. DQN algoritmasını geleneksel Q-learning yaklaşımından farklı kılan özelliği, bu temel denklemi derin sinir ağları aracılığıyla uygularken getirdiği deneyim tekrarı (experience replay) ve hedef ağ (target network) mekanizmaları olmak üzere iki kritik yenilikte yatmaktadır. Deneyim tekrarı mekanizması ajanın geçmiş deneyimlerini matematiksel olarak $D = (s_t,a_t,r_t,s_{t+1})$ şeklinde ifade edilen bir bellekte saklayarak öğrenme sürecinde bu deneyimlerden rastgele örneklemeler yapmasını sağlamaktadır. Bu yaklaşım,

örnekler arasındaki tutarsızlığı azaltarak öğrenmenin kararlılığını önemli ölçüde artırmaktadır (Lin, 1992). Hedef ağ mekanizması ise Q-değerlerinin hesaplanmasında periyodik olarak güncellenen ayrı bir ağ kullanarak öğrenme sürecindeki değişken yapıyı minimize etmeyi amaçlamaktadır (Mnih ve diğerleri, 2015).

DQN algoritmasının başarılı sonuçlar getirmesi ve yaygınlaşması ile beraber araştırmacılar bu temel yaklaşımın çeşitli sınırlamalarını aşmak için önemli geliştirmeler yapmışlardır. 2016 yılında yapılan bir çalışmada, araştırmacılar tarafından önerilen Çift DQN (Double DQN), Q-değerlerinin sistematik olarak fazla tahmin edilmesi (over estimation) problemini çözmek amacıyla geliştirilmiştir. Bu yöntem, hedef değerlerin hesaplanmasında farklı bir strateji izleyerek daha kararlı öğrenme sağlamaktadır (Van Hasselt ve diğerleri, 2016). Bir diğer önemli geliştirme olan Dueling DQN mimarisi ise Q-değerlerinin durum değeri (state value) ve avantaj fonksiyonu (advantage function) olarak iki bileşene ayrıştırılması prensibine dayanmaktadır (Wang ve diğerleri, 2016).

DQN ve bağlı metodlarının başarısı özellikle Atari 2600 oyunları üzerinde yapılan kapsamlı testlerle kanıtlanmıştır. Space Invaders, Breakout ve Pong gibi klasik oyunlarda insan seviyesinde performans sergileyen bu algoritmalar, yüksek boyutlu görsel girdileri işleyebilme ve genelleme yapabilme yetenekleriyle ön plana çıkmıştır (Mnih ve diğerleri, 2015). Ancak DQN algoritmasının bazı önemli sınırlamaları bulunmaktadır. DQN yöntemi yalnızca ayrık eylem uzaylarında kullanılabilmektedir. Başka bir deyişle sürekli eylem uzaylarını desteklememektedir. Ayrıca, örnek verimliliği açısından oldukça maliyeti yüksek bir algoritma olup, yüksek sayıda deneme gerektirmektedir. Q-değerlerinin fazla tahmin edilmesi (overestimation bias) de diğer bir önemli sorun olarak karşımıza çıkmaktadır (Van Hasselt ve diğerleri, 2016). Bu sınırlamaları aşmak için geliştirilen yöntemler arasında Dağıtımsal DQN (C51), Noisy Nets ve Rainbow DQN gibi yaklaşımlar sayılabilir (Hessel ve diğerleri, 2018). Bu gelişmeler, DQN'nin günümüzde hala pekiştirmeli öğrenme araştırmalarında temel bir referans noktası olarak kabul edilmesini sağlamaktadır. Pekiştirmeli öğrenme döngüsünün derin öğrenme ile birleşimi olan DQN algoritması için oluşturulmuş hali Şekil 2.5.’te görülmektedir.



Şekil 2.5. DQN algoritması şeması (Alam, 2023)

DQN'ın otonom sürüş sistemlerine uygulanması, özellikle şerit değiştirme ve trafikte seyir gibi karmaşık karar verme süreçlerinde etkili sonuçlar vermiştir. Örneğin, bir otonom araç kamerasından gelen görüntüleri işlemek için evrişimli sinir ağlarını kullanabilir ve bu görüntülerden çıkarılan özellikleri DQN algoritmasına girdi olarak verebilir. Araç, farklı durumlarda (örneğin, yoğun trafik, yağmurlu hava) hangi eylemleri gerçekleştireceğini (sola geçiş yap, hızını artır, fren yap) bu yöntemle öğrenebilir. Ancak DQN'ın bazı sınırlamaları da bulunmaktadır. En önemli sınırlama, ayrık eylem uzaylarıyla çalışmasıdır. Yani, eylemler sonlu ve ayrık bir küme içermelidir (örneğin, sola dön, sağa dön, düz devam et). Sürekli eylem uzaylarında (örneğin, direksiyon açısının -30 derece ile +30 derece arasında herhangi bir değer alabilmesi) ise DQN doğrudan uygulanamaz. Bu tür problemler için DDPG gibi alternatif yöntemler geliştirilmiştir.

Otonom şerit değiştirme manevrası için süreç karar verme, rota planlama ve rota takibi olmak üzere üç adımda gerçekleşir. Bu adımlardan karar verme yapısı, çalışmalarda çoğunlukla yapay zekâ tabanlı algoritmalarla tasarlanarak karar verme davranışının insan benzerliğinin maksimize edilmesi amaçlanmaktadır. Bununla beraber rota planlama ve rota takibi adımları için de yapay zekâ tabanlı sistemler kullanılsa da deterministik ve optimizasyon tabanlı algoritmalar da sıklıkla görülmektedir.

2.3. Otonom Şerit Değiştirme Rota Planlama Yöntemleri

Şerit değiştirme rotasının planlanması, otonom aracın belirli bir zaman aralığında izleyeceği yolun belirlenmesini kapsar. Bununla beraber rota planlama yöntemleri genellikle parametrik eğriler, optimizasyon tabanlı fonksiyonlar ve öğrenme temelli modeller olarak üç ana grupta incelenebilir. En yaygın kullanılan deterministik yöntemlerden biri polinom tabanlı rota planlamasıdır. Bu yöntemde şerit değiştirme rotası genellikle üçüncü veya beşinci dereceden polinomlar ile temsil edilir. Polinom fonksiyonlar başlangıç ve bitiş koşullarına uygun şekilde belirlenerek aracın yumuşak bir geçiş yapmasını sağlar (Werling ve diğerleri, 2010). Rota planlamada en sık kullanılan polinom tabanlı yöntemlerden biri de Quintic polinomlardır. Quintic polinomlar özellikle robotik ve otonom sürüş sistemlerinde, başlangıç ve bitiş noktaları arasındaki süreklilik özelliğine sahip rotalar oluşturmak amacıyla kullanılan etkili bir yöntemdir. Beşinci dereceden bir polinom yapısı, başlangıç ve bitiş noktalarında pozisyon, hız ve ivme değerlerinin istenen şekilde tanımlanmasına imkân sağlar. Bu yöntem, altı bilinmeyenli denklem sisteminin çözülmesiyle polinom katsayılarının belirlenmesine dayanır (Zhao ve diğerleri, 2017). Quintic polinomların matematiksel ifadesi aşağıdaki eşitlikte verilmiştir.

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{2.6}$$

Bu eşitlikteki t değişkeni zamanı ifade eder. $a_{1\to n}$ ifadesi ise polinom katsayılarıdır. Bu katsayılar, rota planlama sırasında belirlenen başlangıç ve bitiş pozisyonu, hızı ve ivmesi gibi sınır koşulları kullanılarak hesaplanır. Yani bu katsayılar rotanın şekline karar verir. Farklı sınır koşulları için bu katsayılar değişir (Zhao ve diğerleri, 2017). Ancak polinomlar, yol eğriliği ve dinamik engel durumu gibi faktörleri doğrudan değerlendirmediğinden, daha esnek yöntemler geliştirilmiştir.

Alternatif olarak, Bézier eğrileri ve B-spline fonksiyonları, rota planlamada daha fazla kontrol noktası sunarak esnek geçişler sağlamaktadır. Bu yöntemler, özellikle karmaşık çevresel koşullarda daha uyarlanabilir yörünge profilleri oluşturmada avantaj sağlar (Ziegler ve diğerleri, 2014). Bununla birlikte bu geometrik yöntemler, araç dinamiklerini açıkça modele dahil etmedikleri için gerçek zamanlı güvenlik değerlendirmeleriyle birlikte çalıştırılmaları gerekmektedir.

Güncel çalışmalarda sıklıkla tercih edilen bir diğer yöntem ise optimizasyon tabanlı planlama yaklaşımlarıdır. Bu yöntemlerde şerit değişim rotası, bir maliyet fonksiyonu (cost function) içerisinde optimize edilerek belirlenir. Maliyet fonksiyonu genellikle konfor (ivme, jerk), güvenlik (engel mesafesi) ve yol uygunluğu gibi parametreleri içerir. Bu bağlamda, kuadratik programlama (Quadratic Programming- QP), doğrusal programlama (Linear Programming- LP) ve model öngörülü kontrol (Model Predictive Control- MPC) gibi optimizasyon teknikleri tercih edilmektedir (Falcone ve diğerleri, 2007).

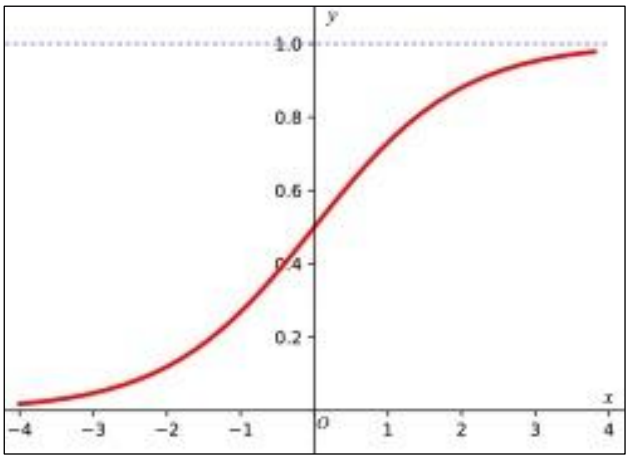
Son yıllarda ise makine öğrenmesi ve pekiştirmeli öğrenme tabanlı rota planlayıcıları, özellikle bilinmeyen çevresel koşullara adaptasyon yetenekleri sayesinde dikkat çekmektedir. Bu yöntemlerde ajan, ödül fonksiyonuna dayalı olarak birçok deneme yaparak optimal rotayı zaman içinde öğrenir. Bununla birlikte, bu yöntemlerin eğitim süreci oldukça karmaşık ve zaman alıcıdır (Sallab ve diğerleri, 2017).

Her yöntemin avantajları ve sınırlılıkları bulunsa da şerit değiştirme gibi karmaşık manevralarda başarı sağlamak için aracın dinamiğine uygun, çevresel faktörleri hesaba katan ve kontrol sistemleriyle entegre çalışabilecek bir rota planlama algoritmasına ihtiyaç vardır. Bu bağlamda sigmoid tabanlı yaklaşım hem pratik uygulama kolaylığı hem de yüksek konfor faktörleri sebebiyle bu çalışmada tercih edilmiştir.

Sigmoid tabanlı rota planlama yöntemi hem matematiksel sadeliği hem de yumuşak geçişler üretme kapasitesi nedeniyle farklı odaklarda ilerleyen birçok probleme ve çalışmaya çözüm sunabilecek niteliktedir. Sigmoid fonksiyonu 6 numaralı eşitlik ile ifade edilir.

$$\sigma = \frac{1}{1+e^{-a(x-x_0)}} \tag{2.7}$$

Şerit değişiminde sigmoid kullanımı, aracın yumuşak bir yörüngeyle hedef şeride geçmesini sağlarken, ani yönelme değişimlerinden kaynaklanabilecek konfor kayıplarını da minimize eder (Han ve Moraga, 1995).



Şekil 2.6. Sigmoid fonksiyon grafiği (Chen ve diğerleri, 2024)

Sigmoid fonksiyonu, "S" şeklindeki karakteristik eğrisiyle matematikte ve mühendislikte yaygın olarak kullanılan temel bir aktivasyon fonksiyonudur (Bkz. Şekil 2.6.). Girdi değerlerini 0 ile 1 arasında ölçekleyerek yumuşak geçişler sağlar. Yapısı gereği türevlenebilir olması Sigmoid fonksiyonunu özellikle kontrol sistemleri ve yapay sinir ağları gibi alanlarda kullanışlı kılmaktadır (Han ve Moraga, 1995).

Sigmoid fonksiyonunun otonom araç uygulamalarında kullanımı, temel olarak üç önemli avantaja dayanmaktadır. Bu avantajlardan ilki süreklilik sağlayarak yumuşak ve doğal geçişlere olanak tanımasıdır. İkincisi, parametrik yapısı sayesinde manevra karakteristiğinin kolayca ayarlanabilmesidir. Üçüncü ve en önemli faktör ise insan sürüş davranışına matematiksel ve fiziksel olarak çok benzer bir profil oluşturabilmesidir. Araştırmalar, insan sürücülerin şerit değiştirirken doğal hareketleri ile sigmoide benzer bir yol izlediğini göstermektedir (Salvucci ve Liu, 2002). Otonom şerit değiştirme probleminde sigmoid fonksiyonunun kullanımı, geleneksel polinom (quintic ve quadratic) tabanlı yaklaşımlara göre birçok üstünlük sunar. Fonksiyonun eğim parametresi (k), manevranın agresifliğini doğrudan kontrol edebilir. Örneğin yüksek hızlarda daha düşük k değerleri kullanılarak uzun ve yumuşak geçişler sağlanırken, düşük hızlarda daha yüksek k değerleriyle daha keskin manevralar mümkün olmaktadır (Huang ve diğerleri, 2019). Bu parametrik esneklik, farklı yol ve trafik koşullarına adaptasyonu kolaylaştırmaktadır.

2.4. Otonom Şerit Değiştirme Rota Takibi Yöntemleri

Otonom taşıtların güvenli ve başarılı bir şekilde manevralarını gerçekleştirebilmesi, yalnızca izlenecek yolun doğru şekilde planlanmasıyla sınırlı değildir. Araç, önceden oluşturulan referans rotayı gerçek zamanlı olarak ne kadar hassas bir biçimde takip edebiliyorsa, manevranın güvenilirliği ve yolcu konforu da o denli artar. Bu noktada devreye giren rota takip sistemleri, aracın belirli bir yörüngeye olan konum ve yönelim farklarını dikkate alarak uygun direksiyon ve hız komutlarını üretir. Özellikle şerit değiştirme gibi hassas manevralarda, takip doğruluğu yalnızca konfor açısından değil, aynı zamanda çarpışma risklerini azaltmak adına da kritik önem taşımaktadır. Bu bağlamda rota takip algoritmaları, referans yörüngeye olan hata büyüklüklerini en aza indirmeyi amaçlayan çeşitli kontrol stratejilerine dayanır.

Rota takibi için geliştirilen yöntemler, genellikle geometrik tabanlı kontrol, model tabanlı kontrol ve öğrenmeye dayalı yaklaşımlar olarak sınıflandırılabilir. Bu yöntemlerin ortak hedefi, aracın referans rotaya göre pozisyon hatasını ve yönelim farkını minimize ederek sürüş doğruluğunu artırmaktır.

En yaygın kullanılan geometrik yöntemlerden biri Pure Pursuit (saf takip) algoritmasıdır. Bu yöntemde araç, belirli bir ön görüş mesafesi kadar ilerdeki bir hedef noktaya yönelerek yörüngeyi takip etmeye çalışır. Basitliği ve gerçek zamanlı uygulanabilirliği sayesinde birçok otonom sistemde kullanılmış olsa da, düşük hızlarda agresif direksiyon açıları üretmesi veya yüksek hızda sapma eğilimi gibi dezavantajları vardır (Coulter, 1992).

Rota takibinde öne çıkan bir diğer yöntem ise Stanley kontrol algoritmasıdır. Bu yöntem, aracın yönelim hatası ve referans yola olan çapraz iz hatasını dikkate alarak direksiyon açısını hesaplar. Stanford'un otonom aracı Stanley ile ünlenen bu algoritma, özellikle düşük hızlarda kararlı sonuçlar üretmesiyle bilinir. Yönelim açısı ile çapraz iz hatası arasında bir dengeleme sağlayan bu yapı, sade bir matematiksel formülasyonla gerçek zamanlı sistemlere kolayca entegre edilebilir (Thrun ve diğerleri, 2006). Tez kapsamında da kullanılan bu yöntem, sigmoid tabanlı rota planlama ile birlikte çalışarak yumuşak ve kararlı şerit değiştirme manevraları sunmayı hedeflemektedir.

Geometrik yöntemlerin sınırlı durumlara uygunluğu nedeniyle, daha gelişmiş kontrol gereksinimleri için model tabanlı kontrol (model-based control) yaklaşımları tercih edilmektedir. Bu bağlamda öne çıkan yöntemlerden biri Model Predictive Control (MPC)'dir. MPC hem taşıtın dinamiklerini hem de referans yörüngeyi dikkate alarak belirli bir zaman ufku içinde kontrol girdilerini optimize eder. Bu yapı sayesinde sistem, ileriyi öngörebilir, sınırlamaları hesaba katabilir ve kontrol sinyallerini yumuşak biçimde üretir (Falcone ve diğerleri, 2007). Özellikle yüksek hızda sürüşlerde ya da karmaşık trafik senaryolarında MPC, araç davranışını kararlı ve güvenli kılmak açısından önemli avantajlar sunar.

Son yıllarda rota takibi için makine öğrenmesi ve pekiştirmeli öğrenme (RL) gibi veri temelli yaklaşımlar da kullanılmaya başlanmıştır. Bu sistemler, çevreyle etkileşim üzerinden hata geri bildirimleri alarak zaman içinde en uygun kontrol stratejisini öğrenir. Örneğin, bir derin pekiştirmeli öğrenme (DRL) ajanı, aracın yanal ve boylamsal sapmalarını minimize edecek şekilde kendini eğiterek dinamik ortamlarda yüksek başarı oranlarına ulaşabilir (Kendall ve diğerleri, 2019). Ancak bu yöntemlerin eğitim süreci hesaplama açısından maliyetlidir ve simülasyon ortamında iyi sonuçlar vermesine rağmen gerçek dünya senaryolarında güvenlik garantisi sağlamak için daha fazla çalışmaya ihtiyaç duyulmaktadır.

Rota takip yöntemleri ile ilgili çalışmalar incelendiğinde otonom algoritmaların başarısı, yalnızca hata büyüklüklerinin azaltılmasıyla değil, aynı zamanda sürüş konforu, tepki hızı ve sistem kararlılığıyla da ölçülmektedir. Bu çalışmada tercih edilen Stanley kontrol algoritması, sigmoid tabanlı rota planlama yöntemiyle uyum içinde çalışarak hem algoritmik sadelik hem de uygulama verimliliği sağlamaktadır.

Otonom taşıtların gelişimiyle birlikte güvenli, konforlu ve verimli bir sürüş elde etmek amacıyla çok sayıda kontrol algoritması geliştirilmiştir. Bu algoritmalardan biri olan Stanley kontrol yöntemi, özellikle yanal kontrol (lateral control) görevlerinde öne çıkan basitliği, kararlılığı ve gerçek zamanlı uygulanabilirliği ile oldukça uygulanabilir bir yöntemdir. Stanford Üniversitesi tarafından geliştirilen ve 2005 DARPA Grand Challenge'ı kazanan otonom aracın kontrol algoritması olarak tanınan Stanley kontrolörü, özellikle düşük hızlı uygulamalarda ve hafif araç koşullarında etkili sonuçlar vermektedir (Thrun ve diğerleri, 2006).

Stanley kontrol algoritmasının temel amacı, aracın mevcut pozisyonunu hedef rota (reference trajectory) ile hizalayarak sapmayı minimize etmektir. Bu doğrultuda kontrol stratejisi, yönelim hatası (heading error) ve çapraz yol hatası (cross-track error) olmak üzere iki temel hata bileşeni üzerinden tanımlanır. Yönelim hatası, aracın boylamasına ekseni ile referans yol üzerindeki teğet doğrultu arasındaki açı farkını tanımlar. Çapraz iz hatası ise aracın referans yoluna olan dik mesafesidir. Bu iki hata bileşeni Stanley algoritmasının kontrol girdisi olan direksiyon açısını belirlemede kullanılır. Stanley denetleyicisi tarafından belirlenen yönlendirme açısı 7 numaralı eşitlik ile hesaplanır:

$$\delta = \theta_e + \arctan\frac{k.e_{ct}}{v} \tag{2.8}$$

Bu denklemde θ_e yönelim hatasını, e_{ct} çapraz yol hatasını (cross-track error), pozitif bir kazanç parametresini, ise aracın anlık hızını temsil etmektedir. Bu formül, aracın referans yoluna hizalanmasını sağlayacak şekilde iki hata türünü birleştirir. İlk terim, aracın yönelimi ile yolun yönü arasındaki farkı düzeltirken, ikinci terim ise çapraz mesafeyi azaltmayı hedefler. Arctanjant fonksiyonunun kullanımı, kontrol girişinin sürekliliğini ve kontrol hassasiyetini arttırır. Hızın payda olarak yer alması ise özellikle düşük hızlarda sistemin daha agresif, yüksek hızlarda ise daha yumuşak tepkiler vermesini sağlar. Bu hız duyarlılığı Stanley kontrol algoritmasını dinamik araç uygulamaları için uygun hale getirir (Paden ve diğerleri, 2016).

Yanal (Lateral) Stanley kontrolörü, temel Stanley algoritmasına benzer prensiplere dayanmakla birlikte özellikle yanal konumlandırma görevlerine odaklanır. Bu bağlamda, longitudinal (boylamsal) kontrol ile entegre çalışarak aracın belirli bir yörüngede sadece hız değil aynı zamanda doğru konumda ilerlemesini sağlar. Yanal Stanley kontrolü, şerit takip, kavşak dönüşleri ve engelden kaçınma gibi görevlerde kullanılarak taşıtın şeritteki pozisyonunu korumasına yardımcı olur. Bu tür uygulamalarda sistem, yalnızca merkez çizgiye olan mesafeyi değil, aynı zamanda yörünge eğrisinin yerel eğriliğini ve aracın anlık yönelimini de göz önüne alır (Ziegler ve diğerleri, 2014).

Yanal Stanley kontrol algoritmasının performansı genellikle takip edilen yolun eğriliğine, taşıt dinamiklerine ve sensör doğruluğuna bağlıdır. Özellikle eğimli veya bozuk zeminlerde, yanlış çapraz yol hatası tahmini sistemin kararlılığını olumsuz etkileyebilir. Bu nedenle lidar, GPS, IMU gibi çoklu sensör verileriyle zenginleştirilmiş bir algı sistemi Stanley kontrol

algoritmasının doğruluğunu artırabilir. Bunun yanı sıra, Stanley kontrolörü genellikle PID veya MPC (Model Predictive Control) gibi boylamsal kontrol yöntemleriyle birlikte kullanılır ve bu kontrolcülerin koordinasyonu aracın genel sürüş kalitesini belirleyici hâle getirir (Falcone ve diğerleri, 2007).

