

**BOŞLUĞU TAKİP ET METODU İLE
OTONOM SOLLAMA MANEVRESİ TASARIMI**

YÜKSEK LİSANS TEZİ

Münire Damla DEMİR

Mekatronik Mühendisliği Ana Bilim Dalı

Mekatronik Mühendisliği Yüksek Lisans Programı

Tez Danışmanı: Doç. Dr. Volkan SEZER

HAZİRAN 2019

**BOŞLUĞU TAKİP ET METODU İLE
OTONOM SOLLAMA MANEVRA SİTASARI**

YÜKSEK LİSANS TEZİ

**Münire Damla DEMİR
(518151033)**

Mekatronik Mühendisliđi Ana Bilim Dalı

Mekatronik Mühendisliđi Yüksek Lisans Programı

Tez Danışmanı: Doç. Dr. Volkan SEZER

HAZİRAN 2019

İTÜ, Fen Bilimleri Enstitüsü'nün 518151033 numaralı Yüksek Lisans Öğrencisi Münire Damla DEMİR, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "BOŞLUĞU TAKİP ET METODU İLE OTONOM SOLLAMA MANEVRASI TASARIMI" başlıklı tezini aşağıdaki imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. Volkan SEZER**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Doç. Dr. Tufan KUMBASAR**
İstanbul Teknik Üniversitesi

Dr. Öğr. Üy. Janset DAŞDEMİR
Yıldız Teknik Üniversitesi

Teslim Tarihi : **03 Mayıs 2019**
Savunma Tarihi : **13 Haziran 2019**





Beni büyüten, gittikçe genişleyen aileme, Nil'e ve Yaz'a,



ÖNSÖZ

Hazırlık, teori ve uygulama aşamalarında hiç bir yardımını ve desteğini esirgemeyen tez danışmanım sayın Doç. Dr. Volkan Sezer'e teşekkürlerimi sunarım.

Daha sonra, benim her anımda yanımda olup hayatım boyunca bana destek olan ailem Abdullah Demir'e, Yurdanur Demir'e, Gaye Sertakar'a ve Ali Aydın Demir'e çok teşekkür ederim. Daha sonradan ailemize katılan Orgun Sertakar ve Gözde Demir'e iyi dilekleri için teşekkür ederim.

Son olarak, değerli meslektaşım ve nişanlım Hüseyin Yılmaz Yıldırım'a her zaman yanımda olduğu çok teşekkür ederim.

Haziran 2019

Münire Damla DEMİR
(Mekatronik Mühendisi)



İÇİNDEKİLER

	<u>Sayfa</u>
ÖNSÖZ	vii
İÇİNDEKİLER	ix
KISALTMALAR.....	xi
SEMBOLLER	xiii
ÇİZELGE LİSTESİ.....	xv
ŞEKİL LİSTESİ.....	xvii
ÖZET	xix
SUMMARY	xxiii
1. GİRİŞ	1
2. ARAÇ MODELİ.....	9
2.1 Kinematik Araç Modeli.....	9
2.2 Dinamik Araç Modeli.....	12
3. ENGELDEN KAÇMA ALGORİTMALARI	15
3.1 Böcek Algoritmaları	15
3.1.1 Böcek-1 algoritması.....	15
3.1.2 Böcek-2 algoritması.....	16
3.2 Yapay Potansiyel Alan Metodu	17
3.3 Boşluğu Takip Et Metodu.....	19
4. HAREKET PLANLAMA ALGORİTMALARI.....	23
4.1 Grafik Arama Tabanlı Algoritmalar	23
4.2 Örneklem Tabanlı Algoritmalar.....	25
4.3 Matematiksel Fonksiyon Tabanlı Algoritmalar	26
5. HAREKET KONTROL ALGORİTMALARI.....	29
5.1 Pure Pursuit Metodu	29
5.2 Stanley Metodu.....	30
6. OTONOM SOLLAMA MANEVRASI TASARIMI	33
6.1 Otonom Sollama Manevrası: Planlama ve Kontrol	33
6.1.1 X-sin fonksiyonu ile otonom sollama manevrasının planlaması.....	34
6.1.2 Stanley metodu ile otonom sollama manevrasının kontrolü.....	35
6.1.3 Boşluğu Takip Et metodu ile otonom sollama.....	36
7. SONUÇLAR	39
8. ÖNERİLER VE GELECEK ÇALIŞMA	49
KAYNAKLAR.....	51
EKLER	55
EK A.1	57
EK A.2.....	59



KISALTMALAR

ACC	: Adaptive Cruise Control
ADAS	: Advanced Driver Assistance Systems
BTE	: Boşluğu Takip Et Metodu
CC	: Cruise Control
DWA	: Dynamic Window Approach
MKK	: Model Kestrim Kontrolü
MOMDP	: Mixed Observable Markov Decision Process
PRM	: Probabilistic Road Map
RRT	: Rapidly-Random Tree
SAE	: Society of Automotive Engineers
XiS	: X-sin Fonksiyonu ile Stanley Metodu
YPA	: Yapay Potansiyel Alan Metodu