Stanley kontrol algoritması, sahip olduğu yapısal basitliğe rağmen otonom taşıtlarda yanal kontrol görevlerinde oldukça etkili ve kararlı bir çözümdür. Araç-yol hizalanmasını sağlamak için yönelim ve çapraz iz hatalarını temel alan bu algoritma, gerçek zamanlı uygulamalar için uygundur ve çeşitli kontrol yöntemleriyle kolaylıkla entegre edilebilir.

2.5. Literatür Taraması

Otonom sürüş teknolojileri, son yıllarda hızla gelişen yapay zekâ ve kontrol sistemleriyle birlikte, özellikle şerit değiştirme gibi karmaşık sürüş manevralarında yeni yaklaşımların geliştirilmesini zorunlu kılmıştır. Bu bağlamda, otonom taşıtların güvenli, konforlu ve etkin bir şekilde şerit değiştirebilmesi için literatürde çok sayıda yöntem önerilmiş; bu yöntemler karar verme, hareket planlama ve kontrol katmanları olmak üzere farklı aşamalarda sınıflandırılmıştır. Yapılan çalışmaların önemli bir kısmı, klasik kural tabanlı modellere dayansa da değişken trafik koşullarında esnek kararlar alabilme ihtiyacı bu yaklaşımların yetersiz kaldığını ortaya koymuştur. Bu doğrultuda son yıllarda makine öğrenmesi, pekiştirmeli öğrenme, bulanık mantık, SVM (Support Vector Machine) ve MPC (Model Predictive Control) gibi yöntemlerin otonom şerit değiştirme problemlerine uygulanması ile daha uyarlanabilir, gerçekçi ve güvenli sistemler geliştirilmeye başlanmıştır. Literatürde yer alan bu çalışmalar, algoritmaların performansını değerlendirmek amacıyla hem simülasyon hem de gerçek sürüş verileri kullanılarak test edilmiş ve yeni modellerin klasik yaklaşımlara göre daha yüksek başarı oranlarına sahip olduğu raporlanmıştır. Bu bölümde, otonom taşıtlarda şerit değiştirme manevrasına yönelik geliştirilen güncel algoritmalar detaylı şekilde incelenmekte ve bu çalışmaların temel motivasyonları, yöntemsel yaklaşımları ve elde edilen sonuçları karşılaştırmalı olarak sunulmaktadır.

Li ve diğerleri, tarafından gerçekleştirilen çalışmada, bağlantılı otonom araçlar (CAV) için çoklu araç trafiğinde iş birliğine dayalı şerit değiştirme manevralarının planlanması için yeni bir model önerilmiştir. Model, zorunlu şerit değişim hareketlerinin bulunduğu yoğun trafik şartlarında hem güvenliği hem de verimliliği optimize etmeyi hedeflemektedir. Önerilen

sistem araçların gruplara ayrılması ve grup bazlı hareket planlaması olarak iki temel adıma dayanmaktadır. Araçların konum ve hız gibi parametreler göz önünde bulundurularak gruplandırılma yapılmaktadır. Her grup içerisindeki araçlar için şerit değiştirme talepleri dikkate alınarak çok değişkenli ve yüksek dereceli polinomlar yardımıyla yumuşak ve sürekli hızlanma-eğrileri üretilmektedir. Model, güvenlik ve konforu artırmak amacıyla hızlanma, yavaşlama ve "jerk" (ivmenin değişim hızı) gibi dinamik sınırlamaları göz önünde bulundurarak bir optimizasyon problemi şeklinde formüle edilmiştir. Model, çeşitli simülasyon testleriyle değerlendirilmiş ve sonuçlar, geliştirilen iş birlikçi planlama yönteminin hem güvenliği sağladığını hem de şerit değiştirme başarısını artırdığını göstermiştir (Li ve diğerleri, 2020).

Wu ve Yang tarafından yapılan çalışmada şerit değiştirme davranışının daha gerçekçi bir şekilde modellenmesine odaklanılmıştır. Mevcut daraltılmış trafik simülasyonu modellerinde, özellikle de şerit değiştirme esnasında, hedef şeritteki takip eden araçların sabit hızda ilerlediği varsayımı yapılmaktadır. Ancak önerilen çalışmada bu varsayımın gerçek trafik durumunu yansıtmadığı ifade edilmiştir. Gerçek sürüş şartlarında bir araç önüne başka bir aracın geçeceğini fark ettiğinde genellikle hızını düşürerek tepki verir. Bu bakış açısından hareketle çalışmada araç takip davranışını da dikkate alan yeni bir şerit değiştirme modeli geliştirilmiştir. Modelde, şerit değiştirmek isteyen aracın hızlanarak geçiş yaptığı ve hedef şeritteki araçların bu duruma çeşitli tepkiler verdiği bir senaryo tanımlanmıştır. Modelin kurulumu üç adımdan oluşmaktadır: Şerit değiştiren aracın mevcut şeridindeki takip eden araçla güvenli mesafeyi koruması, hedef şeritte öndeki araçla güvenli mesafeye ulaşması ve hedef şeritteki arkadaki araçla güvenli mesafenin korunması. Her bir adım için ayrıntılı kinematik denklemlerle güvenli şerit değiştirme koşulları türetilmiştir. Araçların hız, pozisyon ve ivme gibi parametreleri dikkate alınarak geliştirilen model, MATLAB simülasyonlarıyla test edilmiştir. Sonuçlar, modelin gerçek trafik durumlarına uygun ve uygulanabilir olduğunu göstermiştir (Xiaorui ve Hongxu, 2013).

Liu ve diğerleri, gerçekleştirdikleri çalışmada otonom araçların şerit değiştirme kararlarını kendi başına alabilmesine yönelik Support Vector Machine (SVM) tabanlı bir karar verme modeli geliştirmiştir. Bu çalışmanın özgün tarafı şerit değişim kararının nasıl ve ne zaman alınması gerektiğine dair sistematik bir yaklaşım sunmasıdır. Çalışmada karar süreci fayda, güvenlik ve tolerans olmak üzere üç temel faktör üzerinden analiz edilmiştir. Modelin karar verme algoritması tanımlanan parametrelerin lineer olmayan kombinasyonuna

dayanmaktadır. Bu nedenle klasik kural tabanlı yaklaşımların yetersiz kalacağı belirtilerek SVM algoritmasının Gaussian fonksiyonu kullanılarak uygulanması tercih edilmiştir. Ayrıca, modelin parametrelerinin en iyi hale getirilmesi için Bayes optimizasyon yöntemi kullanılmıştır. Eğitim verisi olarak ABD Ulaştırma Bakanlığı tarafından yayımlanan NGSIM adlı gerçek sürüş verisi kullanılmıştır. Araçların şerit değiştirme eğilimleri bu veri setinden yararlanılarak tanımlanmıştır. SVM modeli, kurallara dayalı geleneksel bir modelle karşılaştırmalı olarak test edilmiştir. Sonuçlar SVM tabanlı modelin doğruluk açısından daha başarılı olduğunu ve özellikle sürücü alışkanlıklarını dikkate alabilme kabiliyetiyle daha gerçekçi kararlar verdiğini göstermiştir. Ayrıca modelin gerçek araç testlerinde de uygulanabilirliği denenmiş ve başarılı sonuçlar elde edilmiştir (Liu ve diğerleri, 2019).

Huang, Naghdy ve Du önerdikleri modelde otonom araçların güvenli şerit değiştirme manevralarını gerçekleştirmesi için Model Predictive Control (MPC) tabanlı bir algoritma geliştirilmiştir. Sistem çarpışma riski taşıyan durumlarda aracın önceden planlanmış bir güzergâhı takip ederek manevra yapmasını sağlamak üzerine kurulmuştur. Rota planlaması, konveks optimizasyon kullanılarak gerçekleştirilmiş, taşıt dinamiği sekiz serbestlik dereceli model ve Dugoff lastik modeli ile ifade edilmiştir. Önerilen modelde MPC, aracın direksiyon açılarını ve tekerlek torklarını kontrol ederek belirlenen yolu takip etmesini sağlar. Çalışmada özellikle karmaşık trafik koşullarında güvenli şerit değiştirme için uygulanabilir bir çözüm geliştirilmesine odaklanılması ile beraber simülasyonlar, sistemin çarpışmalardan kaçınmada ve stabil sürüş sağlamada etkili olduğu görülmüştür (Huang ve diğerleri, 2016).

Naranjo ve diğerleri tarafından yapılan çalışmada otonom araçların sollama manevrasını güvenli ve gerçek sürücülü harekete yakın bir şekilde gerçekleştirmesi amacıyla bulanık mantık (fuzzy logic) tabanlı bir kontrol sistemi geliştirilmiştir. Özellikle çift yönlü yollarda yapılan sollamaların karmaşıklığına odaklanılmış, şerit değiştirme, yol takibi ve geri dönüş manevraları ele alınmıştır. Sistem iki seviyeli bir mimariye sahiptir. Alt düzeyde, biri şerit takibi diğeri şerit değiştirme için olmak üzere iki ayrı bulanık mantık direksiyon denetleyicisi yer almaktadır. Üst düzeydeki copilot modülü, sollama kararını verir, rotayı belirler ve uygun alt düzey denetleyicileri devreye sokar. Hız kontrolü ise otonom olarak ya sabit hızla devam edilmekte ya da ön araçla güvenli mesafe korunmaktadır. Oluşturulan algoritmaya göre sollama yapılabilmesi için yolun yeterince düz, sol şeridin boş ve aracın sollamayı tamamlayabilecek hızda olması gibi koşulların sağlanması gerekmektedir. Gerçekleştirilen saha testlerinde, geliştirilen mimarinin başarılı sonuçlar verdiği ve insan

sürücülere benzer tepkiler ürettiği gösterilmiştir. Sonuç olarak, bulanık mantık temelli bu yaklaşım, otonom sistemlerin karmaşık sollama manevralarını daha doğal ve güvenli biçimde yapmasına olanak tanımaktadır (Naranjo ve diğerleri, 2008).

Li ve diğerleri 2022 yılında yaptıkları çalışmada otonom araçlar için insan benzeri hareket planlaması geliştirmiştir. Bu planlama çevredeki araçların sürüş dinamiklerini gözlemleyerek ve hareketlerini öngörerek güvenliği ve sürüş konforunu bir arada sağlamayı amaçlamaktadır. Çalışmada temel olarak çevre araçlarının hareket tahmini, maliyet fonksiyonuna dayalı karar alma ve sürücünün hedef hızına göre hız planlaması olarak üç bileşene odaklanılmıştır. Güzergâh planlaması beşinci dereceden polinomlarla gerçekleştirilmiş ve maliyet fonksiyonunda güvenlik, süreklilik, yumuşak hareket geçişleri gibi kriterler yer almıştır. Araç ve çevresindeki nesneler için çarpışma olasılığı Monte Carlo yöntemiyle hesaplanmıştır. Sürücü karakterine göre (temkinli/agresif) farklı risk eşikleri belirlenerek, kararlar bu eşiklere göre alınmıştır. Çevre araçlarının gelecekteki konumlarını tahmin edebilmek için yapısal LSTM tabanlı bir derin öğrenme ağı geliştirilmiştir. Bu tahminler, gerçek sürüş verileri (NGSIM) kullanılarak eğitilmiş ve araçlar arası etkileşim dikkate alınmıştır. Ayrıca, araçların hız planlaması da LSTM ile gerçekleştirilerek, sürücü alışkanlıklarını taklit eden insana özgü bir hız profili elde edilmiştir. Yapılan analizler algoritmanın farklı sürüş tercihlerine adapte olabileceğini göstermiştir ve önerilen yöntemin klasik deterministik planlamalara kıyasla daha esnek ve insan benzeri bir otonom sürüş davranışı sunduğu görülmüştür (Li ve diğerleri, 2022).

Yavas, Kumbasar ve Ure yaptıkları çalışmada bir derin pekiştirmeli öğrenme yöntemi olan Rainbow DQN'i kullanarak, güvenlik odaklı bir ödüllendirme algoritması ile eğitimin etkinliğini artırmayı amaçlamışlardır. Double DQN'e kıyasla daha üstün performans gösteren Rainbow DQN kullanılarak, güvenli olmayan manevralarda ajanı cezalandıran bir güvenlik geri bildirimli ödüllendirme sistemi geliştirilmiş ve eğitim süreci hızlandırılmıştır. Simülasyon ortamında temkinli, normal, agresif olarak üç farklı sürücü profiliyle, 20 araca kadar değişen yoğunlukta senaryolar oluşturulmuş, otonom araç ve çevresindeki araçların hız ve konum bilgilerini kullanarak kararlar almıştır. Eğitim sürecinde, hız teşviki, çarpışma cezası ve güvenlik ihlallerine dayalı bir ödül fonksiyonu uygulanarak kazalar önemli ölçüde azaltılmıştır. Sonuçlar, Rainbow DQN'nin hem kural tabanlı MOBIL algoritmasına hem de Double DQN'ye kıyasla daha iyi performans gösterdiğini ve güvenlik katmanıyla desteklendiğinde eğitim sürecinin hızlandığını ortaya koymuştur. Çalışma, güvenli ve

verimli bir otomatik şerit değiştirme karar verme sistemi sunarak derin pekiştirmeli öğrenmenin pratik uygulamalarını göstermektedir (Yavas ve diğerleri, 2020).

Peng ve diğerleri, derin pekiştirmeli öğrenmeye dayalı çift katmanlı bir karar verme algoritması geliştirerek otonom hız kontrolü ve şerit değiştirme kararlarını ayrı yönetim sistemleri kullanarak entegre etmeyi amaçlamıştır. Üst katmanda D3QN algoritması şerit değiştirme manevrasını yönetirken, alt katmandaki DDPG algoritması hız kontrolünü yönetmektedir. Makalede ifade edildiği üzere geleneksel çalışmalarda şerit değiştirme, hız ve takip mesafesi kontrolü genellikle ayrı ayrı ele alınırken, bu çalışma bu iki hareketi entegre ederek daha senkron bir sürüş sistemi geliştirmiştir. Model SUMO simülasyon ortamında NGSIM verileri kullanılarak eğitilmiş ve test edilmiştir. Sonuçlar, sunulan sistemin araç hızını %23,99 artırarak diğer yöntemlere kıyasla daha verimli kararlar verdiğini göstermiştir. Çift katmanlı model, geleneksel sistemlere kıyasla daha az şerit değiştirmiş ve daha yüksek hız kazancı sağlamıştır. Bununla beraber güvenlik açısından daha iyi sonuçlar elde etmiştir (Peng ve diğerleri, 2022).

Mirchevska ve diğerleri, yaptıkları çalışmada otonom araçların güvenli ve mantıklı şerit değiştirme kararları alabilmesi için Derin Pekiştirmeli Öğrenme (DRL) tabanlı bir yöntem sunmuştur. Geleneksel kural tabanlı sistemlerin karmaşık olması ve genelleme yeteneklerinin kısıtlı olduğunu belirterek yüksek seviyeli karar verme sürecini oluşturmak için Derin Q-Ağları (Deep Q-Networks - DQN) kullanmışlardır. Sunulan algoritmada minimum bilgiyle en hızlı öğrenmeyi sağlamak için 13 sürekli özellik içeren bir durum kullanılmış ve düşük seviyeli rota takibi süreçlerinden bağımsız olarak yüksek seviyeli kararlar optimize edilmiştir. Aynı zamanda, güvenlik doğrulama mekanizması ile pekiştirmeli öğrenme ajanının yalnızca güvenli eylemleri seçmesini sağlamış, böylece gerçek trafikte bile çarpışma olmadan öğrenme gerçekleştirilebildiği sonucuna ulaşılmıştır. Simülasyon sonuçları, çalışmada sunulan pekiştirmeli öğrenme ajanının geleneksel kural tabanlı bir sisteme kıyasla daha yüksek ortalama hızlara ulaştığını ve daha etkili şerit değiştirme kararları aldığını göstermiştir. Güvenlik doğrulama katmanının kapatıldığı testlerde, pekiştirmeli öğrenme ajanının çoğu senaryoda kaza yaptığı görülerek güvenlik mekanizmasının kritik olduğu kanıtlanmıştır (Mirchevska ve diğerleri, 2018).

Nagarajan ve Yi yaptıkları çalışmada çok ajanlı Derin Q-Ağları (Multi-Agent Deep Q-Network) kullanarak otonom araçların şerit değiştirme kararlarını optimize etmeyi

amaçlamıştır. Geçmiş çalışmalarda öne sürülen yöntemlerin çoğunda tek ajanlı bir yaklaşım benimsendiği ve yalnızca otonom aracının şerit değiştirdiğinin varsayıldığı belirtilmiş, bu çalışmada tüm araçların şerit değiştirebildiği daha gerçekçi bir çoklu ajan ortamı modeli ortaya koyulmuştur. Modelde her aracın hedef şeridine belirli bir mesafe içinde güvenli ve uygun bir manevra ile ulaşması amaçlanmış ve tüm ajanların deneyimleri tek bir ortak politika öğrenmek için kullanılmıştır. Carla üzerinde yapılan testlerde, modelin başarılı bir şekilde sürüş davranışlarını öğrendiği gözlemlenmiştir. Deneyler, büyük bir tekrar buffer'ının kazaları azaltıp politika öğrenme sürecini iyileştirdiğini ve tek ajanlı yöntemlere kıyasla daha güvenli ve verimli sonuçlar ürettiğini göstermiştir (Nagarajan ve Yi, 2021).

Lee ve Won Choi tarafından yapılan çalışmada otonom araçların dinamik şerit değiştirme davranışlarını öğrenmesi için derin pekiştirmeli öğrenme (DRL) ve derin Q ağı (DQN) tabanlı bir politika ağı önerilmektedir. Sunulan sistemde, algılama modülü çevredeki araçların konum ve hız bilgilerini toplayarak bunları 2D bir görüntüye dönüştürmekte, ardından CNN (Convolutional Neural Network) tabanlı politika ağı bu görüntüden etkileşim özelliklerini çıkararak güvenli bir şerit değiştirme kararı almaktadır. Kararın uygulanması, aracın uygun manevraları gerçekleştirmesini sağlayan kontrol modülü ile tamamlanmaktadır. Q-öğrenme yöntemi kullanılarak, aracın çevresiyle etkileşimi sonucu aldığı ödüllerle en iyi karar mekanizması geliştirilmektedir. Model, Pygame tabanlı bir simülasyon ortamında eğitilmiş, trafik kuralları ve farklı sürücü davranışları dikkate alınarak test edilmiştir. Sonuçlar, önerilen sistemin güvenli ve tutarlı şerit değiştirme kararları alabildiğini, çarpışma yaşanmadan operasyonları tamamlayabildiğini ve iş birlikçi olmayan sürücülere karşı bile başarılı şekilde adapte olabildiğini göstermektedir (Lee ve Choi, 2019).

He ve diğerleri, otonom araçların şerit değiştirme kararlarını daha güvenli bir hale getirmek için gözlem karşıtı pekiştirmeli öğrenme (Observation Adversarial Reinforcement Learning - OARL) yaklaşımını öneren bir çalışma sunmuştur. Otonom araçların sensör hataları veya gözlem belirsizlikleri nedeniyle yanlış kararlar almasının önüne geçmek amacıyla gerçekleştirilen çalışmada politika kısıtlamaları ve gözlem bozulmalarını inceleyen kısıtlı gözlem-robust Markov karar süreci (COR-MDP) modeli geliştirilmiştir. Bayes optimizasyonuna dayalı bir Black-Box Attack tekniği kullanılarak, en olumsuz gözlem bozulmaları etkin şekilde tahmin edilmiştir. SUMO tabanlı simülasyonlarla farklı trafik yoğunluklarında test edilen yöntem, yalnızca otonom aracın performansını artırmakla

kalmayıp, aynı zamanda şerit değiştirme politikalarının gözlem bozulmalarına karşı dayanıklılığını da geliştirmiştir (He ve diğerleri, 2022).

Wang ve diğerleri, Geliştirdikleri modelde otonom araçların yol takip performansını geliştirmek amacıyla Derin Pekiştirmeli Öğrenme tabanlı bir kontrol yöntemi geliştirmiştir. Geleneksel rota takibi yöntemlerinin değişken çevresel koşullar ve karmaşık senaryolar karşısında yetersiz kaldığını vurgulayarak bu eksiklikleri gidermek için derin öğrenme ve pekiştirmeli öğrenmeyi birleştiren bir kontrol yapısı önermiştir. Bu kapsamda, Aktör-Kritik (Actor-Critic) mimarisine dayalı bir derin pekiştirmeli öğrenme algoritması uygulanmış ve araç kontrolü hem çevresel geri bildirimlere hem de kendi hareket dinamiklerine göre optimize edilmiştir. Modelin eğitiminde sürekli bir öğrenme yaklaşımı benimsenmiş bu sayede farklı senaryolarda esneklik ve uyum yeteneği kazandırılmıştır. Önerilen yöntem hem düz hem de virajlı yollarda test edilmiştir. Simülasyon sonuçları önerilen derin pekiştirmeli öğrenme tabanlı kontrol yapısının yumuşak direksiyon kontrolü sağladığını, hedef rotanın etkili ve uygun biçimde takibini gerçekleştirdiğini ve geleneksel (PID gibi) yöntemlere kıyasla daha kararlı bir rota takibi sunduğunu göstermiştir (Jiang ve diğerleri, 2019).

Sun ve diğerleri, gerçekleştirdikleri çalışmada insansız otonom araçlar için kinematik modele dayalı bir rota takibi denetleyicisi tasarımı geliştirmiş ve deneysel olarak değerlendirmiştir. Çalışmada aracın yanal yönlendirilmesini sağlayan kontrol yapılarının otonom sürüş güvenliği açısından kritik olduğu vurgulanmıştır. Aracın yanal konum hatasını minimize etmeye yönelik bir kontrol stratejisi tasarlanmış ve bu strateji araç kinematik modeline entegre edilmiştir. Denetleyici tasarımında temel alınan yöntem, öngörülü (deterministic) kontrol ve klasik PID yapılarını harmanlayan bir yapıya sahiptir. Yanal hata, yönelim hatası ve ilerleme hızı gibi faktörler dikkate alınarak kontrol girdileri belirlenmiştir. Bununla beraber sistemin doğruluğu ve sağlamlığı birden fazla test ortamında (hem simülasyon hem de gerçek araç testleriyle) değerlendirilmiştir. Deneysel sonuçlar geliştirilen kontrol algoritmasının yüksek hassasiyetli yanal kontrol sağlayarak rota takibi sağladığını ve gerçek zamanlı uygulamalar için uygun olduğunu ortaya koymuştur (Zhang ve diğerleri, 2024).

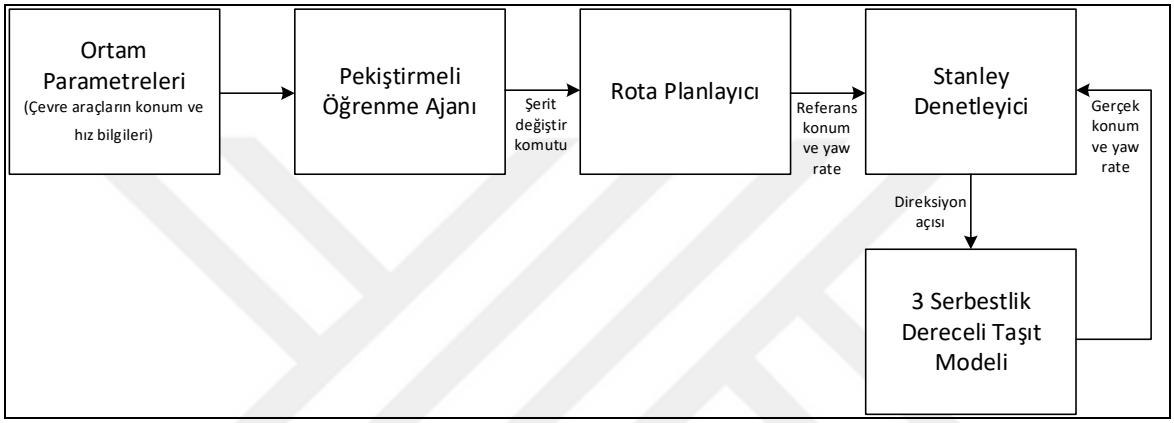
Tariq, Jayaraman ve Abdel-Aty, yaptıkları çalışmada şehir dışı yol koşullarında bağlı otonom elektrikli araçlar (Connected Autonomous Electric Vehicles) için sürdürülebilir bir rota takibi sistemi geliştirilmiştir. Geliştirilen sistem DDPG (Deep Deterministic Policy

Gradient) algoritması üzerine kuruludur. Çalışmanın amacı enerji verimliliğini artırırken aynı zamanda yol güvenliğini ve takip doğruluğunu da optimize etmektir. Araç, çevresinden gelen verileri (hız, sapma, yol eğimi vb.) kullanarak bir öğrenme politikası geliştirilmiştir. Elektrikli araçlar için kritik ve önemli noktalardan biri olan enerji tüketiminin minimize edilebilmesi ve bunun sürdürülebilirliğinin sağlanmasına da odaklanılarak öğrenme politikasının yönelimi belirlenmiştir. Yapılan simülasyonlarda önerilen sistemin hem yüksek takip doğruluğu sağladığı hem de enerji kullanımını azalttığı gösterilmiştir (Basile ve diğerleri, 2024).

Literatürde sunulan çalışmalar, otonom araçların şerit değiştirme kabiliyetlerini artırmak için çok çeşitli yöntemlerin uygulandığını ortaya koymaktadır. Kural tabanlı modellerden bulanık mantık denetleyicilerine, makine öğrenmesinden pekiştirmeli öğrenmeye kadar uzanan bu yöntemlerin her biri farklı avantajlar ve sınırlamalar barındırmaktadır. Örneğin, bulanık mantık sistemleri insan benzeri kararlar üretme konusunda başarılı olurken, trafik yoğunluğu gibi dinamik koşullara adaptasyonları sınırlı kalabilmektedir. Öte yandan, pekiştirmeli öğrenme tabanlı modeller özellikle karmaşık çevresel koşullarda yüksek esneklik ve karar çeşitliliği sunarak ön plana çıkmaktadır. Ancak bu sistemlerin başarısı büyük ölçüde kullanılan eğitim verisinin kalitesi, simülasyon ortamlarının gerçekçiliği ve ödül fonksiyonlarının etkinliğine bağlıdır. Mevcut literatür hem akademik hem de uygulamalı açıdan zengin bir çerçeve sunmakta olup, bu çalışmaların analiz edilmesi sayesinde tez kapsamında geliştirilen yeni yaklaşımın bilimsel zemini oluşturulmuştur.

3. GELİŞTİRİLEN OTONOM ŞERİT DEĞİŞTİRME ALGORİTMASI

Gerçek sürücülü bir taşıtta şerit değiştirme davranışı incelendiğinde temel olarak 3 adımda sürecin gerçekleştirildiği görülmüştür. Bu adımlar karar verme, rota planlama ve rota takibi olarak belirlenmiştir. Bu çalışmada belirlenen adımların her biri ele alınarak hareketin tamamı bütünsel olarak ele alınmıştır. Kurulan otonom şerit değiştirme yapısı Şekil 3.1'de verilmiştir.

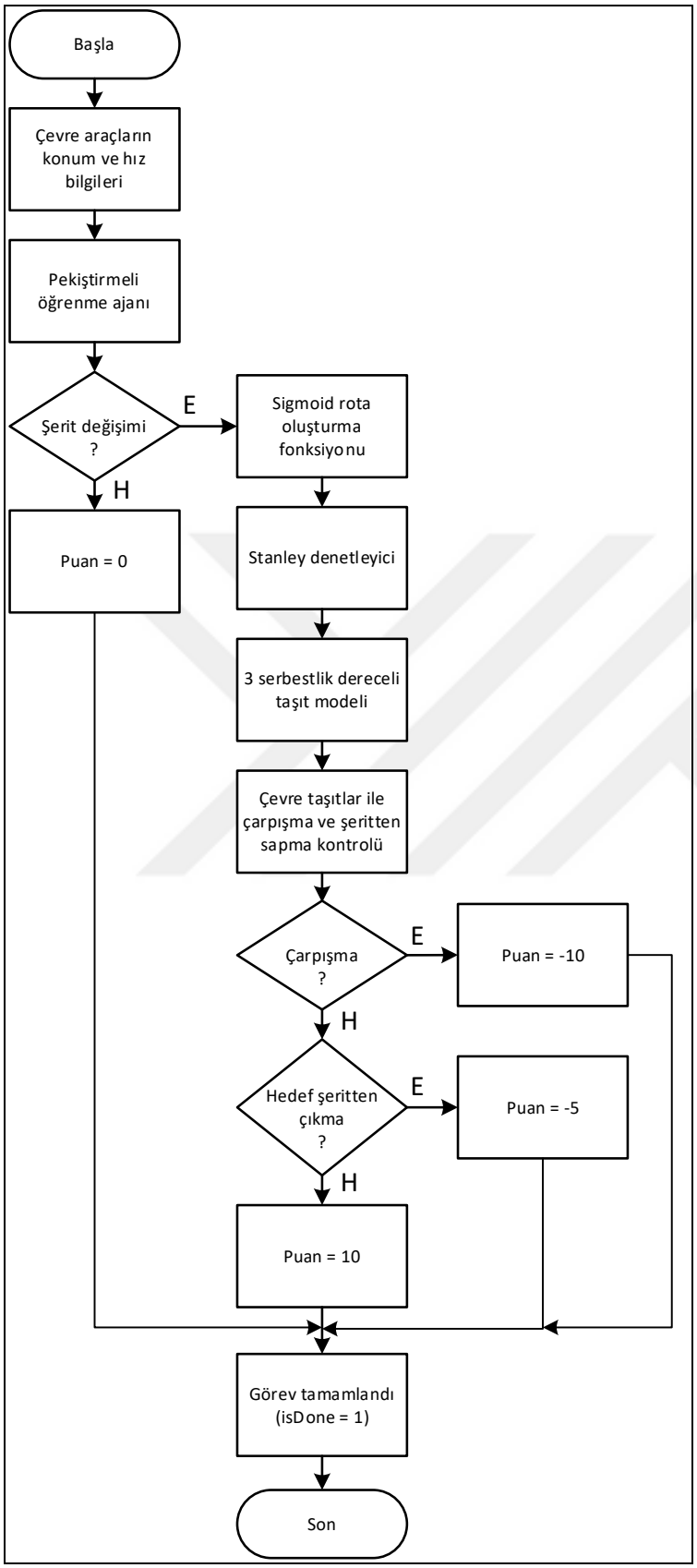


Şekil 3.1. Geliştirilen otonom şerit değiştirme algoritması

Öncelikle taşıta ve senaryoya ait konum ve hız gibi bilgiler tanımlanır. Konum bilgileri taşıtların merkezindeki x ve y koordinatları, hız ise x yönündeki hız olarak tanımlanmıştır. Bu bilgiler Matlab içerisinde fonksiyon olarak tanımlanmış ve her simülasyon başlangıcında rastgele atanacak şekilde planlanmıştır. Buradan elde edilen bilgiler ile otonom araç ve çevresindeki 4 araç için köşe noktalarının hesaplaması yapılır. Bu daha sonra kurulacak kaza kontrol yapısı için gereklidir. Bununla beraber çevre araçların merkez konum bilgileri ile otonom araca ait konum bilgilerinin farkı alınmış ve hız bilgisi ile beraber pekiştirmeli öğrenme ajanının gözlem girişine aktarılmıştır. Araçlara ait köşe noktaları koordinatları bilgileri ise çarpışma kontrolünü sağlayan fonksiyon bloğuna aktarılmıştır. Çarpışma kontrolü sağlayan bloktan çarpışma koşulu çıktısı alınmış ve bu çıktı pekiştirmeli öğrenme ajanının ödül (reward) girişine girdi olarak verilmiştir. Pekiştirmeli öğrenme ajanının simülasyonu durduran isdone girişi de ödül fonksiyonunun çıktılarından sağlanmıştır.

Pekiştirmeli öğrenme ajanı şerit değiştirme komutunu verdiği koşulda, rota planlama ve rota takibi sistemi devreye girecektir. Bu sistemler Simulink içerisinde bir Enabled Subsystem (Aktifleştirilmiş Altsistem) olarak tanımlanmıştır. Aksiyon çıkışı 1 geldiğinde rota

oluşturma sistemi devreye girer ve referans koordinatları oluşturur. Bununla beraber beklenen yaw oranı (desired yaw rate) ve eğrilik (curvature) gibi değerler hesaplanır. Bu bilgiler Stanley denetleyici sistemine aktarılarak direskiyon açısı belirlenir. Belirlenen direksiyon açısı otonom aracın modellendiği 3-DOF Single Track taşıt modeline aktarılır ve simülasyon tamamlanır. Taşıtın hareketine ait veriler (konum, direksiyon açısı vb.) X-Y grafikleri ile izlenebilir. Buna ek olarak otonom araç ve çevre araçlara ait hız ve konum bilgileri 3 boyutlu bir simülasyon görselleştirme modülüne aktarılır. Bu sayede trafik akışı ve otonom aracın şerit değişimi gibi hareketler 3 boyutlu olarak gözlemlenebilmektedir. Geliştirilen modele ait akış şeması aşağıdaki Şekil 3.2 ile verilmiştir.



Şekil 3.2. Pekiştirmeli öğrenme ajanı eğitim akışı

3.1. Taşıt Modeli

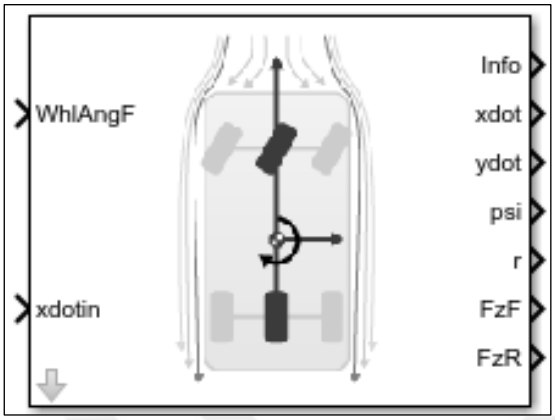
Otonom şerit değiştirme davranışı için karar verme aşaması pekiştirmeli öğrenme yaklaşımları ile kurgulanmıştır. Pekiştirmeli öğrenme yaklaşımında ajanın içerisinde bulunduğu çevre ile etkileşimli olarak deneme-yanılma anlayışı ile hareket etmesi nedeniyle algoritma tasarlanırken çevre tanımlamasının doğru ve yeterli yapılması büyük önem taşımaktadır. Şerit değiştirme kararının ve hareketinin yönetildiği otonom araç için Simulink içerisinde bulunan üç serbestlik dereceli tek izli taşıt modeli bloğu kullanılmıştır. Modelde kullanılan taşıt modeli üç serbestlik dereceli kinematik bisiklet modeli prensibi ile çalışmaktadır. 3 serbestlik dereceli (3-DOF) tek izli model (single track model), taşıtın yanal ve boylamsal dinamiklerini sadeleştirilmiş ancak etkili bir biçimde ifade edebilen bir yaklaşımdır. Özellikle kontrol sistemlerinin tasarımında, manevra simülasyonlarında ve algoritma testlerinde yaygın biçimde kullanılmaktadır. Çalışmada odaklanılan problem farklı trafik senaryolarında aracın doğru hareketi yapıp yapmadığı olduğundan blok ile birlikte verilen parametreler üzerinden uygulama ve simülasyonlar gerçekleştirilmiştir. Taşıta ait ağırlık, eylemsizlik momenti, teker dönüş sertlikleri gibi değerler literatürde rota takibi için geliştirilen bir çalışmada kullanılan değerler olarak belirlenmiştir (Byrne ve Abdallah, 1995). Taşıtlara ait uzunluk ve genişlik bilgileri hem otonom araç hem de çevre araçlar için aynı boyutta olacak şekilde tanımlanmıştır. Simülasyonlarda kullanılan taşıta ait parametrelerin değerleri 3.1. numaralı çizelgede verilmiştir.

Çizelge 3.1. Taşıt parametreleri

Ağırlık (m)	1 727 kg
Ön Aksın Ağırlık Merkezi Mesafesi (l_f)	1.17 m
Arka Aksın Ağırlık Merkezi Mesafesi (l_r)	1.42 m
Genişlik (W)	1.8 m
z Ekseni Eylemsizlik Momenti (I_z)	2 787 kg x m^2
Ön Tekerlerin Dönüş Sertliği (C_f)	47 000 N/rad
Arka Tekerlerin Dönüş Sertliği (C_f)	47 000 N/rad

Bu modelde taşıt, bir bisiklet benzetimiyle temsil edilir; ön ve arka tekerlek çiftleri sırasıyla tek bir tekerlek şeklinde sadeleştirilir. Bu sayede, taşıtın boylamasına hareketi (x yönü), yanal hareketi (y yönü) ve dönme hareketi (yaw, z yönünde dönme) olmak üzere üç temel hareketi dikkate alınır. Bu üç serbestlik derecesi, aracın yatay düzlemdeki hareketini

tanımlamak için yeterli kabul edilir. Taşıt modeline direksiyon açısı, başlangıç konumları ve taşıt hızı giriş olarak verilmiştir. Kullanılan taşıt modeli Şekil 3.3'te verilmiştir.



Şekil 3.3. Simulink 3 serbestlik dereceli taşıt modelli bloğu

Blok içerisinde tanımlanan katı cisim düzlemsel dinamikleri aşağıda verilen denklemler kullanılarak hesaplanmıştır. Denklemlerde kullanılan 'y' y eksenindeki konumu, 'x' x eksenindeki konumu, 'r' sapma oranını (yaw rate), 'a' ve 'b' taşıt ağırlık merkezi ile ön ve arka aks merkezleri arasındaki mesafeyi, $'M_{zext}'$ araç ağırlık merkezindeki dış momenti, 'I_{zz} 'araç gövdesinin sabit z ekseni etrafındaki atalet momentini ve F değerleri araç tekerlerine uygulanan yanal ve doğrusal kuvvetleri ifade etmektedir (Gillespie, 1992).

$$\ddot{y} = -\dot{x}.r + \frac{F_{yf}+F_{yr}+F_{yext}}{m} \tag{3.1}$$

$$\dot{r} = \frac{a.F_{yf}-b.F_{yr}+M_{zext}}{I_{zz}} \tag{3.2}$$

$$r = \dot{\psi} \tag{3.3}$$

$$\ddot{x} = 0 \tag{3.4}$$

Taşıta herhangi bir dış kuvvet uygulanmamaktadır. Blok girişi olarak doğrusal hız belirtildiğinden aşağıdaki denklemler dikkate alınmaktadır. Bu sebeple ön ve arka tekerleklere yanal ve doğrusal olarak uygulanan kuvvetler aşağıdaki gibidir. Verilen denklemlerde F_{xft} ve F_{xrt} ön ve arka tekerleklere etkiyen yanal kuvvetleri, F_{yft} ve F_{yrt} ön ve arka tekerleklere etkiyen doğrusal kuvveti C_{yf} ve C_{yr} ön ve arka tekerlekler için dönüş sertliğini (cornering stiffness), α ön ve arka tekerlekler için kayma açısını, μ ön ve arka tekerlekler için sürtünme katsayısını, F_{znom} ise ön ve arka akslara uygulanan nominal kuvveti temsil eder (Gillespie, 1992).

$$F_{xft} = 0 \tag{3.5}$$

$$F_{yft} = -C_{yf}.\alpha_f.\mu_f.\frac{F_{zf}}{F_{znom}} \tag{3.6}$$

$$F_{xrt} = 0 \tag{3.7}$$

$$F_{yrt} = -C_{yr}.\alpha_r.\mu_r.\frac{F_{zr}}{F_{znom}} \tag{3.8}$$

Blok içerisinde kullanılan taşıt modelinde ağırlık ve yük transferi sırasında etkin sürtünme parametrelerini değiştirmek için normal kuvvetleri nominal normal yüke böler. Pitch ve roll dengesinin korunması için 16 ve 17 numaralı eşitlikler kullanılır.

$$F_{zf} = \frac{b.m.g-(\ddot{x}-\dot{y}.r).m.h+h.F_{xext}+b.F_{zext}-M_{yext}}{a+b} \tag{3.9}$$

$$F_{zf} = \frac{a.m.g+(\ddot{x}-\dot{y}.r).m.h-h.F_{xext}+b.F_{zext}+M_{yext}}{a+b} \tag{3.10}$$

Kayma açılarının belirlenmesi için bölgesel, doğrusal ve yanal hızların oranı kullanılır. Lastik kuvvetlerinin belirlenmesi için kayma açıları kullanılır.

$$\alpha_f = a.\tan(\frac{\dot{y}+a.r}{\dot{x}}) - \delta_f \tag{3.11}$$

$$\alpha_r = a.\tan(\frac{\dot{y}+b.r}{\dot{x}}) - \delta_r \tag{3.12}$$

$$F_{xf} = F_{xft}.\cos\delta_f - F_{yft}.\sin\delta_f \tag{3.13}$$

$$F_{yf} = -F_{xft}.\sin\delta_f + F_{yft}.\cos\delta_f \tag{3.14}$$

$$F_{xr} = F_{xrt}.\cos\delta_r - F_{yrt}.\sin\delta_r \tag{3.15}$$

$$F_{yr} = -F_{xrt}.\sin\delta_r + F_{yrt}.\cos\delta_r \tag{3.16}$$

Taşıt bloğundan alınan konum bilgileri taşıtların ağırlık merkezi olarak belirlenmiş ancak çalışmada ulaşılmak istenen kapsama uygunluk bakımından gerekli hesaplamalar yapılarak köşe noktaları hesaba katılmıştır. Bu sayede dinamik bir trafik ortamının değişkenliğinin modellenmesi sağlanarak aracın farklı koşullarda doğru hareketi yapabilmeyi öğrenmesi amaçlanmıştır. Çevre araçlar için herhangi bir yanal hareket olmaması koşulu belirlenmiştir. Bu koşula bağlı olarak çevre araçlar için köşe noktaları koordinatları aşağıdaki gibidir (Ding ve diğerleri, 2019). Eşitliklerde kullanılan x ve y ağırlık merkezi konumlarını, L araç uzunluğunu ve W araç genişliğini ifade eder.

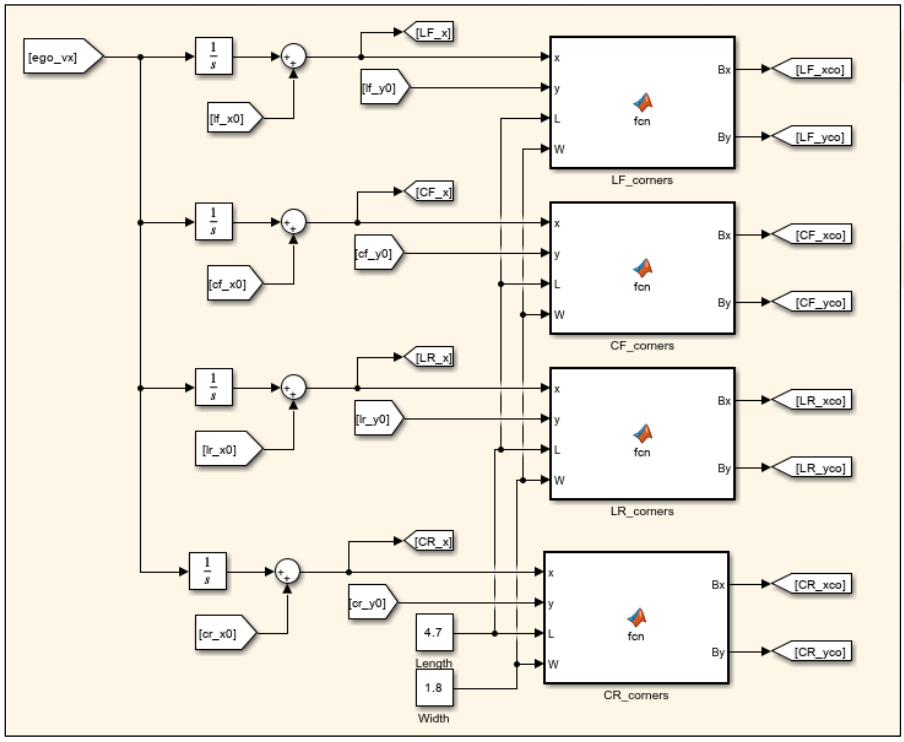
$$B_{1,i} = \left(x_i + \frac{L}{2} \right), \left(y_i + \frac{W}{2} \right) \quad (3.17)$$

$$B_{2,i} = \left(x_i - \frac{L}{2} \right), \left(y_i + \frac{W}{2} \right) \quad (3.18)$$

$$B_{2,i} = \left(x_i - \frac{L}{2} \right), \left(y_i - \frac{W}{2} \right) \quad (3.19)$$

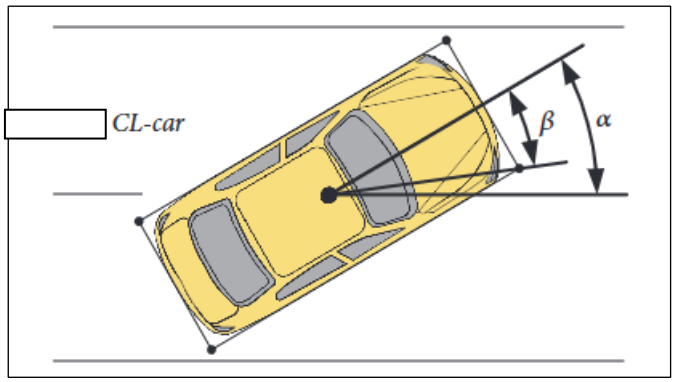
$$B_{4,i} = \left(x_i + \frac{L}{2} \right), \left(y_i - \frac{W}{2} \right) \quad (3.20)$$

Bu formüller Simulink içerisinde araçların merkez konumlarının ve boyutlarının giriş olarak alındığı farklı fonksiyon blokları ile hesaplanarak köşe noktaların koordinatları çıkış olarak verilmiştir. Bu Simulink yapısı Şekil 3.4'te görüldüğü gibidir.



Şekil 3.4. Çevre araç köşe noktaları hesaplama blok yapısı

Çevre araçlardan farklı olarak otonom araç şerit değiştirme hareketi yapabileceğinden taşıtın köşe noktalarının belirlenmesinde taşıtın yönelim açılarının da hesaba katılması gerekmektedir. Otonom araca ait yönelim açıları Şekil 3.5.'te verilmiştir.