SEMBOLLER

α	: BTE'nin ayarlanabilir katsayısı
$\alpha(t)$: Pure Pursuit metodunda kullanılan hedef noktası ile arka tekerlek arasındaki çizgi ile aracın savrulma yönü arasındaki açı
a_{max}	: İzin verilen maksimum ivme
b_n	: BTE'deki n. boşluk mesafesi
C_f, C_r	: Tekerleklerin dönüş dirençleri
d_{ea}	: Aktör araç ile ego araç arasındaki mesafe
d_{min}	: En yakın engele olan uzaklık
d_n	: BTE'deki engellere olan n. mesafesi
δ	: Direksiyon açısı
δ_f, δ_r	: Ön ve arka tekerleklerin direksiyon açısı
δ_{PP}	: Pure Pursuit metodu ile bulunan istenen direksiyon açısı
e	: Yanal yol takip hatası
F_{yf}, F_{yr}	: Ön ve arka tekerleklere etkiyen yanal kuvvetler
I_z	: Atalet momenti
k	: Stanley metodunun ayarlanabilir katsayısı
L	: Araç aodelindeki araç uzunluğu
l_d	: Pure Pursuit metodunda 'İleri Bakma' mesafesi
L_f, L_r	: Aracın ağırlık merkezinden ön ve arka tekerleklere olan uzaklıklar
m	: Araç kütlesi
ϕ_n	: BTE'deki n. boşluk açısı
ψ	: Savrulma açısı
ψ_{ss}	: Ego aracın savrulma açısı ile yolun savrulma açıları arasındaki fark
$\dot{\psi}$: Savrulma hızı
$\ddot{\psi}$: Savrulma ivmesi
R	: Aracın dönüş yarıçapı
t_{cz}	: Çarpışma zamanı
v_{net}	: Aktör araç ile ego araç arasındaki hız farkı
$v_x, \dot{x}, v_y, \dot{y}$: Hız bileşenleri
x, y	: Pozisyon bileşenleri



ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 1.1 : Türkiye'deki trafik kazası istatistikleri [1].	5
Çizelge 1.2 : 2017 yılına ait ölümlü ve yaralanmalı trafik kazalarına sebep olan kusurların oranları [2].	6
Çizelge 2.1 : Dinamik Bisiklet Modeli parametreleri.....	14
Çizelge 7.1 : XiS'nin farklı a_{max} değerlerine göre karşılaştırma çizelgesi.....	41
Çizelge 7.2 : BTE ve XiS karşılaştırma çizelgesi.....	44
Çizelge 7.3 : 1000 adet Monte Carlo simülasyon sonuçları.	48



ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 1.1 : Otonom araç seviyeleri.....	2
Şekil 1.2 : Otonom araçta kullanılan sensörler ve fonksiyonları.....	4
Şekil 2.1 : Kinematik Araç Modeli.....	10
Şekil 2.2 : Dinamik Araç Modeli.....	13
Şekil 3.1 : Böcek-1 algoritmasının gösterimi.	16
Şekil 3.2 : Böcek-2 algoritmasının gösterimi.	16
Şekil 3.3 : Yapay Potansiyel Alan metodunun gösterimi [3].....	17
Şekil 3.4 : θ açısının gösterimi.	18
Şekil 3.5 : Boşluğu Takip Et metodunun genel gösterimi.	20
Şekil 3.6 : Boşluğu Takip Et metodunda açılarının gösterimi.	20
Şekil 4.1 : A* algoritmasının gösterimi.	24
Şekil 4.2 : X-sin fonksiyonu ile hareket planı.....	27
Şekil 4.3 : X-sin fonksiyonunun parametreler ile gösterimi.....	27
Şekil 5.1 : Pure Pursuit metodunun gösterimi.	30
Şekil 5.2 : Stanley metodunun gösterimi.	31
Şekil 6.1 : Klasik yaklaşım ile sollama.....	33
Şekil 6.2 : Önerilen yaklaşım ile sollama manevrası.....	34
Şekil 6.3 : X-sin fonksiyonu gösterimi.	35
Şekil 6.4 : Boşluğu Takip Et metodunun sollama algoritması.....	37
Şekil 6.5 : Şerit değiştirme fazı.....	38
Şekil 6.6 : Birleşme fazı.....	38
Şekil 6.7 : Geçme fazı.....	38
Şekil 7.1 : $a_{max} = 2.5m/s^2$ iken XiS; a. $v_{ego} = 20m/s$, b. $v_{ego} = 25m/s$, c. $v_{ego} = 30m/s$	42
Şekil 7.2 : $a_{max} = 3.5m/s^2$ iken XiS; a. $v_{ego} = 20m/s$, b. $v_{ego} = 25m/s$, c. $v_{ego} = 30m/s$	43
Şekil 7.3 : BTE'nin karşılaştırma sonuçları; a. $v_{ego} = 20m/s$, b. $v_{ego} = 25m/s$, c. $v_{ego} = 30m/s$	46
Şekil 7.4 : XiS'nin karşılaştırma sonuçları; a. $v_{ego} = 20m/s$, b. $v_{ego} = 25m/s$, c. $v_{ego} = 30m/s$	47