Şekil 3.5. Otonom araç yönelim açıları (Ding ve diğerleri, 2019)

otonom aracın yönelim açıları ile hesaplanan köşe noktaları koordinatları aşağıdaki denklemler ile ifade edilmiştir (Ding ve diğerleri, 2019). Denklemlerde kullanılan x ve y ağırlık merkezi konumlarını, α yönelim açısını, β şekil parametresi açısını, L araç uzunluğunu, W araç genişliğini ifade eder.

$$A_{1,x} = x_e + \frac{\sqrt{L^2+W^2}}{2} * \cos(\alpha - \beta) \quad (3.21)$$

$$A_{1,y} = y_e + \frac{\sqrt{L^2+W^2}}{2} * \sin(\alpha - \beta) \quad (3.22)$$

$$A_{2,x} = x_e - \frac{\sqrt{L^2+W^2}}{2} * \cos(\alpha + \beta) \quad (3.23)$$

$$A_{2,y} = y_e - \frac{\sqrt{L^2+W^2}}{2} * \sin(\alpha + \beta) \quad (3.24)$$

$$A_{3,x} = x_e - \frac{\sqrt{L^2+W^2}}{2} * \cos(\alpha - \beta) \quad (3.25)$$

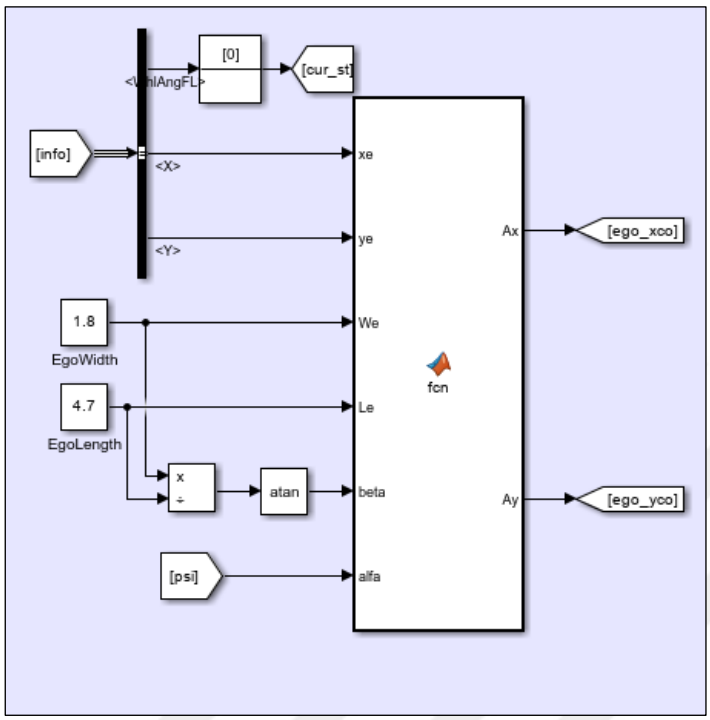
$$A_{3,y} = y_e - \frac{\sqrt{L^2+W^2}}{2} * \sin(\alpha - \beta) \quad (3.26)$$

$$A_{4,x} = x_e + \frac{\sqrt{L^2+W^2}}{2} * \cos(\alpha + \beta) \quad (3.27)$$

$$A_{4,y} = y_e + \frac{\sqrt{L^2+W^2}}{2} * \sin(\alpha + \beta) \quad (3.28)$$

Taşıt bloğundan alınan konum bilgileri taşıtların ağırlık merkezi olarak belirlenmiş ancak çalışmada ulaşılmak istenen kapsama uygunluk bakımından gerekli hesaplamalar yapılarak köşe noktaları hesaba katılmıştır. Bu sayede dinamik bir trafik ortamının değişkenliğinin modellenmesi sağlanarak aracın farklı koşullarda doğru hareketi yapabilmeyi öğrenmesi amaçlanmıştır.

Şekil 3.6'da görüldüğü üzere taşıta ait uzunluk ve genişlik bilgileri ile beraber mevcut x ve y konumları girdi olarak kullanılmış ve bir fonksiyon bloğu ile köşe noktaların koordinat hesaplamaları yapılmıştır.



Şekil 3.6. Otonom araç köşe noktaları hesaplaması

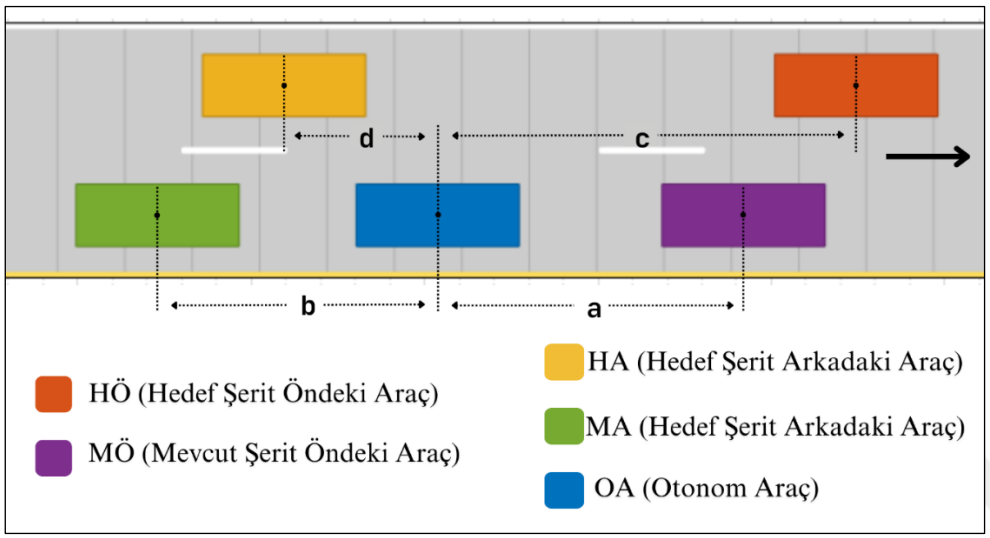
3.2. Senaryo Parametreleri

Taşıtın içerisinde bulunduğu senaryo tanımlanırken otonom araç ile çevre araçların boyut değerleri aynı olarak belirlenmiştir. Taşıtların iki şeritli ve herhangi bir viraj ya da kavşak unsurları olmayan bir yolda ilerledikleri kabul edilmiştir. Araçların ilerlediği yolda bir şerit genişliği literatürdeki benzer çalışmalar ve gerçek yol şartları göz önüne alınarak 3.65m olarak belirlenmiştir. Modelin daha yalın bir yapıda olabilmesi için çevre araçların şerit değişimi, hızlanma ya da yavaşlama gibi herhangi bir hareket yapmadığı, bulunduğu şeritte sabit hızda ilerlediği kabulü yapılmıştır. Bununla beraber gerçek trafik koşullarının uygulanan modele yansıtılabilmesi için her simülasyon başlangıcında otonom araç ve çevre araçlar için rastgele konumlar belirlenmiştir. Buna ek olarak her simülasyon başlangıcında otonom araç ve çevre araçlara ait hız değeri de her bir araç için aynı ve 10 m/s ile 30 m/s arasında değişen değerler olarak rastgele olacak şekilde belirlenmiştir. Bu sayede ajanın eğitimi sırasında her bir adımda farklı koşullarla karşılaşılması ve etkin bir öğrenme

sürecinin sağlanması amaçlanmıştır. Simülasyon senaryoları oluşturulurken kullanılan parametreler Çizelge 3.2.'de görülmektedir.

Çizelge 3.2. Senaryo parametreleri

Şerit Sayısı	2
Şerit Genişliği	3,65 m
Çevre Araç Sayısı	4
Maksimum ve Minimum Hız	10 m/s – 30m/s



Şekil 3.7. Tanımlanan senaryo ve parametreleri

Şekil 3.7.'de verilen şemada oluşturulan trafik senaryosunun görselleştirilmiş hali görülmektedir. Otonom aracın bulunduğu şeritte öndeki araç MÖ, arkadaki araç MA; hedef şeritte öndeki araç HÖ, arkadaki araç HA olarak tanımlanmıştır. Araçların her simülasyon başlangıcındaki rastgele konum belirlemeleri yapılırken araçların merkez noktalarının birbirine olan uzaklıkları temel alınarak hesaplama fonksiyonu oluşturulmuştur. Bu uzaklıklar a, b, c, d ile belirlenmiştir.

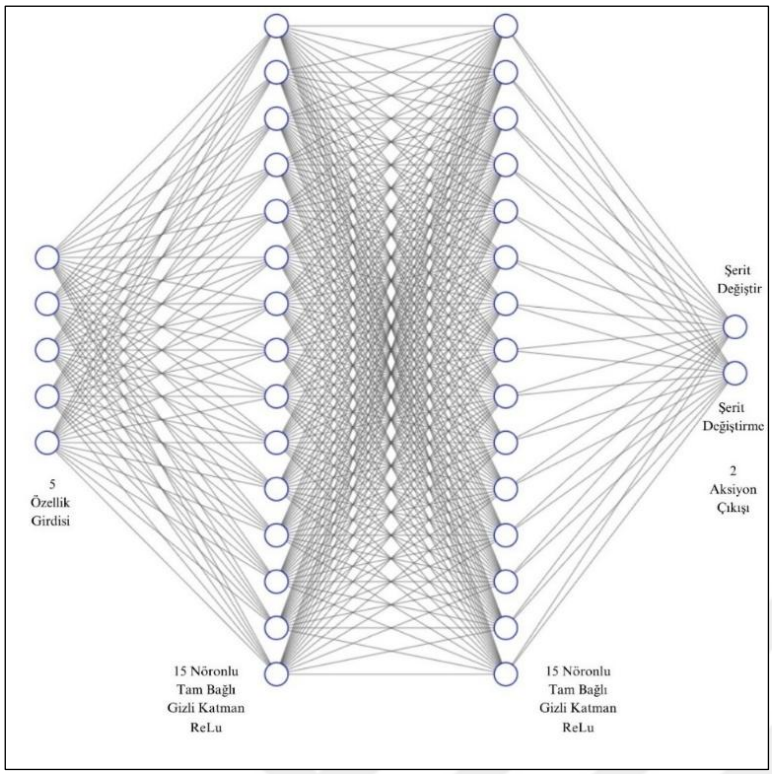
Belirlenen koşulların simülasyona aktarılabilmesi için çalışma alanı üzerinde bir fonksiyon oluşturularak taşıtların başlangıç konumlarının ve hızlarının rastgele olarak belirlenmesi sağlanmıştır. Simülasyonda kullanılmak üzere bu değerler Simulink içerisine aktarılmış ve diğer bloklara iletilebilecek şekilde düzenlenmiştir.

3.3. Karar Verme Aşaması

Otonom şerit değiştirme sürecinin ilk adımı karar verme aşamasıdır. Aracın mevcut durumuna ve çevresindeki şartlara göre "şeritte kal" ya da "şerit değiştir", "hızlan" ya da "yavaşla" gibi temel aksiyonları seçmesini içerir. Bu çalışmada karar verme aşaması seçilen aksiyonlardan elde edilen ödüller doğrultusunda ajanın eğitilmesi ile en uygun aksiyonun seçildiği bir pekiştirmeli öğrenme algoritması kullanılarak yapılandırılmıştır. Ajan çevresel gözlemleri (araçlar arası mesafeler, hız farkları, mevcut şerit bilgisi vb.) kullanarak her zaman adımında aksiyon seçimi yapar. Kullanılan pekiştirmeli öğrenme algoritması, ayrık aksiyon uzayına sahiptir ve "şeritte kal (0)" ve "şerit değiştir (1)" olmak üzere yalnızca iki aksiyon seçeneği bulunmaktadır. Bu sayede, eylem seçimi basit ancak etkili bir yapıda gerçekleştirilmiştir. Karar verme sürecinde özellikle çarpışmadan kaçınma ve konforlu sürüş gibi faktörler ödül fonksiyonu içerisinde tanımlanmıştır. Bu yaklaşıma literatürde yaygın şekilde kullanılan DQN karar verme algoritmalarında da rastlanmıştır (Mnih ve diğerleri, 2015).

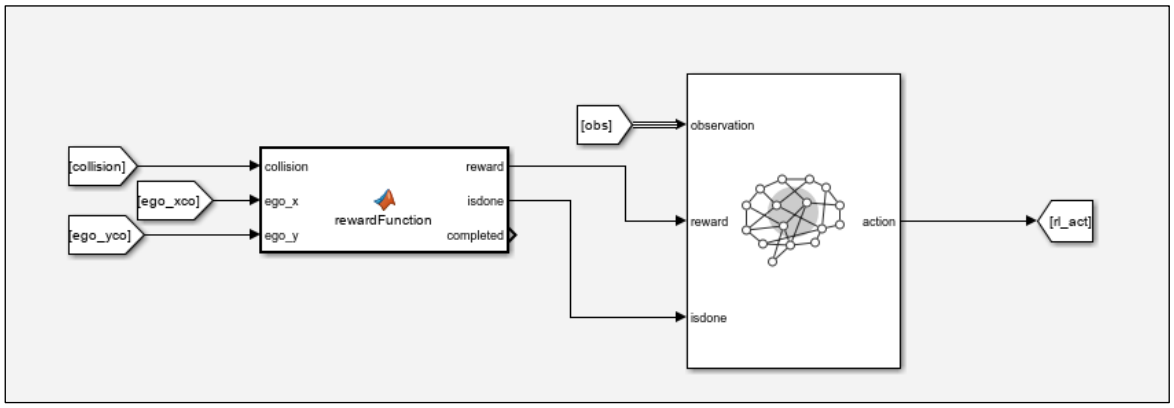
3.3.1. DQN ajanı

Taşıtın doğru şekilde karar verebilmesi için pekiştirmeli öğrenme ajanı DQN metodu ile oluşturulmuş ve eğitilmiştir. DQN metodunda değer bazlı bir pekiştirmeli öğrenme yöntemi olan Q-Learning yöntemi, bir değer tablosu oluşturmak yerine yapay bir sinir ağı kullanarak ajanın süreç içerisinde aldığı ödüller ile eğitilmesini sağlar. Daha basit ve yalın sistemler için daha düşük gizli katman sayıları yeterli olurken sistemin karmaşıklığı ya da kapsamı arttıkça ve sensör kullanımı gibi ek süreçler dahil edildikçe kullanılması gereken katman sayısı ve katmanlarda kullanılması gereken nöron sayısı da artacaktır. Mevcut çalışmada iki gizli katmanda 15 nöron ile oluşturulmuş ve doğrultucu olarak ReLU kullanılmıştır. Ajan için kullanılacak yapay sinir ağı Matlab içerisindeki Deep Network Designer eklentisi ile oluşturulmuştur. DQN ajanı için ağ yapısı Şekil 3.8.'de verilmiştir.



Şekil 3.8. Dqn ağ yapısı

Karar verme eylemini seçecek olan DQN ajanının süreci yürütebilmesi için 3 farklı parametrenin ajana tanıtılması gerekir. Bu parametreler gözlem kümesi (observation), ödül fonksiyonu (reward) ve simülasyonu bitirme komutu şartları (isdone) olarak belirlenmiştir. DQN ajanının Simulink konfigürasyonu Şekil 3.9'da verilmiştir.

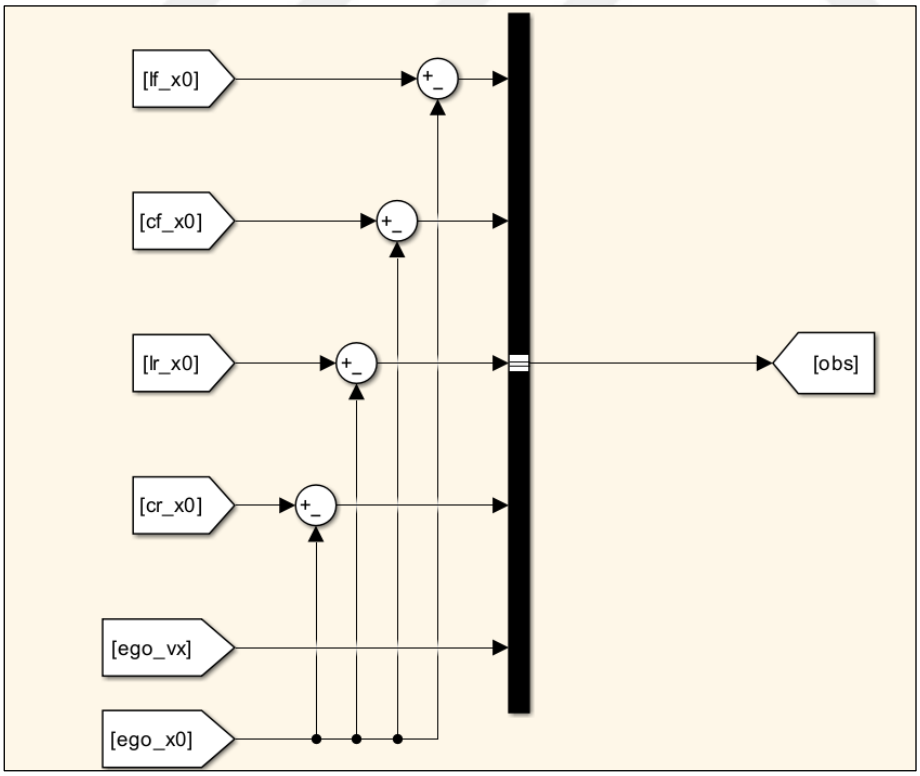


Şekil 3.9. Pekiştirmeli öğrenme ajanı simulink yapısı

3.3.2. Gözlem kümesi

Bu çalışmada kullanılan derin pekiştirmeli öğrenme tabanlı şerit değiştirme sisteminde, aracın çevresel farkındalığını sağlayan temel yapı gözlem kümesidir. DQN algoritması, her karar anında çevre hakkında aldığı bu gözlemleri kullanarak hangi aksiyonu seçeceğine karar verir. Bu nedenle, gözlem kümesinin doğru seçilmesi, öğrenmenin başarısı açısından kritik önem taşır.

Şerit değiştirme senaryosunda, aracın yalnızca kendi iç durumu değil, etrafındaki araçların konumu ve hareketleri de oldukça önemlidir. Çalışmada, aracın hızı, bulunduğu şerit, yönelimi ve ivmelenmesi gibi veriler gözlem kümesinin temelini oluşturur. Bunun yanında, aracın önünde ve arkasında hem aynı hem de hedef şeritte bulunan araçların konumları, hızları ve uzaklıkları da dikkate alınmıştır. Bu sayede, ajan yalnızca anlık pozisyon bilgisiyle değil, göreli hareketleri de göz önünde bulundurarak daha bilinçli kararlar alabilmektedir. Gözlem kümesinde kullanılan parametreler Çizelge 3.3.'te verilmiştir. Bu parametrelerin Simulink modelindeki isimlendirmeleri ve blok bağlantıları Şekil 3.10'da gösterilmektedir.



Şekil 3.10. Simulink modeli gözlem kümesi girişleri

Gözlem kümesinde kullanılan tüm bilgiler, sabit boyutlu bir vektör haline getirilmiştir. Bu yaklaşım sayesinde sistem, hem gerçek zamanlı çalışmaya uygun hale getirilmiş hem de öğrenme sürecinde tutarlılık sağlanmıştır. Görsel veya işlenmemiş sensör verisi yerine, işlenmiş ve özetlenmiş sayısal veriler kullanılmıştır. Bu tercih, hesaplama yükünü azaltırken aynı zamanda eğitim süresini kısaltmıştır. Sonuç olarak sistem, çevresindeki araçların pozisyonları ve hızlarına duyarlı şekilde tepki verebilen, çevreyi anlamlandırabilen bir yapıda geliştirilmiştir.

Çizelge 3.3. Pekiştirmeli öğrenme ajanı gözlem kümesi

a	Otonom araç ile MÖ arasındaki mesafe (m) (ego_x_0 - cf_x_0)
b	Otonom araç ile MA arasındaki mesafe (m) (ego_x_0 - cr_x_0)
c	Otonom araç ile HÖ arasındaki mesafe (m) (ego_x_0 - tf_x_0)
d	Otonom araç ile HA arasındaki mesafe (m) (ego_x_0 -tr_x_0)
v	Otonom ve çevre araçların x eksenindeki hızları (m/s) (ego_v_x)

3.3.3. Ödül fonksiyonu

Pekiştirmeli öğrenme algoritmalarında ajanın davranışlarını yönlendiren en önemli yapı ödül fonksiyonudur. Bu çalışmada, şerit değiştirme görevini üstlenen aracın doğru ve güvenli kararlar alabilmesi için amaca yönelik, sade ancak etkili bir ödül yapısı tasarlanmıştır. Ajan, her yaptığı eylem sonucunda ortamdan bir ödül alır ve bu ödüller zaman içinde bir stratejiye dönüşür.

Ödül fonksiyonu oluşturulurken sistemin güvenli, konforlu ve verimli hareket etmesi önceliklendirilmiştir. Özellikle çarpışmalar durumunda sistemin yüksek oranda cezalandırılması sağlanarak güvenlik kritik bir hedef haline getirilmiştir. Bunun yanında, şerit değiştirmenin başarıyla tamamlandığı durumlarda ise pozitif bir ödül verilmiştir. Ancak burada önemli olan, her şerit değişikliğini ödüllendirmek değil; yalnızca ihtiyaç duyulduğunda ve uygun koşullar altında yapılan değişikliklerin desteklenmesidir. Yolculuk boyunca istikrarlı bir hızda ilerlemek, ani fren veya sert direksiyon hareketlerinden kaçınmak gibi sürüş konforunu etkileyen unsurlar da ödül fonksiyonuna yansıtılmıştır. Böylece sistem, yalnızca hedef şeride geçmek için değil, bunu mümkün olduğunca yumuşak ve dengeli bir şekilde yapmak için de teşvik edilmiştir. Ödül yapısında sade formüller tercih edilmiş, karmaşık fiziksel modeller yerine davranışsal hedeflere odaklanılmıştır. Bu durumda sistem, çevresel koşulları değerlendirip uygun zamanda şerit değiştirmeyi

öğrenirken aynı zamanda çarpışmalardan kaçınmayı, sürüş konforunu korumayı ve gereksiz hareketlerden uzak durmayı da hedeflemiştir. Bu yapının, aracın kendi başına güvenli ve mantıklı kararlar verebilmesini sağladığı gözlemlenmiştir. Çarpışma koşulu kontrol fonksiyonu aşağıda verilmiştir.

Çarpışma koşulu kontrol fonksiyonu

collisionCheck(MÖ, MA, HÖ, HA, OA)
Çarpışma = 0
Çakışma_sayısı = 0
Eğer OA, HÖ ile çakışıyor ise çakışma_sayısı = 1
Eğer OA, MÖ ile çakışıyor ise çakışma_sayısı = 1
Eğer OA, MA ile çakışıyor ise çakışma_sayısı = 1
Eğer OA, HA ile çakışıyor ise çakışma_sayısı = 1
Eğer çakışma_sayısı ≥ 1 ise çarpışma = 1

Ödül fonksiyonu puan değerleri 37 numaralı eşitlikte verilmiştir.

$$R = \begin{cases} -10, \text{çarpışma} = 1\ (Herhangi\ bir\ çevre\ araç\ ile\ çarpışma\ varsa) \\ 10,\ \text{şerit değişimi} = 1 \quad çarpışma = 0 \\ 0,\ \text{şerit değişimi} = 0 \quad çarpışma = 0 \\ -5,\ yol\ sınırları\ dışına\ çıkma = 1 \end{cases} \tag{3.29}$$

3.3.4. Aksiyon kümesi

Bu çalışmada karar verme aşaması için kullanılan DQN algoritması, ayrık bir aksiyon kümesi ile çalışmaktadır. Dolayısıyla aracın alabileceği kararlar önceden belirlenmiş sınırlı sayıda seçenekten oluşur. Bu yapı, öğrenme sürecini daha kararlı ve takip edilebilir hale getirmiştir. Şerit değiştirme gibi kararların zamanlaması ve uygulanma koşulları, bu aksiyon kümesinin sınırları içinde ele alınmıştır. Mevcut çalışmada DQN ajanı için belirlenen aksiyon kümesi aşağıdaki gibidir:

action= (0, 1)

Uygulamada ajan her karar anında yalnızca iki temel seçeneğe sahiptir: bulunduğu şeritte kalmak, ya da sola geçmek. Bu sade yapı, sistemin daha hızlı öğrenmesini sağlamıştır. Ek

olarak, bu iki kararın uygulanabilirliği çevre koşullarına bağlıdır. Yani sistem, hedef şerit doluysa ya da hedef şeritteki araçların konumları uygun değilse hedef şeride geçme kararını alamaz. Bu tür durumlar, ortam tarafından kısıtlanarak sistemin sadece geçerli aksiyonlar üzerinden karar alması sağlanmıştır.

Bu sadeleştirilmiş aksiyon yapısı sayesinde sistem, yalnızca temel yönlendirmelerle karmaşık senaryolarda etkili kararlar almayı öğrenmiştir. Aksiyonların etkileri zamana yayılmıştır. Örneğin bir şerit değiştirme kararı sadece o anı değil, birkaç saniyelik bir geçiş sürecini de kapsar. Dolayısıyla sistem, bu tip kararların zamanlamasını da öğrenmiş ve aceleci ya da geç kalan hamlelerden kaçınmıştır.

Tanımlanan dinamik senaryo yapısı ve sadeleştirilmiş aksiyon yapısı sayesinde sistem, aksiyon sayısı sınırlı olsa da çevresel bilgiyi doğru yorumlayarak karmaşık sürüş senaryolarında anlamlı kararlar verebilecek esnekliğe sahip hale gelmiştir. Hem eğitim hem de test aşamalarında, bu sınırlı ama etkili aksiyon kümesinin öğrenme başarısına olumlu katkı sağladığı gözlemlenmiştir.

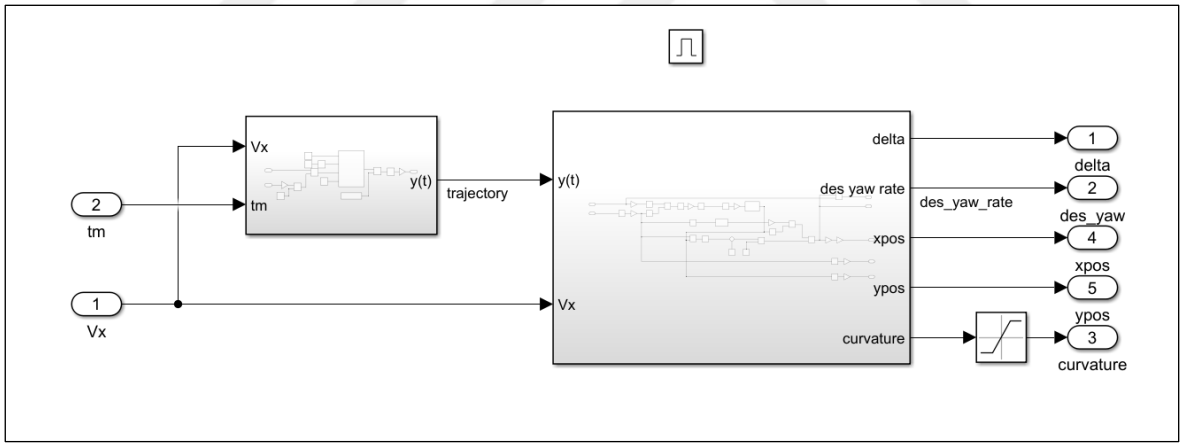
3.4. Rota Planlama Algoritması

Otonom sürüş sistemlerinde rota planlama, bir aracın hedef konuma ulaşmak için izlemesi gereken güzergâhı belirleme sürecidir. Bu çalışma kapsamında geliştirilen sistemde, rota planlama daha önceden tanımlı, düz ve iki şeritli bir yol üzerinde gerçekleştirilmiştir. Uygulama senaryosunda yol geometrisi sabittir ve viraj, kavşak ya da dönüş noktası gibi karmaşık yapılar yer almamaktadır. Bu durum, rota planlamanın sade ancak etkili bir şekilde uygulanmasını mümkün kılmıştır.

Geliştirilen rota planlama yapısında, aracın izleyeceği güzergâh, hedeflenen şeride geçişe yönelik olarak oluşturulmuştur. Aracın başlangıç konumu ile hedef şerit üzerindeki hedef pozisyonu arasında yumuşak bir geçiş sağlamak amacıyla sigmoid fonksiyonu tercih edilmiştir. Sigmoid fonksiyonun karakteristik eğrisi, ani yönelme değişikliklerini önleyerek şerit değişimlerinin daha dengeli ve kontrollü olmasını sağlamıştır. Bu yaklaşım, özellikle iki şeritli düz yollarda, aracın mevcut konumundan hedef şeride geçişini düzgün bir eğri üzerinden tanımlamak için oldukça uygundur.

Rota planlama işlemi, çevresel koşulların sabit olduğu bir ortamda gerçekleştirilmiştir. Simülasyon ortamında dört çevre araç bulunmaktadır ve bu araçların hızları sabit olup, otonom araçla aynıdır. Bu araçlar şerit değiştirme davranışı sergilememekte, bulundukları şeridi koruyarak hareketlerini devam ettirmektedir. Bu nedenle rota planlama sırasında dinamik çarpışma tahmini (çevre araçların da yanal hareketlerinin ve yönelimlerinin hesaba katıldığı) gibi karmaşık hesaplamalara ihtiyaç duyulmamıştır. Ancak, şerit değişimi sırasında bu araçlarla çarpışma riski olup olmadığı kontrol edilerek geçiş için uygun zaman aralığı belirlenmiştir.

Her planlama adımında, sistem aracın bulunduğu şeritten hedeflenen şeride geçişini sağlayacak sigmoid tabanlı bir eğri üretmiştir. Bu eğri yalnızca yatay düzlemde bir sapmayı değil, aynı zamanda aracın sürüş dinamikleriyle uyumlu şekilde gerçekleşen yumuşak bir yön değişimini temsil etmektedir. Bu yön değişiminin hesaplanabilmesi için eğrilik (curvature) değerinin hesaplanması gerekmektedir. Rota planlamasının gerçekleştirildiği Simulink modelinin blok yapısı Şekil 3.11.'de gösterilmiştir.



Şekil 3.11. Rota planlama simulink modeli

3.4.1. Eğrilik (Curvature)

Rota takibi, otonom araçların çevresel unsurlarla uyumlu biçimde belirlenen rotayı izleyerek güvenli ve konforlu bir sürüş gerçekleştirmesini sağlayan temel fonksiyonlardan biridir. Bu sürecin başarılı bir şekilde yürütülmesi yol üzerindeki referans noktalarının izlenmesiyle beraber bu noktalar arasındaki geometrik ilişkilerin doğru şekilde değerlendirilmesiyle

mümkündür. Buna bağlı olarak eğrilik (curvature) kavramı yol geometrisinin temel bir özelliği olarak öne çıkmakta ve rota takibinin hassasiyetini doğrudan etkilemektedir.

Eğrilik, bir yol ya da eğri üzerindeki herhangi bir noktanın ne kadar kıvrıldığını ifade eden bir büyüklüktür. Matematiksel olarak eğrilik, bir nokta etrafındaki yön değişiminin o noktaya olan mesafeye oranı şeklinde tanımlanabilir. Otonom taşıtlarda bu büyüklük, yön kontrol sistemlerinin en kritik girdilerinden biridir. Özellikle direksiyon açısı belirleme süreçlerinde, aracın izlediği yolun eğriliği doğrudan yönlendirme komutlarının temelini oluşturur. Bu bağlamda birçok rota takip algoritması (örneğin Stanley, Pure Pursuit ya da Model Predictive Control (MPC)) kararlarını doğrudan eğrilik bilgisine dayandırarak üretmektedir (Ziegler ve diğerleri, 2014).

Eğrilik, yalnızca anlık direksiyon komutlarını belirlemekle kalmaz, aynı zamanda taşıtın gelecekteki hareketlerini planlamasında da önemli bir rol üstlenir. Aracın karşılaşacağı yol şartlarının ne ölçüde kıvrımlı olduğunu önceden bilmesi, hız profili oluşturma, direksiyon açısının doğru belirlenmesi ve sürüş güvenliğini sağlama açısından kritik öneme sahiptir. Özellikle yüksek hızda yapılan manevralarda, aracın yol tutuş sınırları yolun eğriliğiyle doğrudan ilişkilidir. Keskin virajlarda daha düşük hızlarla ilerlenmesi gerektiği, yol eğriliği dikkate alınarak belirlenebilir.

Ayrıca eğrilik, rota üzerindeki yanal hata (lateral error) ve yönelme açısı hatası (heading error) gibi kontrol girdilerinin daha anlamlı şekilde hesaplanabilmesine de katkı sunar. Referans yolun eğriliği ile aracın anlık konumu arasındaki farklar, denetleyicinin vereceği tepkilerin büyüklüğünü belirler. Bu sayede, taşıtın yönelmesi gereken doğrultunun ne kadar değişmesi gerektiği daha doğru şekilde belirlenebilir. Bunun yanı sıra, sadece aracın hedef noktaya ne kadar uzak olduğu değil, bu hedefe hangi eğrilik altında yaklaşacağı da kontrol stratejisinin önemli bir bileşeni haline gelir.

Oluşturulan rotaya bağlı olarak direksiyon açısı belirleneceğinden, bu belirlemeyi yapacak rota takibi algoritmasının kullanacağı girdilerden biri de rotanın eğrilik (curvature) özelliğidir. Bu özellik matematiksel olarak $k = \frac{d\theta}{ds}$ ile ifade edilir. Oluşturulan rotanın devamında rotanın eğriliğinin hesaplanması için aşağıdaki eşitlikler kullanılmıştır (Kreyszig, 1991).

$$ds = \sqrt{dx^2 + dy^2} \tag{3.30}$$

$$\tan\theta = \frac{dy}{dx} \tag{3.31}$$

$$\frac{d}{dx}(tan\theta) = \frac{d}{dx}\left(\frac{dy}{dx}\right) \tag{3.32}$$

$$\frac{d}{dx}(tan\theta)\frac{d\theta}{dx} = \frac{d^2y}{dx^2} \tag{3.33}$$

$$(1 + tan\theta)\frac{d\theta}{dx} = \frac{d^2y}{dx^2} \tag{3.34}$$

$$\left(1 + \left(\frac{dy}{dx}\right)^2\right)\frac{d\theta}{dx} = \frac{d^2y}{dx^2} \tag{3.35}$$

$$\frac{d\theta}{dx} = \frac{\frac{d^2y}{dx^2}}{1+\left(\frac{dy}{dx}\right)^2} \tag{3.36}$$

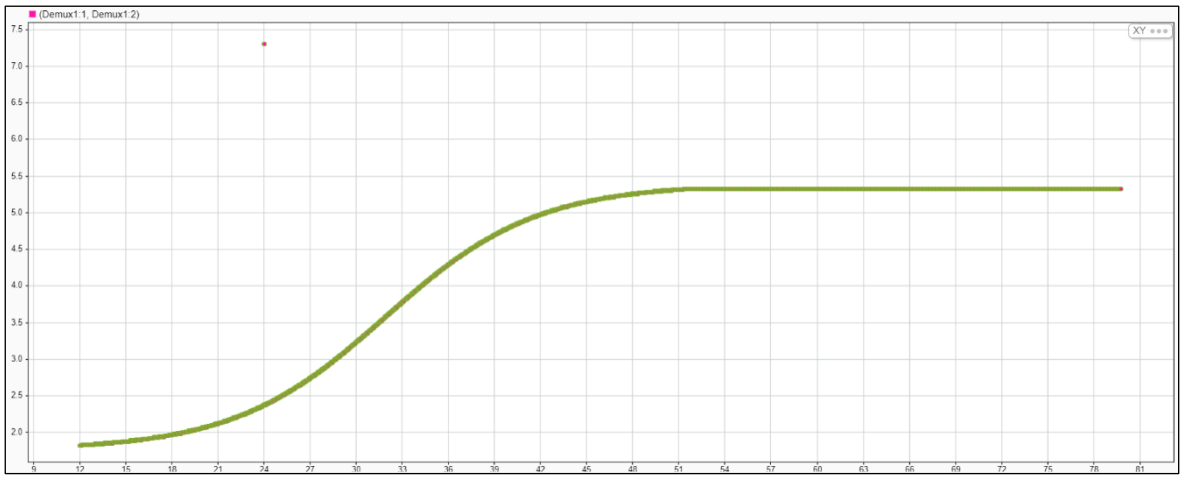
$$\frac{d\theta}{ds}\frac{ds}{dx} = \frac{\frac{d^2y}{dx^2}}{1+\left(\frac{dy}{dx}\right)^2} \tag{3.37}$$

$$\frac{ds}{dx} = \sqrt{\frac{dx^2+dy^2}{dx^2}} = 1 + \left(\frac{dy}{dx}\right)^2 \tag{3.38}$$

$$\frac{d\theta}{ds} = \frac{\frac{d^2y}{dx^2}}{\left(1+\left(\frac{dy}{dx}\right)^2\right).\left(1+\left(\frac{dy}{dx}\right)^2\right)^{1/2}} \tag{3.39}$$

$$\frac{d\theta}{ds} = \frac{\frac{d^2y}{dx^2}}{\left(1+\left(\frac{dy}{dx}\right)^2\right)^{\frac{3}{2}}} = k \tag{3.40}$$

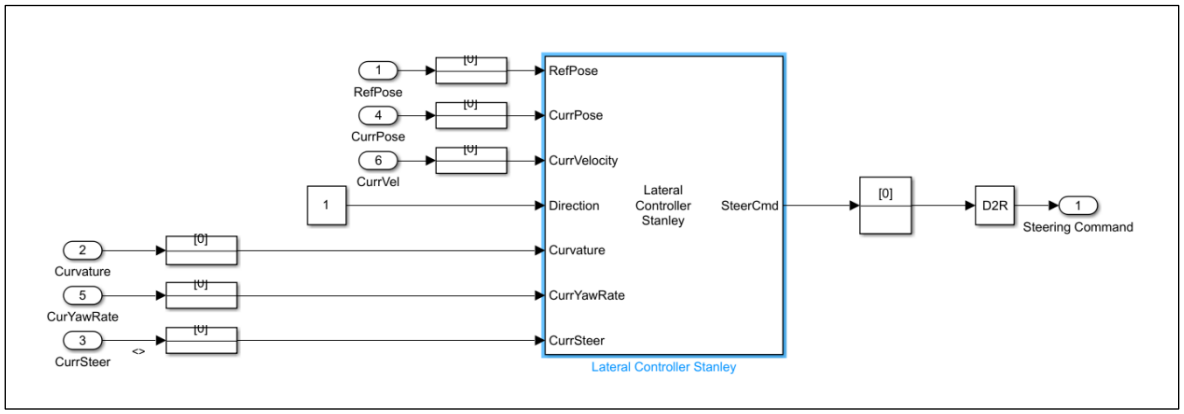
Bu yaklaşımla, karmaşık yol geometrilerine ihtiyaç duymadan, sabit yapılı ve iki şeritli düz bir yol üzerinde etkili bir rota planlama gerçekleştirilmiştir. Sistem, bu sade ortamda bile gerçekçi bir şerit değiştirme davranışını destekleyecek biçimde yapılandırılmıştır. Böylece, otonom sürüş sistemlerinin temel taşlarından biri olan rota planlama, sade ve güvenli bir ortamda başarıyla uygulanmış ve rota takip algoritmasına uygun bir altyapı sunmuştur. Modelde kullanılan rota oluşturma sistemi sonucunda otonom taşıtın takip etmesi için oluşturulan rota Şekil 3.12'de görüldüğü gibidir.



Şekil 3.12. Sigmoid fonksiyonu ile oluşturulan referans rota

3.5. Rota Takibi Algoritması

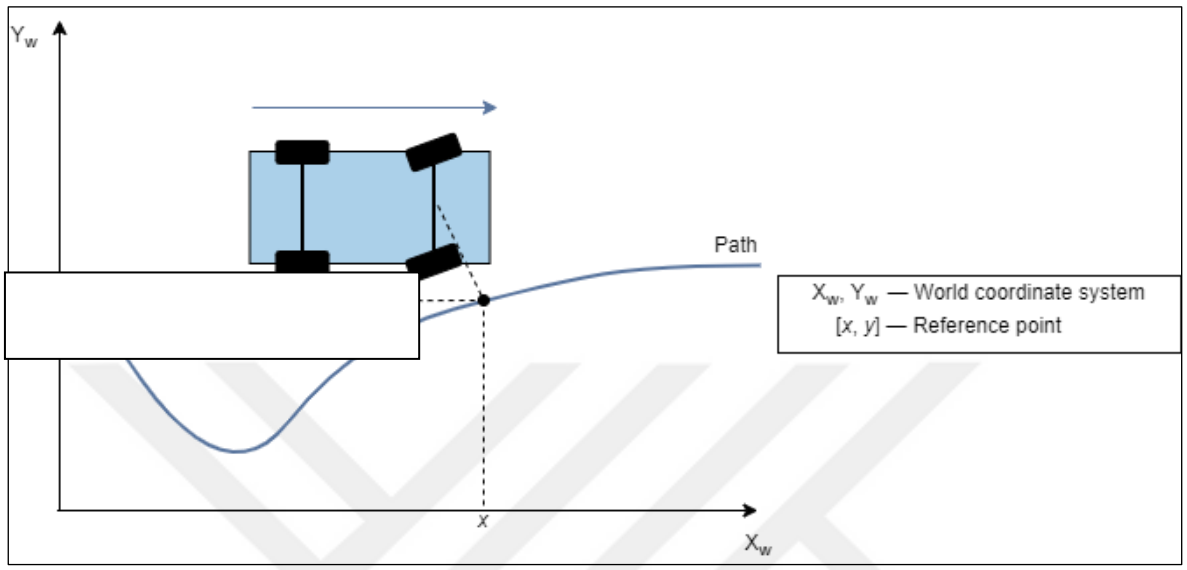
Rota takibi aşaması için kullanılan sistem Stanley denetleyici temel alınarak geliştirilmiştir. Yanal (Lateral) Stanley denetleyicisi için simulink içerisinde bulunan Lateral Controller Stanley bloğu kullanılmıştır. Blok içerisinde kinematik ve dinamik bisiklet modeli olarak seçilebilecek taşıt modellerinden dinamik bisiklet modeli tercih edilmiştir. Bu tercihin sebebi parametreler üzerinde daha hassas ayarlamalar yaparak taşıtın referans rotayı olabilecek en yakın şekilde takip edebilmesini sağlamaktır. Bu yapıda referans konum, mevcut konum, mevcut hız, hareket yönü, eğrilik, mevcut yaw rate ve mevcut direksiyon açısı bilgileri girdi olarak istenmektedir. Yanal Stanley denetleyicisine ait blok Şekil 3.13.'te görülüğü gibidir.



Şekil 3.13. Simulink stanley denetleyici bloğu

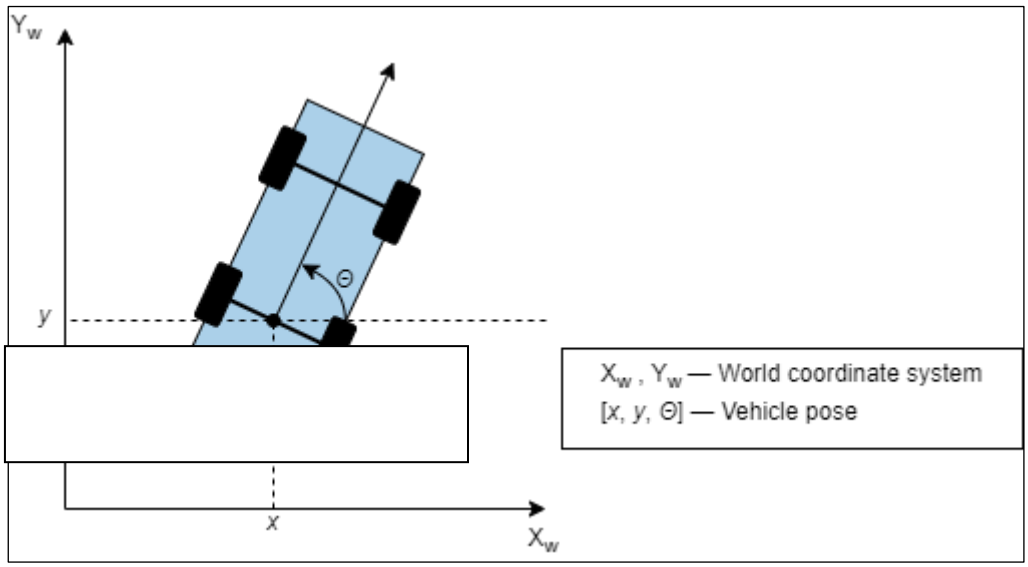
Referans konum [x, y, θ] vektörü olarak tanımlanmıştır. x ve y konumları metre cinsindendir ve θ açısı derece cinsindendir. x ve y noktaları aracın yönlendirileceği referans noktaları

belirtir. θ ise bu referans noktasındaki yolun yönelim açısını belirtir ve saat yönünün tersine pozitiftir (Thrun ve diğerleri, 2006). Stanley denetleyicinin belirleyici parametrelerinin şema üzerindeki gösterimi Şekil 3.14 ve 3.15.'te verilmiştir.



Şekil 3.14. Stanley denetleyici bloğu referans konum

Mevcut konumda referans konumda olduğu gibi [x, y, θ] vektörü olarak tanımlanmıştır. x ve y konumları metre cinsindendir ve θ açısı derece cinsindendir. x ve y noktaları aracın yönlendirileceği referans noktaları belirtir. θ ise bu referans noktasındaki yolun yönelim açısını belirtir ve saat yönünün tersine pozitiftir (Thrun ve diğerleri, 2006).