BOŞLUĞU TAKİP ET METODU İLE OTONOM SOLLAMA MANEVRASI TASARIMI

ÖZET

Otomotiv endüstrisi, hızla gelişen ve her gün yeni akımların ortaya çıktığı bir sektördür. Bu yeni akımlar tam otonom sürüşü ön plana çıkaracak gelişmelerdir. Aslında otonom sürüş ilk kez 1920'lerde düşünülmüştür. 1939 yılında da General Motors şirketi New York Dünya Fuarı'nda sürücüsüz tasarım fikrini ilk kez sunmuştur. Daha sonra 1958'de ilk prototiplerini üretmiş, kendi kendine sürüş yapabilecek bir araç geliştirmişlerdir. Bu araç günlük hayatta kullanıma uygun olmamasına rağmen otonom aracın ilk adımı olarak kabul edilir. 20 yıl sonra, japon bilim adamları iki adet kamera ile aracı kontrol ederek otonom araç için önemli bir adım atmışlardır. 2000'li yıllara geldiğimizde ise otonom araçlar arasında yarış başladı. İlk olarak 2004 yılında DARPA, 1 milyon dolar ödüllü bir yarış başlattı. 10 saatte 142 millik rotayı gidebilecek olan bir otonom araç arıyordu. Yarışa 15 araç katıldı, ancak en iyi araç sadece 7 mil yol alabildi. Bir sonraki yıl ise buna benzer bir yarışma yapıldı. Bu kez geçilmesi gereken 100 tünel, 100'den fazla dönüş ve sarp kayalıklar ile 132 millik bir rota söz konusuydu. Ödül ise 2 milyon dolara çıkarıldı. Stanford Üniversitesinde bir ekip tarafından geliştirilen sürücüsüz bir Volkswagen, 7 saatin altında bu zorlu parkuru tamamlamayı başardı. Bugün birçok araçta tam otonom olmasa da yarı otonom olarak kabul edilebilecek bir sürü fonksiyon bulunmaktadır. Park sistemleri, şerit takip sistemleri, acil durum frenleme sistemleri gibi fonksiyonlar günümüzde yollarda gördüğümüz birçok araçta bulunmaktadır. Özellikle acil durum frenleme sistemleri, hayat kurtarıcı rol oynayabilmektedir.

Her yıl dünyada yaklaşık 1.2 milyon insan trafik kazası nedeniyle hayatını kaybetmektedir. 50 milyon civarı insan ya trafik kazasından dolayı yaralanmaktadır ya da trafik kazası bu insanlarda kalıcı hasar bırakmaktadır. Trafik kazasında zarar gören insanların çoğu yaya, motosiklet kullanıcı veya bisiklet kullanıcıdır. Bu da gösteriyor ki, otomobil sürücülerinin hataları, diğer insanlara zarar vermektedir. Bu kazalar sürücülerin dikkatsizliklerinden kaynaklanmaktadır [4].

Otonom sürüş için geliştirilen fonksiyonlarda, vurgulanması gereken çok önemli iki nokta vardır. Birincisi, sürücü dikkatsizliğine nedeniyle insan hayatını tehdit edecek durumlar azaltılmalıdır. ikincisi ise otonom sürüş algoritması insan hayatını riske sokacak herhangi bir karar vermemelidir. Bu tezde, otonom sürüş için çok önemli olan sollama manevrası problemi için otonom sollama manevra algoritması tasarlanmıştır. Bu şekilde, sollama manevrası sırasında herhangi bir kaza olmadan, kolay ve güvenli bir şekilde sollama manevrası sağlamak amaçlanmaktadır. Bu otonom sollama manevra algoritması, sollama manevrasını başarıyla gerçekleştirmeye odaklanır, sollama manevrası sırasında da bir yaya ve sürücüye zarar vermemesi istenir. Sıradan bir sürücü yolu takip etmek için sadece direksiyon, gaz ve freni kontrol eder ve aracın hızını düşünmeden ayarlar. Yine de, sollama manevrası problemi, literatürden de görüleceği gibi, yıllardır araştırılmaktadır. Çünkü otonom sürüş fonksiyonu derinlemesine analiz edilirse, birçok disiplinin ve konuların iş birliğinin gerektiği

görülmektedir. Örnekleyecek olursak, bu disiplin ve konular; sensör teknolojileri, konum bulma ve haritalama teknolojileri, görüntü işleme algoritmaları, karar verme ve yol oluşturma algoritmaları, araç kontrol teorileri ve otomotiv mühendisliği gibi sıralanabilir.

Otonom sollama problemi, üç ayrı fazla incelenmektedir. Bu fazlar, şerit değiştirme fazı, geçme fazı ve birleşme fazı olarak adlandırılırlar. Şerit değiştirme fazı aktör araç bile ego araç arasında belli bir hız farkı olduğunda, geçilecek şerit güvenli ise gerçekleştirilecek bir eylemdir. Şerit değiştirme fazında sürücü, aktör araç ile arasındaki mesafeyi korumalıdır. Aynı zamanda, şerit değiştireceği şeritteki güvenlikten emin olmalıdır. Otonom sollama manevrasının başlaması için, belli koşulların gerçekleşmiş olması gerekmektedir. Bu koşullardan bir tanesi geçilecek şeridin güvenliğidir. Bu tezdeki otonom sollama manevrasında geçilecek şerit her zaman güvenli kabul edildi. Öteki koşul ise, ego araç aktör aracın arkasında olmalı ve hızının daha yüksek olması gerekmektedir. Bu tezde, bu hız farkına bağlı olarak azalıp belli bir süre sonra çarpışma olacağı anlaşıldığında otonom şerit değiştirme manevrası başlamaktadır. Şerit değiştirme fazı, geçilecek şeridin orta noktasına gelindiğinde tamamlanmaktadır.

İkinci faz geçme fazıdır. Bu fazda ego araç sollama şeridine geçmiştir ve aktör araçla boylamsal düzlemde birbirlerine çok yakındırlar. Bu fazdaki aksiyon, ego aracın aktör aracı geçene kadar şerit takibi yapmasıdır. Bu faz, ego araç aktör aracın belli bir mesafe önüne geçene kadar devam eder. Ego araç ile aktör araç yatak düzlemde farklı şeritlerdir. Boylamsal düzlemde ise ego araç aktör aracın belli bir mesafe önüne geçtiğinde geçme fazı tamamlanır.

Otonom sollama manevrasındaki son faz birleşme fazıdır. Bu fazda ego araç sollama şeridinden çıkıp aktör aracın önüne gelecek şekilde eski şeridine dönecektir. Sürücünün sollamayı tamamlarken, eski şeridindeki aracın iyi gözlemlemesi gerekmektedir. Önüne geçeceği aktör araç ile çarpışmaması için mesafeyi iyi ayarlamalıdır. Otonom sollama manevrasında, geçme fazı tamamlandığında ego araç aktör aracın belli mesafe önündedir. Eğer hala bu mesafeyi korumuş ise, birleşme fazı başlar. Birleşme fazı, ego araç ile aktör araç aynı şeritte olup, ego araç aktör aracın belli bir mesafe önüne geldiğinde sonlanır.

Bu tezde bu tasarım, iki ayrı yaklaşımla gerçekleştirilmektedir. Bu yaklaşımlardan bir tanesi çarpışma önleme fonksiyonun yola odaklanması ve engelden kaçması üzerine, doğal olarak sollama manevrasının ortaya çıkmasıdır. Engelden kaçma algoritması otonom sollama manevrasını şerit değiştirme fazına uygulanırken, aracın sol arka köşesinin bir engel, ona şerit çizgisinde karşılık gelen nokta ikinci engel olarak kabul edilir. Sollama şeridinin tam ortası ise hedef noktası olarak kabul edilir. Buna göre engelden kaçma algoritması şerit değiştirme fazı başarılı bir şekilde tamamlamış olur. Geçme fazında şerit takibi yapması gerektiğinden, aktör aracın sol ön köşesinden belli bir mesafe ötesi bir engel ona karşılık gelen sol şerit çizgisinde öteki engel olarak kabul edilir. Hedef noktası sollama şeridinin orta noktası kabul edilip, ego araçtan belli bir mesafe ötede kabul edilerek uygun direksiyon açısı bulunur. Geçme fazını da tanımlanmış engellerden kaçarak tamamlamış olur. Birleşme fazında engel olarak, aktör aracın sol ön köşesinin belli bir mesafe ötesi birinci engel, o noktanın yatak eksen pozisyonu ile, sol şeridin o yata eksen pozisyonundaki noktası bulunarak ikinci engel olarak kabul edilir. Hedef noktası aktör aracın belli bir mesafe ötesi kabul edildi. Aktör araç zaten eski şeridin orta noktasından gittiği varsayıldığından,

otonom sollama manevrasının birleşme fazı eski şeridin orta noktasında ve aktör aracın belli bir mesafe ötesinde tamamlanmış olur. Diğer yaklaşım ise, X-sin fonksiyonu ile sollama manevrasının planlaması yapılmaktadır. Planlanan manevra Stanley metodu ile kontrol edilmektedir. Sollamaya karar verildiğinde, şerit değiştirme fazı başlar. Sollama şeridinin orta noktasına gelecek şekilde x-sin fonksiyonu yol planlanır. Bu yol stanley kontrolcü ile takip edilir. Böylece sollama şeridine geçilmiş olur. Geçme fazında, aktör aracın belli bir mesafe önüne gelene kadar X-sin fonksiyonu tarafından şerit takip yapılacak yol planlanır ve Stanley kontrolcüsü ile yolun takibi yapılır. Geçme fazı bittikten sonra, birleşme fazı başlar. Aktör aracın belli bir mesafe önüne geçilmişse artık eski şeridin orta noktasına doğru aktör aracın belli bir mesafe ötesi hedef noktası olacak şekilde X-sin fonksiyonu tarafından yol planlanır. Stanley kontrolcüsü ile bu yol takip edildiğinde otonom sollama manevrası ise klasik yaklaşım için tamamlanmış olur.

Her iki yaklaşım Matlab/Simulink yazılımı ile otoban koşulları için gerçekleştirilmiştir ve Bölüm 6'da detaylı olarak açıklanmaktadır.





AUTONOMOUS OVERTAKING MANEUVER DESIGN BASED ON FOLLOW THE GAP METHOD

SUMMARY

The automotive industry is a sector that is rapidly developing and new trends are emerging every day. These new trends are developments that will bring full autonomous driving to the fore. In fact, autonomous driving was first thought in the 1920s. In 1939, General Motors presented the idea of driver-less design for the first time at the New York World's Fair. Then, in 1958, they produced their first prototypes and developed a vehicle that can perform self-driving. Although this car is not suitable for daily life, it is considered as the first step of the autonomous vehicle. 20 years later, Japanese scientists took an important step for the autonomous vehicle by controlling the vehicle with two cameras. By the 2000s, competition between autonomous vehicles was started. First in 2004, DARPA launched a \$1 million contest. He was looking for an autonomous vehicle that could travel the 142-mile route in 10 hours. 15 vehicles participated in the race, but the best vehicle was only 7 miles. The next year, a similar contest was held. This time there were 100 tunnels, more than 100 turns and steep cliffs and 132 miles. The prize was raised to \$2 million. A driver-less Volkswagen, developed by a team at Stanford University, managed to complete this challenging track in under 7 hours. Although there is no commercial fully autonomous level-5 autonomous vehicles now, there are many functions that can be considered semi-autonomous. Parking systems, lane tracking systems, emergency braking systems are nowadays in many vehicles we see on the road. In particular, emergency braking systems can play a life-saving role.

Approximately 1.2 million people die every year in the world due to traffic accidents. Around 50 million people are either injured due to traffic accidents or traffic accidents cause permanent damage to these people. Most people who are injured in a traffic accident are pedestrians, motorcyclists or cyclists. This shows that the faulty driving of automobile drivers are damaging to other people. These accidents are caused by careless driving [4].