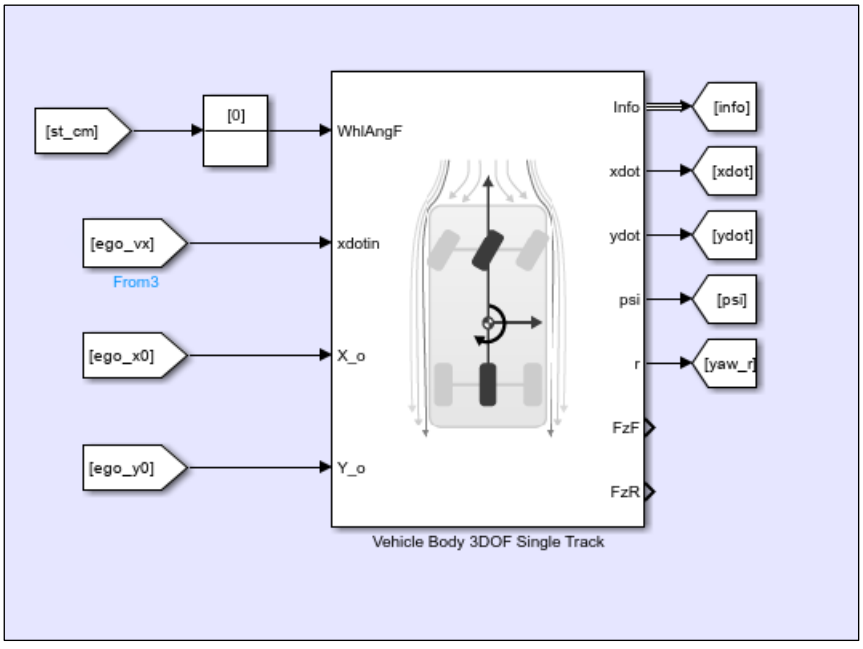


Şekil 3.15. Stanley denetleyici bloğu yönelim açısı

Aracın mevcut hızı x yönündeki skaler değeri olarak giriş yapılmaktadır. Hız birimi saniye başına metre (m/s) cinsindendir. Araç ileri hareket ediyorsa hız değeri 0'dan büyük olmalıdır. Araç geri yönde hareket halindeyse bu değer 0'dan küçük olmalıdır. Bu değer 0 olduğunda hareket halinde olmayan bir aracı temsil etmektedir. Aracın mevcut yönü ileri hareket için 1 ve geri hareket için -1 olarak tanımlanmaktadır. Sürüş yönü, direksiyon açısı değerini hesaplamak için kullanılan konum hatasını ve açı hatasını belirler (Thrun ve diğerleri, 2006).

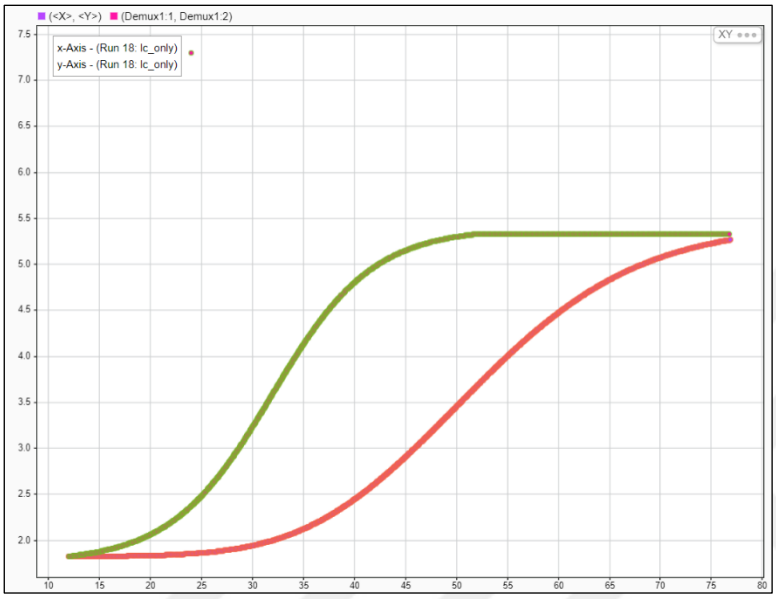
Aracın ileri hareket halindeyken pozitif olarak belirtilen konum kazancı blok içerisinden ayarlanabilmektedir. Bu değer konum hatasının direksiyon açısını ne kadar etkilediğini belirleyen parametredir. Bununla beraber yine blok içerisinde maksimum direksiyon açısı değeri de belirlenmelidir. 0 – 180 derece aralığında bir atama yapılmalıdır. Mevcut çalışmada literatürdeki benzer örneklerden yola çıkılarak 35 derecelik açı değeri sınır olarak belirlenmiştir.

Stanley denetleyicinin çıkışı olan direksiyon açısı komutu kinematik taşıt modelinin direksiyon açısı girişine aktarılarak otonom aracın şerit değiştirme hareketini yapması sağlanmıştır. Stanley denetleyici çıkışından alınan direksiyon değeri derece olduğundan taşıt bloğuna aktarılmadan önce bir dönüştürücü blok ile radyan cinsine çevrilerek taşıt bloğunun okuyabileceği formatta düzenlenmiştir. 3 serbestlik dereceli taşıt modelinin Simulink bloğuna ait giriş ve çıkış parametreleri Şekil 3.16'da verilmiştir.



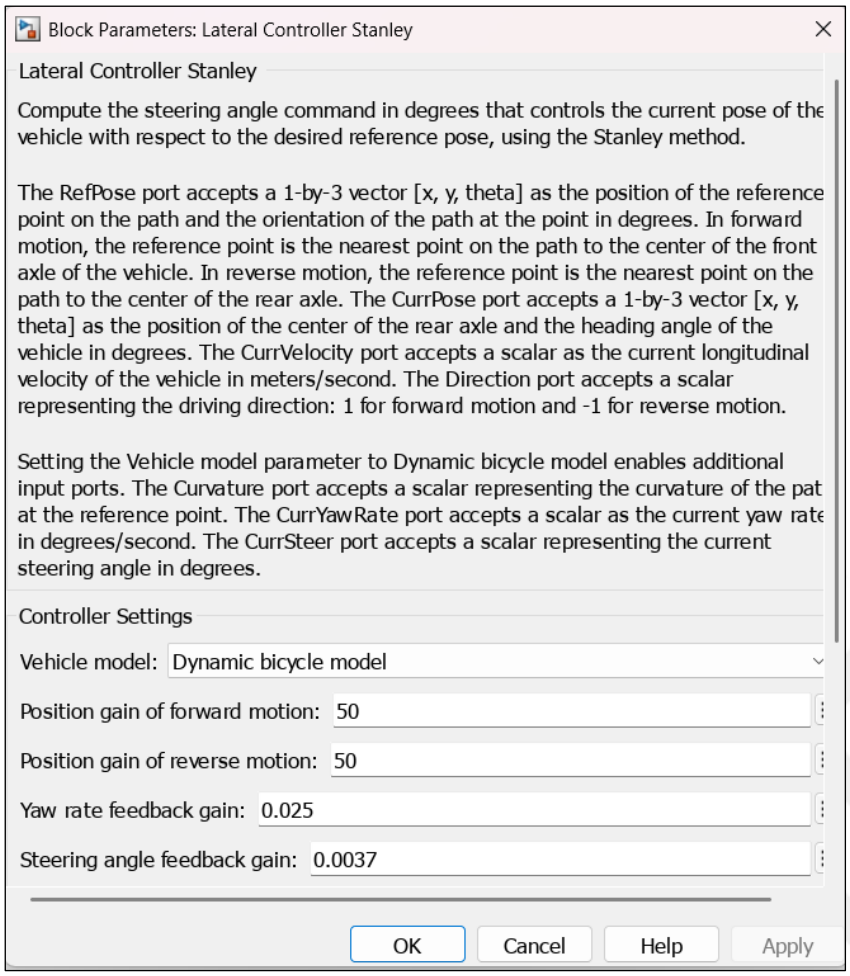
Şekil 3.16. Mevcut çalışmada kullanılan taşıt modeli bloğu

Stanley denetleyiciden elde edilen direksiyon açısı değerleri taşıt modeli bloğuna aktarıldıktan sonra taşıtın izlediği gerçek rotanın referans rota ile uyumu kontrol edilmiştir. Blok içerisindeki varsayılan değerler ile kontrol edildiğinde Şekil 3.17'de olduğu gibi yalnızca başlangıç ve bitiş noktalarının eş olduğu, eğrilik ve direksiyon açısı değerlerinin referansa uygun olmadığı görülmüştür.



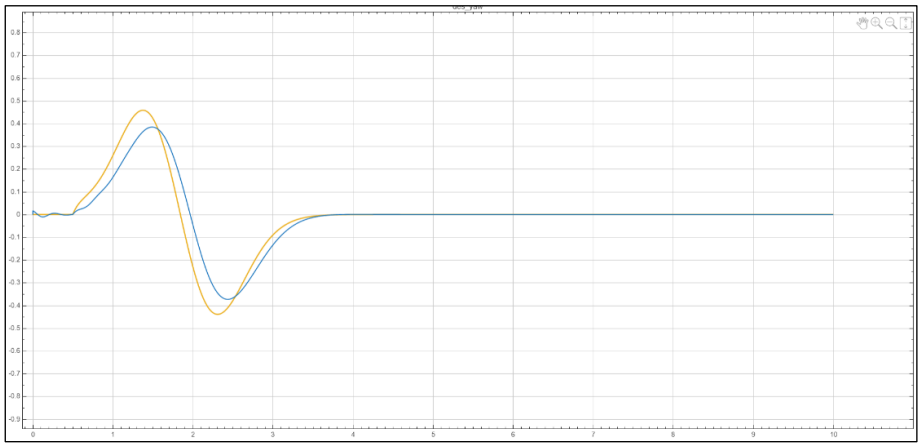
Şekil 3.17. Optimizasyon öncesi izlenen rota ile referans rota grafikleri

Bunun ardından dinamik bisiklet modeli olarak seçilen Stanley denetleyici parametreleri incelenmiş; denetleyici kazancı, yaw rate kazancı ve direksiyon açısı kazancı olmak üzere 3 farklı parametrenin rotaların uygunluğunu belirlediği görülmüştür. Mevcut sisteme uygun olan yöntem olarak manuel optimizasyon işlemi gerçekleştirilmiştir. Optimize edilmiş değerler Şekil 3.18 ile verilen blok parametreleri ekranında gösterilmektedir. Varsayılan olarak her biri 2,5 olarak verilen kazanç değerleri manuel olarak değiştirilerek çalışmanın hız aralığı olan 10 m/s ve 30 m/s aralığında test edilerek birbirleri ile senkronize olacak şekilde kazanç değerleri belirlenmiştir. Yapılan optimizasyon sonucunda denetleyici kazancı 50, yaw rate kazancı 0,02 ve direksiyon açısı kazancı 0,037 olduğunda optimum uygunluk elde edildiği görülmüştür.

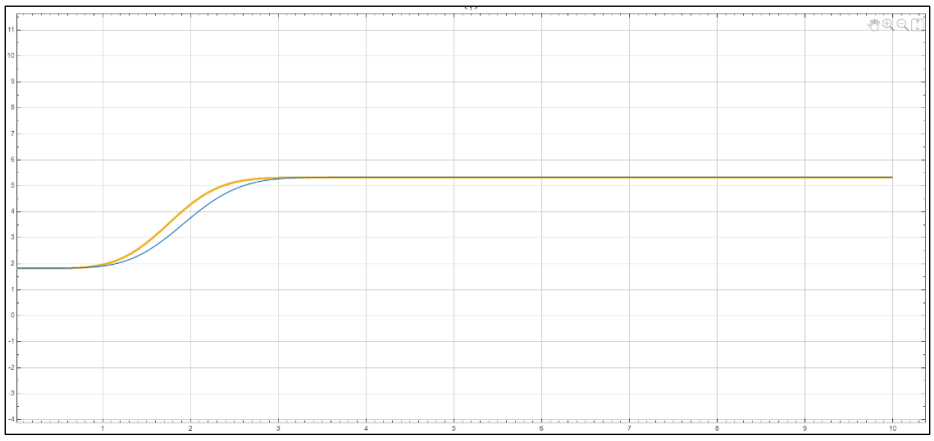


Şekil 3.18. Dinamik bisiklet modeli stanley denetleyici parametreleri

Optimizasyon esnasında uygunluk değerlendirmesi yapılırken gerçek yaw rate ile referans yaw rate değerleri bir grafikte, gerçek rota ile referans rota bir grafikte incelenerek referans değerlere yakınsama sağlanmaya çalışılmıştır.

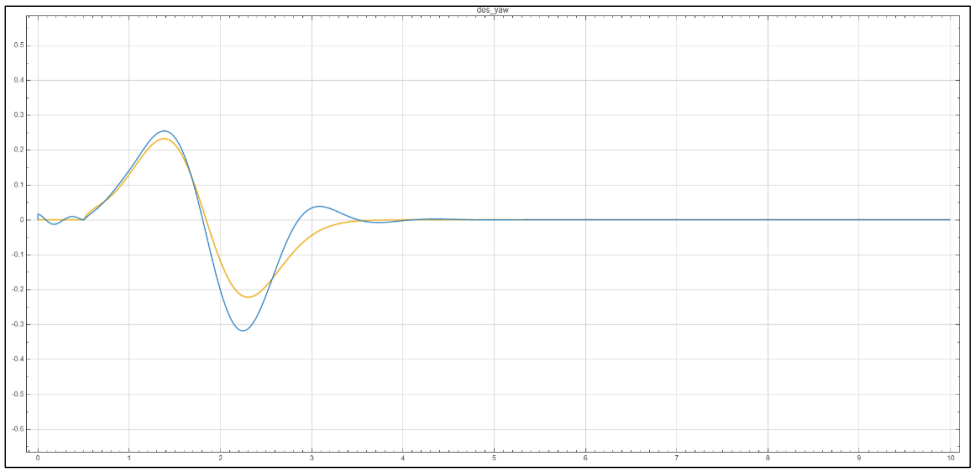


Şekil 3.19. 10 m/s için gerçek yaw rate ile referans yaw rate grafikleri

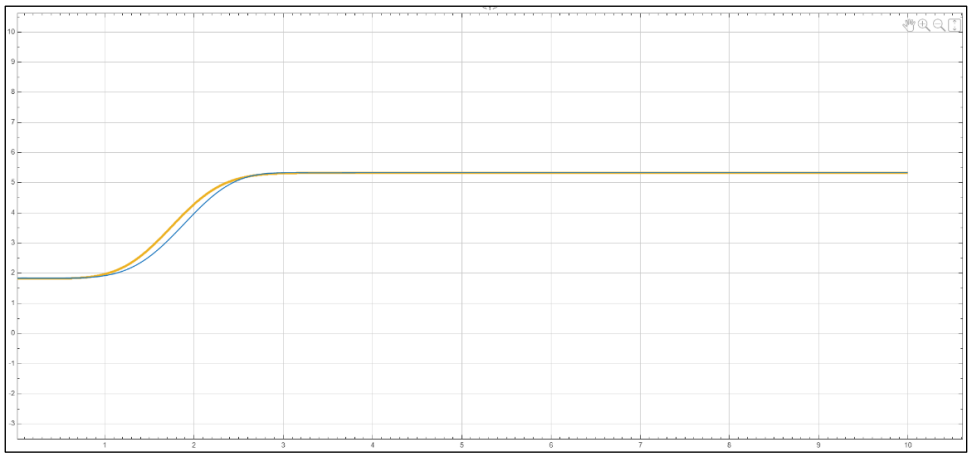


Şekil 3.20. 10 m/s için gerçek rota ile referans rota grafikleri

Şekil 3.19 ve Şekil 3.20'de verilen grafiklerde yaw rate ve rota çıktıları için gerçek ve referans değerler 10 m/s hız değeri koşulunda karşılaştırılmıştır. Referans ve gerçek değer grafikleri incelendiğinde rotanın büyük oranda referansa uygun olarak takip edildiği görülmüştür. Yaw rate değerleri için şerit değiştirme başlangıcında değerlerde minimal bir dalgalanma görülse de hareketin devamında referans yaw rate değerlerine büyük oranda yaklaşıldığı tespit edilmiştir.

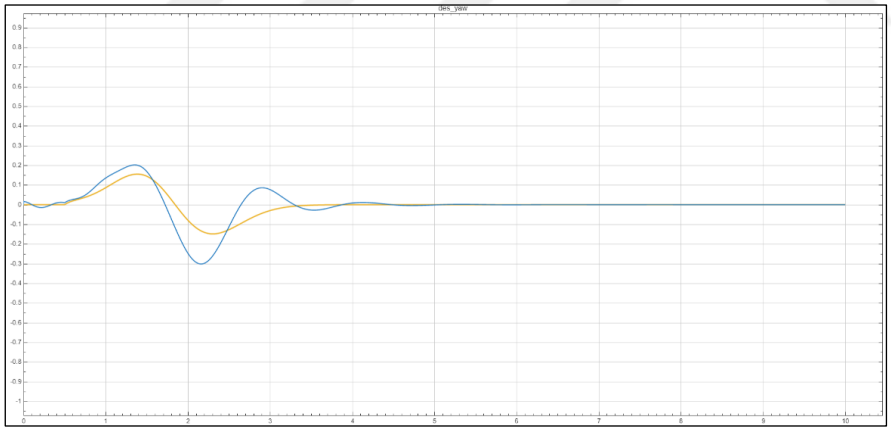


Şekil 3.21. 20 m/s için gerçek yaw rate ile referans yaw rate grafikleri

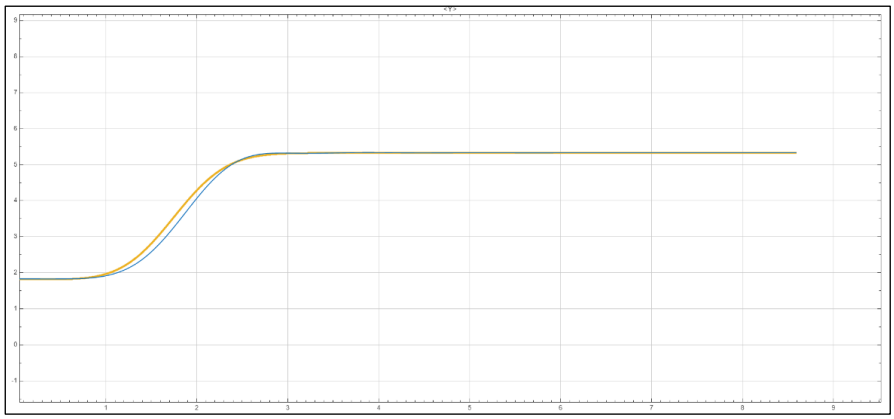


Şekil 3.22. 20 m/s için gerçek rota ile referans rota grafikleri

Şekil 3.21 ve Şekil 3.22'de verilen grafiklerde yaw rate ve rota çıktıları için gerçek ve referans değerler 20 m/s hız değeri koşulunda karşılaştırılmıştır. Gerçek rotanın bu hız değerinde de daha düşük hız değerlerinde olduğu gibi referans rotaya yaklaştığı görülmüştür. Yaw rate değeri için şerit değişimi öncesi dalgalanma ile beraber hareketin ilerleyen safhalarında da bir miktar sapma gerçekleştiği gözlemlenmiştir.



Şekil 3.23. 30 m/s için gerçek yaw rate ile referans yaw rate grafikleri



Şekil 3.24. 30 m/s için gerçek rota ile referans rota grafikleri

Şekil 3.23 ve Şekil 3.24'te verilen grafiklerde yaw rate ve rota çıktıları için gerçek ve referans değerler 30 m/s hız değeri koşulunda karşılaştırılmıştır. Hız değeri önceki gözlemlere göre daha yüksek olsa da gerçek rotanın referans rotaya uygunluğu açısından herhangi bir sapma ya da ıraksama problemi gözlemlenmemiştir. Diğer bir deyişle izlenen rota referans rotaya oldukça yakın konum ve açı değerlerine sahiptir. Yaw rate değeri bu hız şartında düşük hızlardan daha fazla sapma eğilimi göstermiştir. Sapma ve dalgalanma eğilimlerinin bu hız değeri için daha fazla olduğu görülmüştür. Yaw rate değerindeki sapmalar ve dalgalanmalar için taşıt modelinin daha yüksek serbestlik derecelerinde modellenmesi bir çözüm olarak sunulabilir.

4. GELİŞTİRİLEN OTONOM ŞERİT DEĞİŞTİRME ALGORİTMASI EĞİTİMİ VE TEST EDİLMESİ

Pekiştirmeli öğrenme ajanı için gerekli gözlem kümesi, aksiyon kümesi ve ödül fonksiyonu parametreleri tanımlandıktan sonra ajanın beklenen davranışı öğrenebilmesi için bir eğitim süreci gerekir. Ajanın eğitimi başlangıç durumunun çevreden alınarak okunması, mevcut duruma göre bir aksiyon seçilmesi, seçilen aksiyonun çevreye uygulanması, yeni durum ve ödülün alınması, Q-değerleri veya politika parametrelerinin güncellenmesi ve yeni durumla bir sonraki adımın gerçekleştirilmesi gibi adımlardan meydana gelir. Bu süreç birçok epizot (episode) boyunca tekrar edilir ve ajan deneyimlerini pekiştirerek en doğru politikayı öğrenmeye çalışır.

4.1. Eğitim Ortamı ve Parametreleri

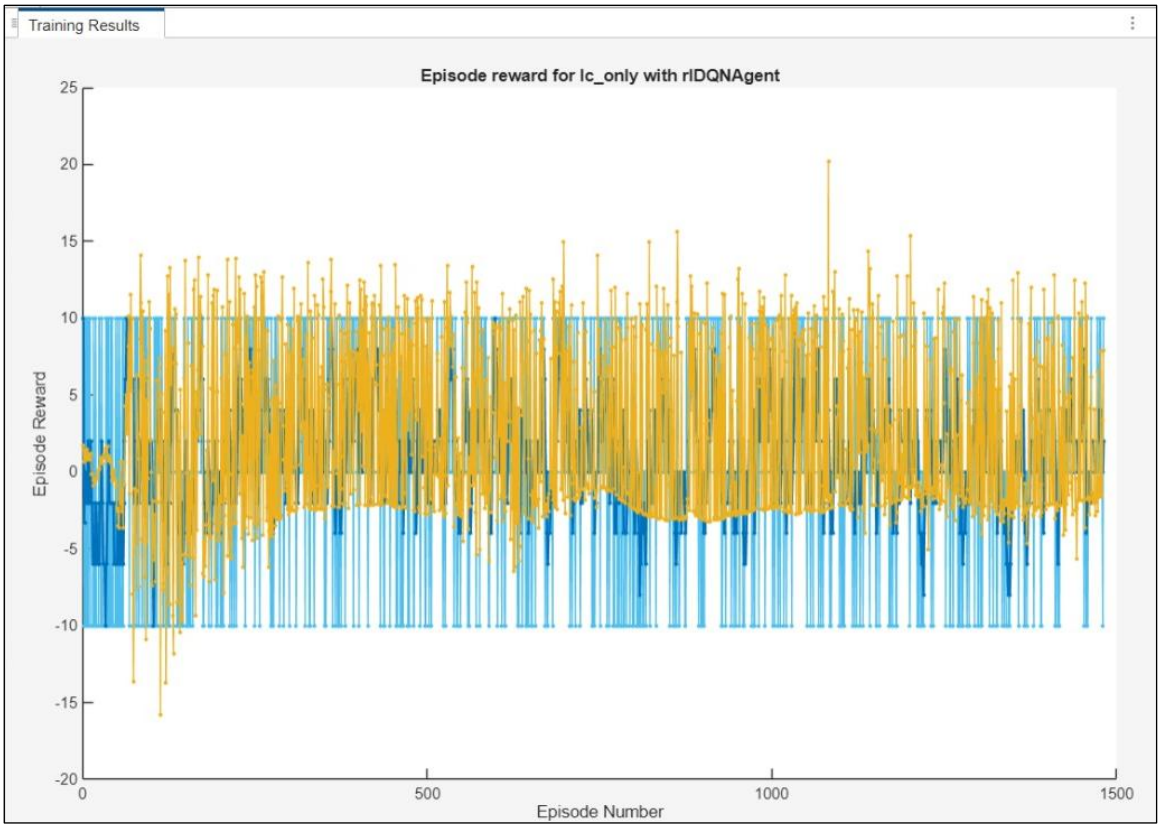
Çalışmada otonom araç için karar verme algoritması DQN yöntemi ile oluşturulmuştur. Ajan için gerekli gözlem, aksiyon, ödül gibi bilgiler sağlanarak aksiyon elde edilmiştir. Eğitim süreci için Matlab içerisindeki Reinforcement Learning Designer modülü kullanılmıştır. DQN ajanı için 1 000 bölüm olacak şekilde eğitim süreci tanımlanmıştır. Eğitim grafiği oluşturulurken Reinforcement Learning Designer içerisindeki eğitim ekranı kullanılmıştır.