For functions developed for autonomous driving, there are two very important points to emphasize. First, it should reduce the situation that threatens human life due to driver carelessness, and second, the autonomous driving algorithm should not make any decisions that would put human life at risk. In this thesis, autonomous overtaking motion algorithm is designed for the problem of overtaking maneuver which is very important for autonomous driving. In this way, it is intended to provide an easy and safe overtaking maneuver without any accidents during the overtaking maneuver. This autonomous overtaking maneuver algorithm focuses on successful overtaking maneuver, also during the overtaking maneuver it is desirable not to damage a pedestrian and driver. Automated driving requires deep understanding and cooperation of many different disciplines and topics, such as sensor technologies, localization and mapping technologies, estimation and fusion algorithms, image processing algorithms, decision making and trajectory generation algorithms, vehicle controls theory and

automotive engineering An ordinary driver just steers the steering wheel and apply brake or gas pedal to follow the lane and adjust the speed of the vehicle even without thinking. Nevertheless, this overtaking maneuver problem is under research for years as can be observed from the literature.

The autonomous overtaking problem is studied in three separate phases. These phases are called the lane change phase, the transition phase and the junction phase. The lane change phase is an action to be taken if the lane to be passed is safe, even when the actor vehicle has a certain speed difference between the ego vehicle. In the lane change phase, the driver must maintain the distance from the actor to the vehicle. At the same time, the lane should be sure of the security in the lane it will change. Certain conditions must be met for the autonomous overtaking maneuver to begin. One of these conditions is the safety of the lane to be crossed. In the autonomous overtaking maneuver in this thesis, the lane to be crossed was always considered safe. The other condition is that the ego vehicle must be behind the actor vehicle and the speed should be higher. In this thesis, the autonomous lane change maneuver starts when it is understood that this speed difference will decrease and collision will occur after a certain period of time.

The second phase is the passing phase. This excess ego vehicle has passed to the overtaking lane and the actor and the vehicle are very close to each other in the longitudinal plane. The action in this phase is that the ego vehicle tracks the lane until the actor vehicle passes. This phase continues until the ego vehicle passes a certain distance from the actor vehicle. Ego vehicle with actor car are in different lane in bed plane. In the longitudinal plane, the engagement phase is completed when the ego vehicle passes the actor vehicle a certain distance.

The last phase in the autonomous overtaking maneuver is the coupling phase. In this phase, the ego vehicle will leave the overtaking lane and the actor will return to the old lane in front of the vehicle. When completing the overtaking of the driver, the vehicle on the old lane should observe well. The actor in front of the vehicle must be well-adjusted to avoid collision with the vehicle. In the autonomous overtaking maneuver, the ego vehicle is at a certain distance from the actor vehicle when the engagement phase is completed. If it still maintains this distance, the coupling phase begins. The merging phase ends when the ego vehicle and the actor vehicle are in the same lane, when the ego vehicle is within a certain distance of the actor vehicle.

In this thesis, this design is realized with two different approaches. One of these approaches is that the overtaking maneuver naturally occurs when the collision prevention function focuses on the road and avoids the obstacle. While the obstacle avoidance algorithm applies the autonomous overtaking maneuver to the lane change phase, an obstacle of the vehicle's left rear corner is considered to be the second obstacle to the corresponding point in the lane line. The center of the overtaking lane is considered the target point. Accordingly, the obstacle avoidance algorithm completes the lane change phase successfully. Since the lane must be tracked during the engagement phase, the actor is considered to be an obstacle beyond a distance from the left front corner of the vehicle and the corresponding left lane line. The target point is assumed to be the midpoint of the overtaking lane and the ego is located a distance away from the vehicle, providing the appropriate steering angle. It completes the passing phase by avoiding the defined obstacles. As an obstacle in the coupling phase, the first obstacle beyond a certain distance of the left front corner of the actor

vehicle is considered to be the second obstacle with the bearing axis position of that point and the point of the left lane at the axis position to that yacht. The target point actor was considered a certain distance from the vehicle. Since the actor vehicle is presumed to have already gone from the midpoint of the old lane, the junction phase of the autonomous overtaking maneuver is completed at the midpoint of the old lane and beyond a certain distance of the actor vehicle. The other approach is the planning of overtaking maneuver with X-sin function. The planned maneuver is controlled by the Stanley method. When overtaking is decided, the lane change phase starts. The x-sin function plans the path to the midpoint of the overtaking lane. This path is followed by stanley controller. Thus, the overtaking lane was passed. In the pass phase, the lane is planned by the X-sin function until the actor comes to a certain distance of the vehicle and the track is followed with the Stanley controller. After the engagement phase is finished, the coupling phase starts. If a certain distance of the actor vehicle is prevented, the X-sin function will then plan the path towards the midpoint of the old lane so that the actor vehicle is a certain cross-distance target point. Following this path with the Stanley controller, the autonomous overtaking maneuver is completed for the classical approach.

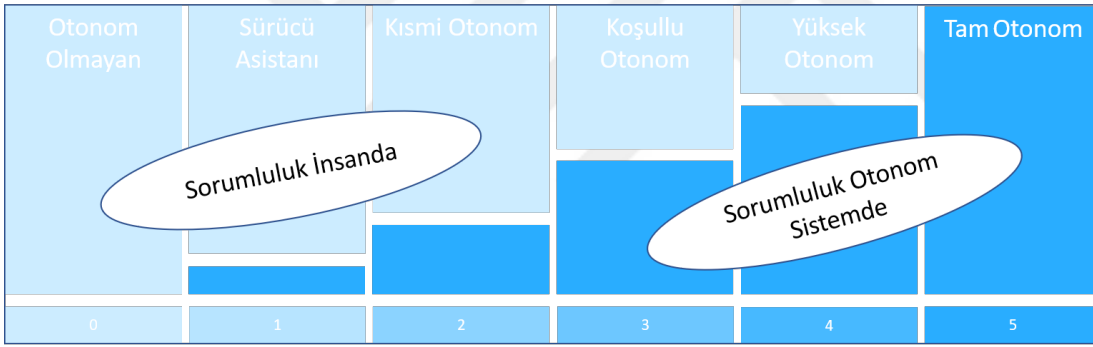
Both approaches have been implemented for highway conditions in Matlab / Simulink software. The details are explained in section 6.



1. GİRİŞ

Engelden kaçma sistemleri; mesafe ölçme, haritalama, konum gibi verileri elde edecek sensörleri kullanarak, olası bir çarpışma durumunda çarpışma önleme veya hasarı azaltma amacı ile otomotiv sektöründe adı geçen sistemlerdir. Bu sistemlere her geçen gün yeni yeni fonksiyonlar eklenerek bu sistemlerin otomotiv sektöründe gelişimleri devam etmektedir. Çarpışma önleme sistemleri ilk olarak 1950'lerin sonunda bahsedilmeye başlanmıştır. İlk olarak Cadillac üretilmiş, bu prototipte radar kullanılarak, aracın önündeki objeler tespit edilmiştir O zaman için çok maliyetli olduğuna karar verilip, üretim yapılmamıştır. [5]'a göre önden çarpışma önleme sistemleri 1955 yılında Kaliforniya'da bir araştırma laboratuvarında tanıtılmıştır. Bu sistem radar tabanlı olup herhangi bir yerde kullanılmamıştır. Bunun yanı sıra ilk uyarlanabilir seyir kontrolü sistemini Toyota 1977'de Celsior modelinde tanıttığından ve sadece Japonya'da satışa sunduğundan bahseder. Günümüzde, fren destek sistemleri, manevra kontrol sistemleri, uyarlanabilir seyir kontrolü sistemleri, kör nokta destek sistemleri, şerit takip sistemleri, ve şerit değiştirme destek sistemleri gibi farklı alanlara odaklanan çarpışma önleyici sistemler, araçlarda kullanılmaya başlanmıştır. Ocak 2014'te SAE International mühendisleri tarafından belirlenen standartlar, otonom otomobil akımının altı seviyeli bir gelişim devinimi olduğunu ortaya koyuyor [6]. Şekil 1.1'de detaylı olarak gösterilmektedir. Seviye 0'da bulunan araçlarda herhangi bir otonom fonksiyon mevcut değildir. Seviye 1'de sürücüye yardımcı fonksiyonlar bulunmaktadır. Bu fonksiyonlar sürücü asistanı olarak geçer ve çeşitli durumlarda sürücünün direksiyon yönlendirmesine veya hızı değiştirme yönünde sürücüye destek verebilir. Bu seviyedeki tüm sorumluluk sürücüye aittir. Bu seviye için, otomobillerdeki hızı korumak için kullanılan Seyir Kontrolü (CC) fonksiyonu örnek verilebilir. Seviye 2, diğer bir adlandırma ile 'Kısmi Otonom' seviyesi, aracın çeşitli fonksiyonları kendisinin yapabilmesidir. Bu fonksiyonların izni sürücü tarafından verilir. Bu durumda otonom fonksiyona sahip araç, hızı ve direksiyonu kontrol edebilir. Bu kararı sürücü verdiği için, tüm sorumluluk sürücüye

aittir. Adaptif Seyir Kontrolü (ACC) fonksiyonu lider araç ile ego araç arasındaki mesafeyi koruyacak şekilde aracın hızını kontrol eder. İki araç arasındaki mesafe, sensörler tarafından ölçülür bu ölçüye göre bir hız kontrolü yapılmaktadır. Hız kontrolü ile birlikte kullanılacak şerit takip fonksiyonu 2. seviye bir sürüş örneğidir. Bu seviyede sürücü direksiyon hakimiyetine ve yol hakimiyetine sahip olmalıdır. Seviye 3, 'Koşullu Otonom' seviyesidir. Bu seviyede, araç çevreyi algılar, hızı ve direksiyonu kontrol eder. Sürücü bu seviyede kontrolü ele almaya her an hazır olmalıdır. 'Yüksek Otonom' seviyesi 4. seviyeyi temsil eder. Bu seviyede artık sürücü, sürüşe müdahil olmaz. Yalnız acil durumlarda veya aracın algıyalamadığı durumlarda yüksek otonom sistem, sürücüye kontrolü ele almasını söyler bu uyarıda sürücü kontrolü ele almasa dahi kendisi başa çıkabilir. Henüz bu seviyede bir aracın üretimi yoktur ve yollarda olması da yasal değildir. Seviye 5, tam otonom seviyedir. Bu seviyede sürücü yoktur. Araç bir insanın yapabileceği her şeyi yapar. Prototipleri denenmektedir fakat yasallaşmamıştır. Otonom fonksiyonlar için gerekli