4.1.1. DQN ajanı eğitimi

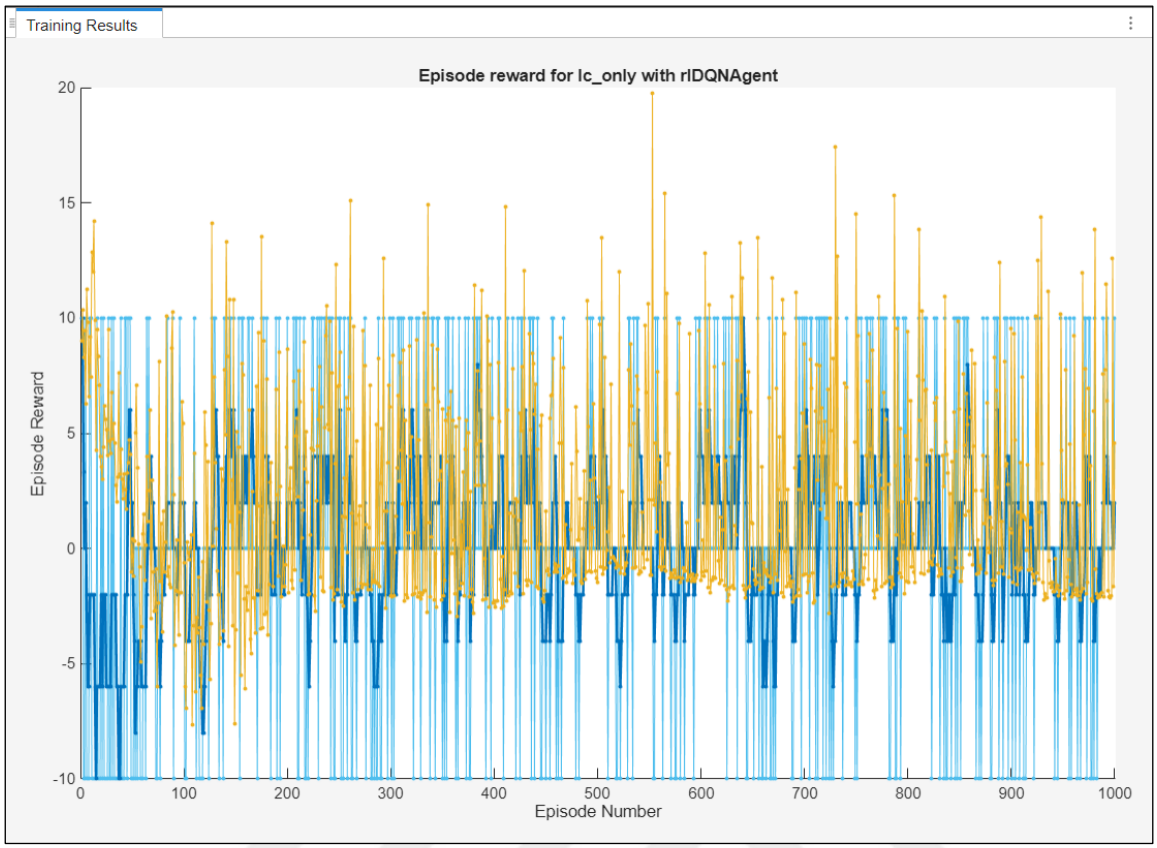
Karar verme mekanizması için oluşturulan DQN ajanının eğitilmesi için Matlab modüllerinden biri olan Reinforcement Learning Designer uygulaması kullanılmıştır. Mevcut çalışmada DQN ajanı için iki gizli katmanlı ve her katmanda 15 nöron bulunan ağ yapısı tercih edilmiştir. Bu ağdaki katman sayısı belirlenirken derin sinir ağlarına uygun ve sistemin olabildiğince hızlı eğitilebilmesi faktörleri göz önüne alınmıştır. Bu kapsamda ağ, bir derin sinir ağının sahip olabileceği en düşük gizli katman sayısı olan 2 gizli katman ile oluşturulmuştur. Ağlarda kullanılan nöron sayıları ise farklı nöron sayılarının ağlara işlenerek eğitilmesi ile optimum sayı belirlenerek son haline getirilmiştir.

Şekil 4.1 ve 4.2'de görüldüğü üzere eğitim grafikleri benzerlik göstermekle beraber eğitim süreleri arasında önemli farklılıklar bulunmaktadır. Her bir eğitim için gizli katman sayısı

sabit tutulup nöron sayısı güncellenmiştir. Her bir ağ yapısı için eğitim süresi 15-25 saat aralığında gerçekleştiği görülmüştür. Buna ek olarak, grafiklerde sarı renkte görülen ajanın seçtiği eyleme göre Q0 tahminini gösteren değer 11 ve 17 nöronlu yapılarda ani yükselişler ve sürecin normalinin üzerinde beklenti değerleri üretmiştir. Bu durum öğrenme davranışının dengeli olmadığını göstermektedir.



Şekil 4.1. 11 nöronlu ağ eğitimi



Şekil 4.2. 17 nöronlu ağ eğitimi

Modelde kullanılan DQN ajanının doğru ve optimal öğrenmeyi sağlayabilmesi için uygun parametre seçimleri yapılarak eğitim gerçekleştirilmiştir. Eğitim sürecinin en önemli noktalarından olan eğitim parametreleri Çizelge 4.1 ile verilmiştir.

Çizelge 4.1. Pekiştirmeli öğrenme DQN ajanı eğitim parametreleri

Maksimum Bölüm Sayısı	1 000
Her Bölümdeki Adım Sayısı	100
Learning Rate	0,01
Discount Factor	0,99
Exploration Rate	1
Epsilon Decay	0,005
Epsilon Min	0,01
Batch Size	64

Çizelgede verilen parametrelerden learning rate (öğrenme oranı) ajan tarafından elde edilen yeni bilgilerin mevcut bilgiye etkisinin derecesini belirleyen temel bir hiperparametredir. Diğer bir deyişle ajanın yeni deneyimlerden ne ölçüde etkileneceğini belirler. Q-learning ve alt dallarında Q-değeri güncelleme denkleminin kritik bir bileşenidir. Öğrenme oranı ne kadar yüksek olursa, ajan yeni deneyimlerden o kadar fazla etkilenir. Düşük öğrenme oranı

ise daha istikrarlı, ancak daha yavaş bir öğrenme sürecine neden olur. Uygun öğrenme oranı değeri hem öğrenme hızını hem de öğrenme kararlılığını etkileyerek eğitim sürecinin başarısını doğrudan belirler. Mevcut modelde seçilen 0,01 değeri eğitim sırasında istikrarlı bir öğrenme sağlayacak şekilde belirlenmiştir. Daha düşük değerler öğrenmeyi yavaşlatabilirken, daha yüksek oranlar ağın kararsız davranmasına sebep olabilmektedir (Sutton ve Barto, 1998).

İskonto faktörü ajan tarafından gelecekte elde edilecek ödüllerin bugünkü değerini ifade eder. Bu faktör 0 ile 1 aralığında bir değerdir ve bu parametrenin büyüklüğü ajanın uzun vadeli hedeflere ne ölçüde önem verdiğini gösterir. Buna bağlı olarak iskonto faktörü 0'a yaklaştıkça algoritma kısa vadeli ödüllere odaklanırken, 1'e yaklaştıkça daha uzun vadeli ödüllere odaklanır. Bu parametrenin yüksek olması genellikle daha iyi genel performans sağlar ancak öğrenme sürecini yavaşlatabilir. Eğitim için belirlenen 0,99 gibi yüksek bir değer ajanın uzun vadeli ödüllere önem vermesini sağlar. Seçilen bu değer bu çalışmada odaklanılan otonom sürüş gibi sistemlerde güvenlik odaklı davranılarak ajanın daha sabırlı ve sürdürülebilir politikalar öğrenmesine yardımcı olur (Kaelbling ve diğerleri, 1996).

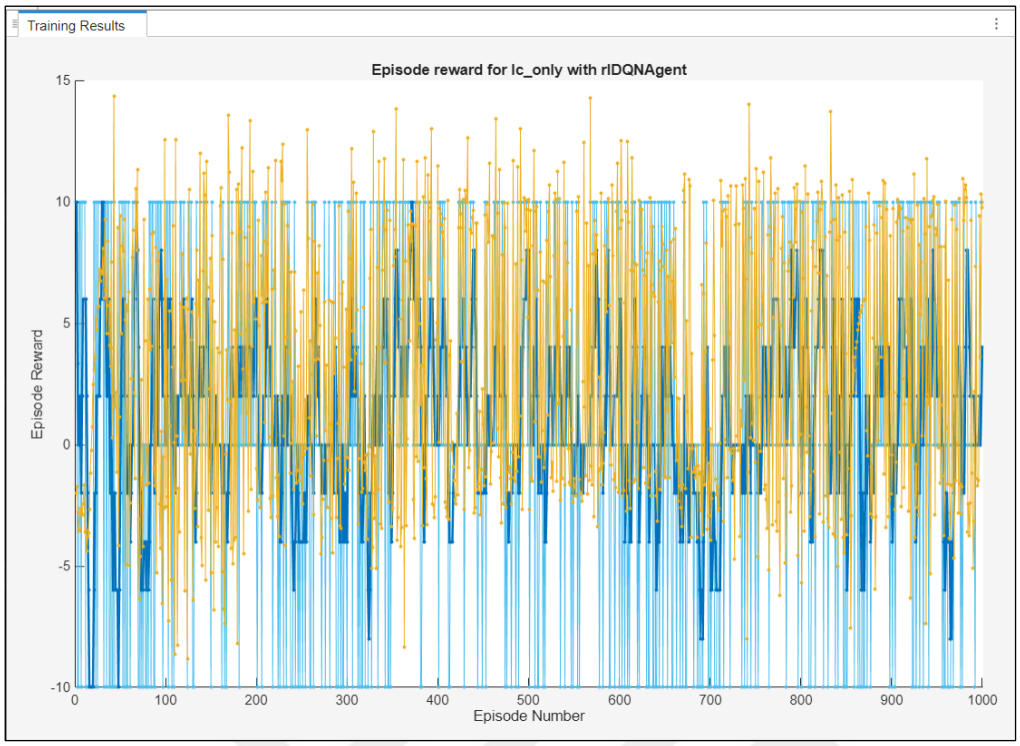
Ajanın öğrenme sürecinde kaç adet geçmiş deneyimi tek seferde kullanacağını belirleyen parametre batch size değeridir. Özellikle deneyim tekrarı (experience replay) kullanılan derin pekiştirmeli öğrenme algoritmalarında, batch size parametresi eğitim sürecinin istikrarı ve verimliliği açısından belirleyici bir roldedir. Küçük batch size değerleri daha sık güncellemeler sağlasa da dalgalanmaların ve gürültülerin olduğu bir öğrenme sürecine neden olabilir. Büyük batch size değerleri ise daha kararlıdır ancak hesaplama maliyeti artar ve genelleme yeteneği düşebilir. Mevcut çalışmada kullanılan 64’lük batch size boyutu literatürde hem istikrarlı tahminler hem de yeterli hesaplama verimliliği açısından optimum değerlerden biridir (Mnih ve diğerleri, 2015; Schulman ve diğerleri, 2015).

Exploration rate (keşif oranı) ajanın bir karar verirken hangi yoğunlukta rastgele ya da keşif odaklı aksiyon seçeceğini belirler. Epsilon (ε) ile ifade edilir. Keşif oranı başlangıçta genellikle yüksek tutulur ve zamanla azaltılır. Bu yaklaşım, ajanın öğrenme sürecinde yeterli çeşitlilikte deneyim edinmesini sağlar. Pekiştirmeli öğrenme problemlerinde keşif (exploration) ve sömürü (exploitation) arasında denge kurulması gerekmektedir. Epsilon-greedy stratejisi bu dengeyi sağlamak amacıyla sıklıkla kullanılan bir yöntemdir. Bu yöntemle ajanın rastgele bir eylem seçme olasılığı ε olarak belirlenir ve bu ölçüde keşif

yapar. En yüksek değerli eylemi seçme olsaılığı ise (1 - ε) olarak belirlenir ve bu ölçüde mevcut bilgileri sömürür. Başlangıçta keşif oranı değerinin 1 olarak belirlenmesi ajanın ilk adımlarda çevreyi tamamen keşfetmeye odaklanmasını sağlar (Sutton ve Barto, 1998).

Epsilon decay parametresi ile keşif oranının zamanla düşürülmesi sağlanır. Zamanla keşif oranının azalarak ajanın daha kararlı ve bilgiye dayalı seçimler yapması beklenir. Bu süreç eğitim sürecinin başlangıcında yoğunlukla keşif yapılması, ilerleyen aşamalarda ise sömürüye ağırlık verilmesi prensibi ile çalışır. Keşif oranının her adımda 0,005 oranında azalması keşif-sömürü dengesinin zamanla daha istikrarlı bir şekilde sağlanmasına yardımcı olmakla beraber literatürde yaygın olarak tercih edilen değer aralığında yer almaktadır (François-Lavet ve diğerleri, 2018).

Keşif oranı değerinin zamanla sıfıra yaklaşması, ajanın tamamen sömürüye odaklanıp keşif yapmamasına neden olabilir. Ancak bu durum ajanın yer yer ani yükselmelere maruz kalması riskine neden olur. Bu nedenle keşif oranının belirli bir alt sınırda tutulması sistemin sağlıklı işleyişi bakımından gereklidir. Minimum keşif oranı değeri (ε_{min}), ajan eğitim sürecinin sonlarına gelinmiş olsa bile zaman zaman rastgele eylemler seçerek keşif yapmasını sağlar. Bu strateji eğitim sürecinin sonunda da ortamın dinamiklerine karşı duyarlılığı korumak açısından önemlidir. Bu çalışmada kullanılan keşif oranı değerinin zamanla 0,01'e düşürülmesi ajanın eğitim sürecinin sonlarında hâlâ az da olsa keşif yapabilmesini sağlar (Mnih ve diğerleri, 2015; Sutton ve Barto, 1998).



Şekil 4.3. Dqn ajanı eğitim süreci

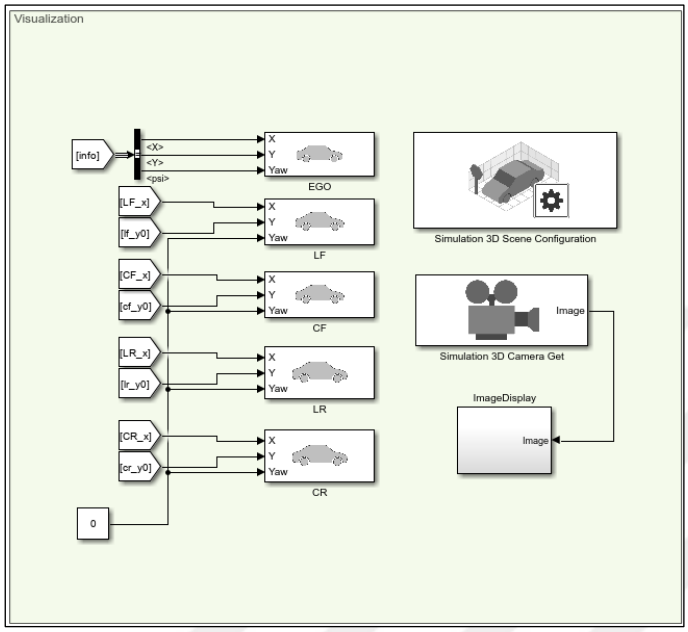
Eğitim sonucunda Şekil 4.3.'te görülen grafik elde edilmiştir. Deneme eğitimleri yapılan 11 ve 17 nöronlu ağlardan farklı olarak beklenmeyen Q0 çıkışları ve düşüşleri olmadığı görülmüştür. Bu durum ajanın etkin ve stabil bir öğrenme süreci gerçekleştirdiğini, eğitim sürecinin sistemden beklenen başarıyı sağlayacağını gösteren parametrelerdendir. Seçilen aksiyona bağlı olarak ajan -10,-5, 0 ve +10 ödül değerlerini almıştır. Mevcut durumlara göre seçtiği eylemler sonucunda doğru konum ve hızda, doğru direksiyon açısı ile, yol sınırları dışına çıkmadan şerit değişimi yapabilecek noktaya ulaşmıştır.

4.2. Eğitilen Ajanın Test Edilmesi

Pekiştirmeli öğrenme ile karar veren ve uygulayan sistemin hangi senaryo durumlarında hangi kararı verdiği ve nasıl bir sonuç aldığının gözlenebilmesi için Simulink modeli içerisine 3 boyutlu bir simülasyon görselleştirme yapılandırması entegre edilmiştir. Bu yapılandırmada simülasyon çalıştırıldığında taşıtın hareketi gözlemlenerek doğruluğu incelenmiştir.

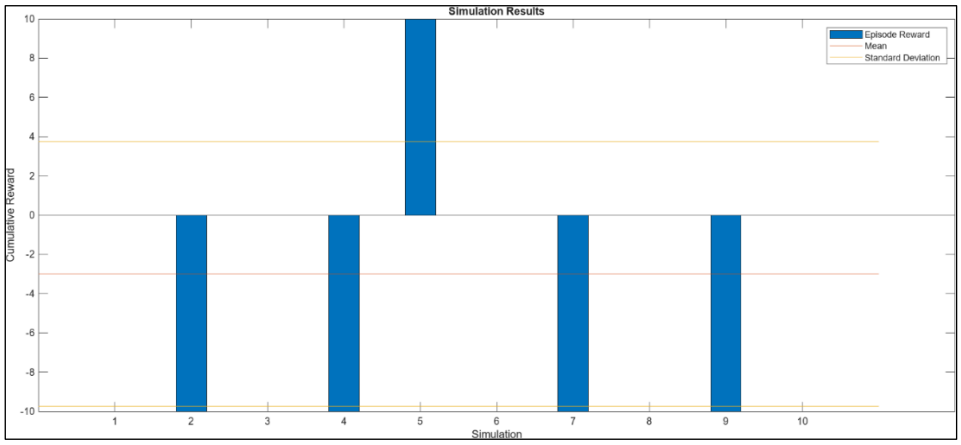
DQN ajanının Reinforcement Learning Designer ortamında eğitilmesinin ardından aynı ortamda bulunan simülasyon eklentisi ile sürekli simülasyon tekrarları gerçekleştirilmiştir.

Simulink modeline entegre edilen üç boyutlu görüntüleme modülü oluşturularak eğitimin amacına ulaşıp ulaşmadığı kontrol edilmiştir. Gerekli konum ve hız bilgileri bu modüle aktarılarak taşıtın simülasyon içerisindeki hareketleri görselleştirilmiştir. Görselleştirme için kullanılan modül yapısı Şekil 4.4 ile verilmiştir.



Şekil 4.4. Simulink simülasyon görselleştirme modülü

Karşılaştırma yapılabilmesi için öncelikle eğitilmemiş ajana ait bilgiler ile simülasyon başlatılarak taşıt hareketleri gözlenmiştir. Eğitilmemiş ajan simülasyon esnasında rastgele doğru kararlar verebilmiş olsa da şerit değiştirme davranışını doğru anlarda, konumlarda ve açılarda gerçekleştiremediği görülmüştür.



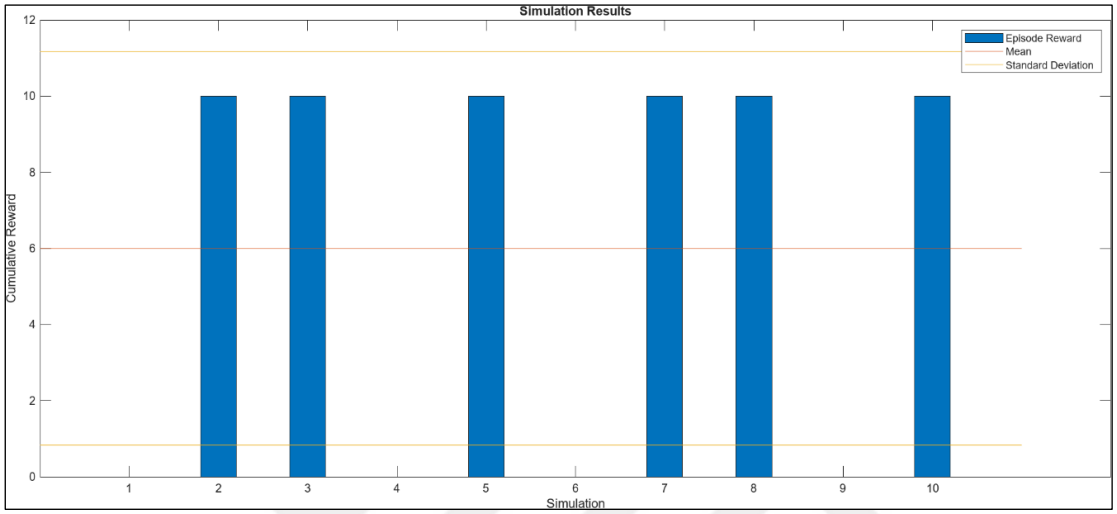
Şekil 4.5. Eğitilmemiş ajan simülasyon grafiği

Eğitilmemiş ajan ile üç boyutlu simülasyon görüntüleri de Şekil 4.5'te görülen grafikle paralel çıktılar sunmuştur. Taşıtın uygun ve gerekli koşullarda şerit değiştirmediği ya da Şekil 4.6 ve Şekil 4.7.'de görüldüğü üzere uygun olmayan koşullarda şerit değişimi yaparak kaza gerçekleştiği görülmüştür.

Şekil 4.6. Eğitilmemiş ajan kaza durumu

Şekil 4.7. Eğitilmemiş ajan kaza durumu

Eğitilmiş ajanın, eğitimin yapıldığı Reinforcement Learning Designer ortamında gerçekleştirilen sürekli simülasyon tekrarına ait grafik Şekil 4.8.'de görülmektedir. Eğitilmiş ajanın şerit değiştirme kararını doğru konum ve hızlarda verdiği, buna bağlı olarak pozitif ödüller alarak hareketi doğru ve beklenen biçimde gerçekleştirdiği saptanmıştır.