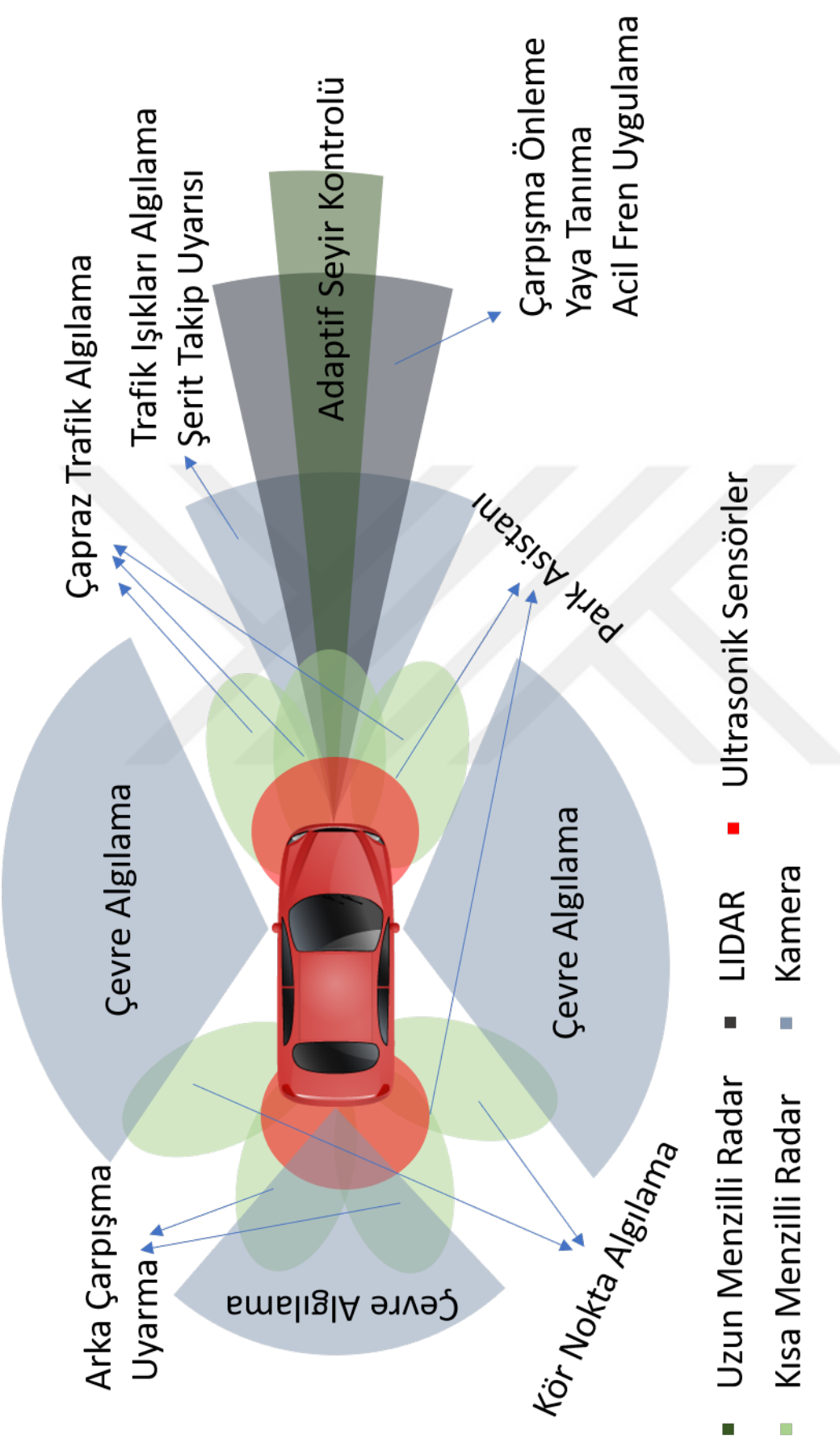


Şekil 1.1 : Otonom araç seviyeleri.

donanımlar [7]'de bahsedilmektedir. Burada önemli olan nokta aracın çevresini en doğru şekilde algılayabilmesidir. Ultrasonik sensörlerin alıcısı ve vericisi vardır. Vericisinden ses dalgası gönderip alıcısından sesi alırlar aradan geçen süreye göre de engelin mesafesini ölçerler. Ultrasonik sensörler genelde daha kısa mesafelerde önemli olan, otonom park fonksiyonlarında kullanılmaktadır. Radar ve LIDAR'ın algılama mesafesi ultrasonik sensörlerden çok daha fazladır. Radarda da alıcı ve verici vardır yalnız gönderdiği dalga radyo dalgasıdır. Radarlar kısa menzilli ve uzun menzilli olmak üzere farklı amaçlar için üretilmektedir. Kısa menzilli radarlar aracın ön tarafına yerleştirildiğinde, kısa mesafedeki tıkanmış trafik uyarısı için; aracın arkasına yerleştirildiğinde ise arka kaza çarpışma uyarısı ve yanlara yerleştirilen radarlar kör nokta algılama fonksiyonlarında kullanılmaktadır. Uzun menzilli radarlar

ise ACC fonksiyonlarında sıkça tercih edilmektedir. LIDAR ise ışık gönderir ve alır. LIDAR radardan farklı olarak, döner bir yapıya sahip olabilir ve bu özelliğe sahip olan LIDAR çevrenin 360° dilimini alır, bu sayede tüm çevresini algılamış olur. Genelde, acil durum freni, yaya algılama ve engelden kaçma fonksiyonları için kullanılmaktadır. Kamera otonom aracın çevresini tanımada kullanılır. Kamera ile genelde şerit çizgileri algılanır ve kamera şerit takibini sağlar. Trafik ışıkları kamera ile algılanır. Arka park sensörü olarak kamera, sürücüye asistanlık yapmaktadır. Kameranın yağmurlu, sisli hava koşullarında veya yoğun güneş ışığı nedeniyle parlayan havalarda kullanımı durumunda doğru görüntü alınamamaktadır. Şekil 1.2, hangi sensörlerin hangi fonksiyonlarda kullanabildiğinin bir gösterimidir.