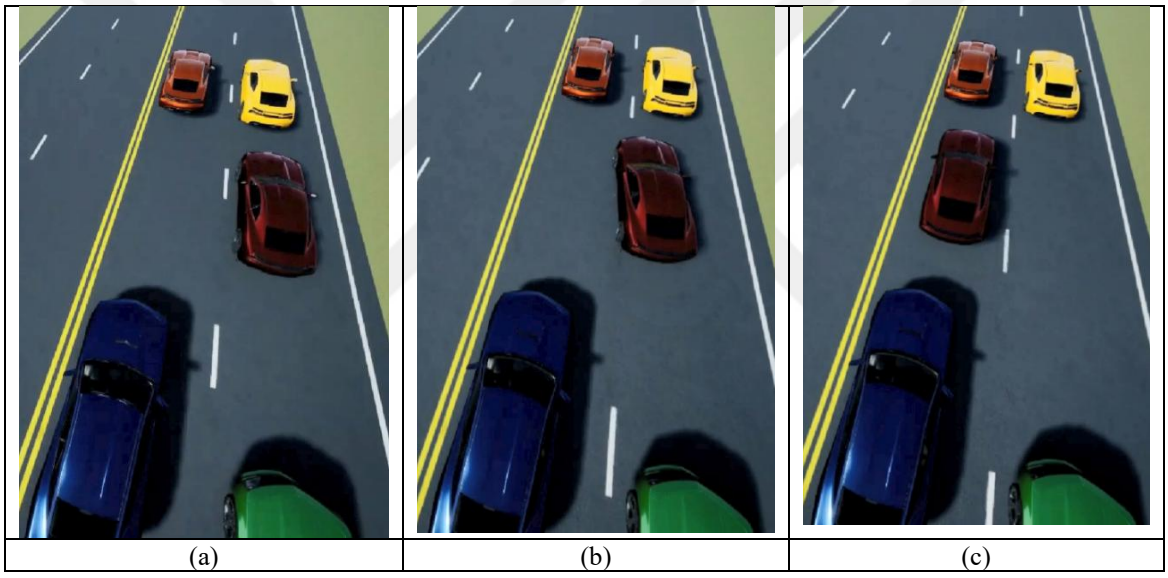
Şekil 4.8. Eğitilen dqn ajanı simülasyon grafiği

Ajan eğitildikten sonra gerekli tanımlamalar yapılarak yeniden üç boyutlu görüntüleme modülü ile taşıt hareketleri gözlemlenmiştir. Farklı senaryolarda manuel olarak simülasyon başlatılarak mevcut senaryodaki aksiyon seçimine bağlı hareketler gözlemlenmiştir.

Şekil 4.9. Eğitilen ajan şeritte kalma davranışı

Şekil 4.9’da görüldüğü üzere eğitilmiş ajan rastgele olarak belirlenen değerler üzerine üretilen senaryoda, uygun olmayan konum şartlarında şerit değiştirme kararı vermeyerek bulunduğu şeridi korumuş, kaza ve yoldan çıkma gibi istenmeyen durumların önüne geçmiştir.

Rastgele belirlenen konum ve hız şartları ile oluşan senaryo şerit değişimine uygun olduğunda eğitilmiş DQN ajanı şerit değiştirme aksiyonunu seçmiş ve uygun rota ile şerit değişimi sağlanmıştır. Şerit değişimi esnasında herhangi bir yönelim hatası, kaza durumu ya da yol sınırları dışına çıkma davranışı gözlemlenmemiştir. Eğitilmiş ajanın gerçekleştirdiği şerit değişimi süreci Şekil 4.10 ile gösterilmiştir.



(a) (b) (c)

Şekil 4.10. Eğitilen ajan şerit değiştirme süreci

Şekil 4.10 (a) aşamasında sistem gözlem olarak kullandığı çevre araçlarla mesafe değerlerinin uygun olduğuna karar vererek şerit değiştirme hareketini başlatır. Şerit değiştirme kararının verilmesinin ardından rota oluşturulur ve Stanley denetleyici ile taşıtın rotayı takip etmesi sağlanır. Şekil 4.10 (b)’de görüldüğü üzere rota oluşturulmuş ve taşıt referans rotayı takip etme sürecine başlamıştır. Şekil 4.10 (c)’de ise şerit değiştirme hareketinin sonlandırıldığı görülmektedir. Taşıt verilen rastgele senaryo şartlarında şerit değiştirme kararı vermiş, rota oluşturulmuş ve taşıt rotayı başarılı bir şekilde takip etmiştir. Şerit değişimi esnasında çevre araçlarla herhangi bir kaza durumu ya da şeritten çıkma gibi bir durum gözlenmemiştir. Bu simülasyon seçilen parametrelerin ve gerçekleştirilen DQN ajanı eğitiminin başarılı bir şekilde sonuçlandığını göstermektedir.

5. SONUÇ VE DEĞERLENDİRME

Bu tez çalışmasında, otonom taşıtlar için güvenli şerit değiştirme kararlarının alınmasına yönelik bir kontrol yaklaşımı geliştirilmiştir. Taşıtın çevresindeki dört farklı aracı göz önünde bulundurarak gerektiğinde uygun manevralarla şerit değiştirebilmesi hedeflenmiştir. Bu amaçla, pekiştirmeli öğrenme tabanlı bir karar verme sistemi Simulink ortamında modellenmiş, taşıt dinamikleri 3 serbestlik dereceli bir tek izli taşıt modeli (3-DOF single track model) ile temsil edilmiştir. Karar verme aşaması için pekiştirmeli öğrenme yöntemlerinden DQN algoritması, rota oluşturma aşaması için Sigmoid fonksiyonu ve rota takibi için Stanley denetleyici kullanılmıştır.

Geliştirilen sistemde, aracın çevresindeki öndeki ve arkadaki araçlar (MÖ, MA, HÖ, HA) ayrı ayrı analiz edilmiş ve her biri için dinamik olarak değişen mesafeler hesaplanmıştır. Her bir simülasyonda değişen bu mesafeler sayesinde gerçek trafik koşullarının dinamikliği ve değişkenliği sisteme entegre edilmeye çalışılmıştır. Sistem, Simulink üzerinde modellenmiş, gerekli modüller ve bloklar kullanılarak oluşturulan çok şeritli bir yol ortamında test edilmiş ve gerçek sürüş senaryolarına yakın bir yapı kurulmuştur. Simülasyon ortamında araç öncelikle ayrık bir aksiyon uzayı kullanarak şerit değişimi aksiyonunu belirlemiş, şerit değiştirme kararının verildiği durumda aktifleşen rota planlama sisteminin ardından direksiyon açısının belirlenebilmesi için Stanley denetleyici ile rota takibini gerçekleştirmiştir.

Eğitim sürecinde DQN algoritması için uygun ağ yapısı Deep Network Designer kullanılarak yapılandırılmıştır. DQN ajanı için gözlem kümesi, otonom taşıta ait hız ile otonom aracın çevre araçlarla merkez mesafeleri farkı olmak üzere beş girdiden oluşmaktadır. Aksiyon kümesi ise 0 (şeritte kal) ve 1 (şerit değiştir) olmak üzere iki elemandan oluşmuştur.

Simülasyonlar sonucunda geliştirilen modelin, önceden belirlenen dinamik trafik senaryolarına karşı etkili şekilde tepki verebildiği gözlemlenmiştir. Otonom aracın sol şeridin konum ve hıza bağlı olarak daha avantajlı olduğu durumlarda uygun direksiyon açısı ile güvenli ve konforlu şerit değişimi hareketi yaptığı görülmüştür. Aynı zamanda gerçekleştirilen simülasyonlar sistemin gereksiz şerit değişimlerinden kaçınarak kaza riskini minimize ettiğini; güvenli ve konforlu sürüş profilini devam ettirdiğini ortaya koymuştur. Eğitim süreci boyunca kullanılan ödül fonksiyonu, kaza durumlarının önlenmesi, gereksiz

manevralardan kaçınılması ve ilerleme odaklı ödülleri analiz ederek çok boyutlu bir davranış geliştirilmesini sağlamıştır.

Bununla beraber, çalışmanın bazı sınırlılıkları da bulunmaktadır. Öncelikle, eğitim ortamı yalnızca simülasyon temelli olduğu için, gerçek yol verileriyle test edilmemiştir. Gerçek dünyadaki algılayıcı hataları, yol yüzeyi değişimleri, insan faktörü gibi parametreler sistemin davranışını etkileyebilecek faktörler arasında yer almaktadır. Ayrıca, eğitim süreci boyunca yalnızca sınırlı sayıda senaryo üzerinden değerlendirme yapılmış; farklı yol topolojileri, trafik yoğunlukları ve hava koşulları gibi değişkenler sisteme dahil edilmemiştir. Bununla birlikte, kullanılan taşıt dinamiği modeli her ne kadar birçok uygulama için yeterli olsa da daha karmaşık dinamiklerin (örneğin lastik-sürtünme modelleri, eğim değişimleri) dikkate alınması durumunda kontrol performansında bazı farklılıklar gözlemlenebilir.

Gelecek çalışmalar için çeşitli geliştirme önerileri sunulabilir. Öncelikle sistemin, gerçek sensör verileriyle (örneğin LIDAR, RADAR, kamera) beslenecek bir yapay algılama modülüyle entegre edilmesi planlanabilir. Bu sayede sistem, gerçek zamanlı olarak hem çevre algısı hem de karar üretimi açısından daha gerçekçi senaryolarda test edilebilir hale gelir. Ayrıca sistemin yalnızca şerit değiştirme değil, kavşak geçişi, sollama, dar yolda geçiş gibi daha karmaşık sürüş senaryolarına uyarlanması da mümkündür. Eğitim sürecinde daha gelişmiş pekiştirmeli öğrenme algoritmaları (Soft Actor-Critic (SAC), Proximal Policy Optimization (PPO) gibi) ile karşılaştırmalı analizler yapılması, performansın daha da iyileştirilmesine olanak tanıyabilir.

Sonuç olarak bu çalışma, otonom taşıtlarda güvenli, çevresel farkındalığa sahip, karar alma ve hareketin uygulanması süreçlerinde öğrenmeye dayalı bir kontrol mimarisinin başarıyla uygulanabileceğini göstermiştir. Derin pekiştirmeli öğrenme yöntemlerinin, klasik kural tabanlı sistemlerin ötesine geçerek dinamik trafik ortamlarında efektif kararlar verebilen sürüş sistemlerinin geliştirilmesinde güçlü bir alternatif olduğu görülmüştür. Gerek sistem mimarisi gerekse kullanılan algoritmalar açısından modüler bir yapı sunan bu modelin, otonom sürüş alanında yapılan diğer çalışmalara hem metodolojik hem de deneysel anlamda katkı sağlaması hedeflenmektedir.

KAYNAKLAR

Alam, M (2023). *Data driven control for marine vehicle maneuvering*. Yayımlanmamış Yüksek Lisans Tezi, Indian Institute of Technology Madras, Chennai, 85-96.

Arulkumaran, K., Deisenroth, M. P., Brundage, M. ve Bharath, A. A (2017). *Deep reinforcement learning: A brief survey.* IEEE Signal Processing Magazine, 34(6), 26–38.

Basile, G., Leccese, S., Petrillo, A., Rizzo, R. ve Santini, S (2024). *Sustainable DDPG-based path tracking for connected autonomous electric vehicles in extra-urban scenarios.* IEEE Transactions on Industry Applications, 60(6), 9237–9250.

Bellman, R. (1966). *Dynamic programming*. Science, 153(3731), 34–37.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Müller, U., Zhang, J., Zhang, X., Zhao, J. ve Zieba, K (2016). *End to end learning for self-driving cars.* arXiv preprint, 1, 07316.

Bre, F., Gimenez, J. M. ve Fachinotti, V. D (2018). *Prediction of wind pressure coefficients on building surfaces using artificial neural networks.* Energy and Buildings, 158(2), 1429–1441.

Bro, R. ve Smilde, A. K (2014). *Principal component analysis*. Analytical Methods, 6(9), 2812–2831.

Byrne, R. ve Abdallah, C (1995). *Design of a model reference adaptive controller for vehicle road following.* Mathematical and Computer Modelling, 22(4–7), 343–354.

Chen, Y., Li, L., Wei, L., Guo, Q., Du, Z., Xu, Z., Pajic, M., Zhong, L., Banerjee, S., Daily, S. ve Krolik, J (2024). *AI computing systems: An application driven perspective*. New York: Elsevier, 85-96.

Dayan, P. ve Watkins, C (1992). *Q-learning. Machine Learning,* 8(3), 279–292.

Ding, T., Li, X., Zheng, L. ve Di, Y (2019). *Research on safety lane change warning method based on potential angle collision point.* Journal of Advanced Transportation, 2019(1), 1281425.

Dixon, M., Klabjan, D. ve Bang, J. H (2017). *Classification based financial markets prediction using deep neural networks.* Algorithmic Finance, 6(3–4), 67–77.

Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E. ve Hrovat, D (2007). *Predictive active steering control for autonomous vehicle systems.* IEEE Transactions on Control Systems Technology, 15(3), 566–580.

François Lavet, V., Henderson, P., Islam, R., Bellemare, M. G. ve Pineau, J (2018). *An introduction to deep reinforcement learning*. Foundations and Trends® in Machine Learning, 11(3–4), 219–354.

Gherardini, L. ve Cabri, G (2024). *The impact of autonomous vehicles on safety, economy, society, and environment*. World Electric Vehicle Journal, 15(12), 579.

Gillespie, T (Ed.) (2021). *Fundamentals of vehicle dynamics*. Warrendale: SAE International, 78-85.

Goodfellow, I., Bengio, Y. ve Courville, A (2016). *Deep learning* (Vol. 1). Cambridge: MIT Press, 66-78.

Gunning, D. ve Aha, D (2019). *DARPA's explainable artificial intelligence (XAI) program.* AI Magazine, 40(2), 44–58.

Günay, U (2021). *Yapay sinir ağır tabanlı otonom park sistemi geliştirilmesi*. Yayımlanmamış Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 102-105.

Han, J. ve Moraga, C (1995). *The influence of the sigmoid function parameters on the speed of backpropagation learning.* In International Workshop on Artificial Neural Networks.

Hastie, T., Tibshirani, R. ve Friedman, J (2009). *The elements of statistical learning: Data mining, inference, and prediction* (Vol. 2). New York: Springer, 65-77.

He, X., Liu, Y., Wu, Q., Wang, J. ve Zhao, D (2022). *Robust lane change decision making for autonomous vehicles: An observation adversarial reinforcement learning approach.* IEEE Transactions on Intelligent Vehicles, 8(1), 184–193.

Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M. G. ve Silver, D (2018). *Rainbow: Combining improvements in deep reinforcement learning.* Proceedings of the AAAI Conference on Artificial Intelligence, 32(1), 3215–3222.

Huang, C., Li, M. ve Peng, Z (2016). *Model predictive control-based lane change control system for an autonomous vehicle.* In 2016 IEEE Region 10 Conference (TENCON), Singapur, 123–128.

Huang, X., Zhang, Y. ve Song, Y (2019*). A path planning method for vehicle overtaking maneuver using sigmoid functions.* IFAC-PapersOnLine, 52(8), 422–427.

Inagaki, T. ve Sheridan, T. B (2019). *A critique of the SAE conditional driving automation definition, and analyses of options for improvement.* Cognition, Technology & Work, 21, 569–578.

İnternet: Coulter, R. C (1992). Implementation of the pure pursuit path tracking algorithm (Tech. Rep. CMU RI TR 92 01). Robotics Institute, Carnegie Mellon University. Web: https://apps.dtic.mil/sti/html/tr/ADA255524/, Son Erişim Tarihi: 22/05/2025.

İnternet: Wienrich, J (2022). Autonomous driving: The steps to self-driving vehicles. Web: https://www.zf.com/mobile/en/technologies/automated_driving/stories/6_levels_of_automated_driving.html. Son Erişim Tarihi: 29/04/2025.

İnternet: ZF Technologies (2022). Automated driving: An overview. Web: https://www.zf.com/mobile/en/technologies/automated_driving/automated_driving.html. Son Erişim Tarihi: 29/04/2025.

Jiang, L., Wang, Y., Wang, L. ve Wu, J (2019). *Path tracking control based on deep reinforcement learning in autonomous driving.* In Proceedings of the 2019 3rd Conference on Vehicle Control and Intelligence (CVCI). China: Hefei, 1-6.

Jurafsky, D. ve Martin, J. H (2020). *Speech and language processing*. Draft of 3rd edition. New York: CRC Press, 12-25.

Kaelbling, L. P., Littman, M. L. ve Moore, A. W (1996). *Reinforcement learning: A survey.* Journal of Artificial Intelligence Research, 4, 237–285.

Kendall, A., Hawke, J., Janz, D., Mazur, P., Reda, D., Allen, J.-M., Lam, V. D., Bewley, A. ve Shah, A (2019). *Learning to drive in a day.* In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA). Montréal, 1-8.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S. ve Pérez, P (2021). *Deep reinforcement learning for autonomous driving: A survey*. IEEE Transactions on Intelligent Transportation Systems, 22(6), 4909–4926.

Kober, J., Bagnell, J. ve Peters, J (2013). *Reinforcement learning in robotics: A survey*. The International Journal of Robotics Research, 32(11), 1238–1274.

Konda, V. ve Tsitsiklis, J (1999). *Actor-critic algorithms.* Advances in Neural Information Processing Systems, 12.

Kreyszig, E (1991). *Differential geometry* (Vol. 11). New York: Courier Corporation, 78-85.

LeCun, Y., Bengio, Y. ve Hinton, G (2015). Deep learning. *Nature*, 521(7553), 436–444.

Lee, J. ve Choi, J. W (2019). *May I cut into your lane?: A policy network to learn interactive lane change behavior for autonomous driving.* In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*.

Li, P., Li, X., Zhu, P., Wang, X. ve Song, Y (2022). *Human-like motion planning of autonomous vehicle based on probabilistic trajectory prediction*. Applied Soft Computing, 118, 108499.

Li, T., Lv, C., Chang, X. ve Yin, Y (2020). *A cooperative lane change model for connected and automated vehicles*. IEEE Access, 8, 54940–54951.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D. ve Wierstra, D (2015). *Continuous control with deep reinforcement learning.* arXiv preprint, arXiv:1509.02971.

Lin, L.-J (1992). *Self-improving reactive agents based on reinforcement learning, planning and teaching.* Machine Learning, 8, 293–321.

Liu, Y., Hu, J. ve Sun, H (2019). *A novel lane change decision-making model of autonomous vehicle based on support vector machine*. IEEE Access, 7, 26543–26550.

Mirchevska, B., Gong, X. ve Sobh, I (2018). *High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning.* In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, 1–6.

Mirchevska, B., Gong, X. ve Sobh, I (2018). *High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning*. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems *(ITSC)* (pp. 1–6).

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. ve Hassabis, D (2015). *Human level control through deep reinforcement learning.* Nature, 518(7540), 529–533.

Nagarajan, K. ve Yi, Z (2021). *Lane changing using multi-agent DQN*. In 2021 IEEE International Conference on Autonomous Systems (ICAS), Online.

Naranjo, J. E., González, C., García, R. ve de Pedro, T (2008). *Lane change fuzzy control in autonomous vehicles for the overtaking maneuver*. IEEE Transactions on Intelligent Transportation Systems, 9(3), 438–450.

Paden, B., Čáp, M., Yong, S. Z., Yershov, D. ve Frazzoli, E (2016). *A survey of motion planning and control techniques for self-driving urban vehicles.* IEEE Transactions on Intelligent Vehicles, 1(1), 33–55.

Peng, J., Zhang, S., Zhou, Y. ve Li, Z (2022). *An integrated model for autonomous speed and lane change decision making based on deep reinforcement learning*. IEEE Transactions on Intelligent Transportation Systems, 23(11), 21848–21860.

Puterman, M. L (2014). *Markov decision processes: Discrete stochastic dynamic programming*. Hoboken: John Wiley & Sons, 55-65.

Russell, S. ve Norvig, P (2021). *Artificial intelligence: A modern approach* (4th Global ed.). Harlow: Pearson, 45-52.

Sallab, A. E., Abdou, M., Perot, E. ve Yogamani, S (2017*). Deep reinforcement learning framework for autonomous driving.* arXiv preprint, arXiv:1704.02532.

Salvucci, D. D. ve Liu, A (2002). *The time course of a lane change: Driver control and eye-movement behavior.* Transportation Research Part F: Traffic Psychology and Behaviour, 5(2), 123–132.

Schulman, J., Levine, S., Moritz, P., Jordan, M. I. ve Abbeel, P (2015). *Trust region policy optimization.* In Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, 1889-1897.

Shortreed, S. M., Laber, E. B., Lizotte, D. J., Zhao, Y., Ni, L. ve Murphy, S. A (2011). *Informing sequential clinical decision making through reinforcement learning: An empirical study.* Machine Learning, 84, 109–136.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T. ve Hassabis, D (2016). *Mastering the game of Go with deep neural networks and tree search.* Nature, 529(7587), 484–489.

Sutton, R. S. ve Barto, A. G (1998). *Reinforcement learning: An introduction* (Vol. 1). Cambridge: MIT Press, 45-52.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. ve Rabinovich, A (2015). *Going deeper with convolutions*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, 1-9.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Jensen, S. ve Mahoney, P (2006). *Stanley: The robot that won the DARPA Grand Challenge*. Journal of Field Robotics, 23(9), 661–692.

van Hasselt, H., Guez, A. ve Silver, D (2016). *Deep reinforcement learning with double Q learning.* In Proceedings of the 30th AAAI Conference on Artificial Intelligence, Online.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. ve Polosukhin, I (2017). *Attention is all you need.* In Advances in Neural Information Processing Systems, 30.

Vázquez-Canteli, J. R. ve Nagy, Z (2019). *Reinforcement learning for demand response: A review of algorithms and modeling techniques.* Applied Energy, 235, 1072–1089.

Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M. ve Freitas, N (2016). *Dueling network architectures for deep reinforcement learning.* In Proceedings of the 33rd International Conference on Machine Learning (ICML), New York, 1995–2003.

Werling, M., Bennewitz, M. ve Burgard, W (2010). *Optimal trajectory generation for dynamic street scenarios in a Frenet frame.* In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA) (pp. 987–993).

Williams, R. J (1992). *Simple statistical gradient-following algorithms for connectionist reinforcement learning.* Machine Learning, 8, 229–256.

Xiaorui, W. ve Hongxu, Y (2013). *A lane change model with the consideration of car following behavior.* Procedia – Social and Behavioral Sciences, 96, 2354–2361.

Yavas, U., Tüfekçioğlu, H. ve Güven, E (2020). *A new approach for tactical decision making in lane changing: Sample efficient deep Q learning with a safety feedback reward.* In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, 1503–1508.

Yeong, D. J., Lux, A. ve Aspinall, D (2021). *Sensor and sensor fusion technology in autonomous vehicles: A review.* Sensors, 21(6), 2140.

Zhang, K., Li, X., Wang, Y. ve Zhu, Z (2024). *Kinematic model based lateral trajectory tracking control design and experimental validation for unmanned autonomous cars*. In Proceedings of the 2024 8th CAA International Conference on Vehicular Control and Intelligence (CVCI), Chongqing, 45–50.

Zhao, J., Li, X., Wang, Y. ve Zhang, Z (2024). *Autonomous driving system: A comprehensive survey*. Expert Systems with Applications, 242, 122836.

Zhao, X. ve Li, Y (2017). *Trajectory planning for 6 DOF robotic arm based on quintic polynomial*. In Proceedings of the 2017 2nd International Conference on Control, Automation and Artificial Intelligence (CAAI), Sanya, 120–125.

Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C. ve Thrun, S (2014). *Making Bertha drive—An autonomous journey on a historic route.* IEEE Intelligent Transportation Systems Magazine, 6(2), 8–20.

Gazili olmak ayrıcalıktır…