Şekil 1.2 : Otonom araçta kullanılan sensörler ve fonksiyonları.

Çizelge 1.1 : Türkiye’deki trafik kazası istatistikleri [1].

YIL	KAZA SAYISI	KAZA YERİ ÖLÜ SAYISI	KAZA SONRASI ÖLÜ SONRASI(1)	TOPLAM ÖLÜ SAYISI	YARALI SAYISI
2008	950.120	4.236	-	-	184.468
2009	1.053.346	4.324	-	-	201.380
2010	1.104.388	4.045	-	-	211.496
2011	1.229.928	3.835	-	-	238.074
2012	1.296.634	3.750	-	-	268.079
2013	1.207.354	3.685	-	-	274.829
2014	1.199.010	3.524	-	-	285.059
2015	1.313.359	3.831	3.699	7.530	304.421
2016	1.182.491	3.493	3.807	7.300	303.812
2017	1.202.716	3.534	3.893	7.427	300.383

(1) Trafik kazasında yaralanıp sağlık kuruluşuna sevk edilen kazanın sebep ve tesiriye 30 gün içinde ölenleri kapsamaktadır.
- Bilgi yoktur

Dünyada her yıl 1.2 milyondan fazla insan trafik kazalarında vefat ediyor. 50 milyon kişi yaralanıyor veya bu kaza yaralıları kalıcı bir hasar bırakıyor [4]. Çizelge 1.1’de görülen veriler Türkiye’deki trafik kaza istatistiklerini gösteriyor. Örneğin, 2016 yılında 7300 hayatını kaybetmiş ve 308.812 kişi yaralanmıştır. 2015 verilerine bakıldığında ise 7530 kişi hayatını kaybetmiş bu insanların 3831 tanesi olay yerinde vefat etmiştir. Aynı sene 304.421 kişi trafik kazasından dolayı yaralanmıştır [8]. Trafik kazalarının sebepleri incelendiğinde KGM, Çizelge 1.2’deki verileri sunmaktadır. Bu veriler yerleşim yeri ve yerleşim dışında bulunan otoyol, bağlantı yolu, il yolu ve devlet yolunda meydana gelen kazaları kapsamaktadır. Burada açıkça görülüyor ki, sürücülerden dolayı oluşan kazaların oranı %95.43’tür. Sürüş asistanlarının geliştirilmesinin en önemli amacı bu oranı düşürmektir.

Trafik kazalarının %90’ı, Dünya Sağlık Örgütü’nün verilerine göre arkadan çarpmalı kazalardır ve eğer 1.5 sn önce farkedilirse bu oran önemli ölçüde azaltılabilir. Engelden kaçma sistemlerinin, sürücünün 1.5 sn’lik eksikliğini gidererek kazaların sayısını önemli ölçüde düşüreceği öngörülmektedir [4]. Hareket planlama algoritmaları robotun yapması istenilen hareketleri ayırık hareketler olarak ifade eden fonksiyonlardır. Hareket planlama, 1991 yılında J.-C. Latombe bir robot gerçek dünyada hareket ederek görevlerini gerçekleştirebilir demesiyle ortaya atılmış oldu [9]. Otomotiv, yapay zeka, otomasyon gibi bir çok alanda geliştirilip, kullanılabilirler. Otonom araba seviyelerinde sürücünün sürüşe dahil olmaması adına hareket planlama algoritmaları son derece önemlidir. Sürücü olmayan aracın hareketi için de makul bir yol çizmek amaçlanır. Bu yol aynı zamanda yolcuların en başta güvenliğini daha sonra da konforunu düşünerek planlanmalıdır. Bütün bu amaçlara hizmet edecek ‘İleri Yardım Sürüş Sistemleri’

Çizelge 1.2 : 2017 yılına ait ölümlü ve yaralanmalı trafik kazalarına sebep olan kusurların oranları [2].

Kaza Faktörleri	Kusur Sayısı	%
Sürücü	59.973	95,43
Yaya	1.774	2,82
Taşıt	555	0,88
Yol	277	0,44
Yolcu	265	0,42

(ADAS) her gün biraz daha gelişmektedir. Günümüzde otonom sollama manevrasının planlanması için en iyi denilecek tek bir yöntem yoktur. Geliştirilen yöntemlerin avantajları ve dezavantajları vardır. Hareket planlama algoritmaları kullanılacak yere göre global veya yerel planlama olabilir. Global planlama yapabilmek için aracın anlık pozisyonuna ve bütün çevre bilgisine ihtiyacı vardır veya daha gelişmiş sistemlerde bir karar verme mekanizması vardır buradan gelen komutları dikkate alarak bir hareket planlaması gerekmektedir. Yerel planlamada tüm çevre bilgisine gerek olmamakla birlikte sadece bir sonraki adım için çevre bilgisi yeterlidir. Hareket kontrol algoritmaları, robotun istenilen yere gidebilmesini sağlayan fonksiyonlardır. Bu fonksiyonlar geometrik ve modelleme tabanlı kontrol algoritmaları olarak iki ana gruba ayrılır. En popüler geometrik algoritmaların başında Pure Pursuit ve Stanley metodu gelir [10]. Geometrik tabanlı algoritmalar, bozuk girdilere daha gürbüz tepki verirler. Yüksek hızlarda performansları düşebilir. Yüksek hızlardaki performanslar incelendiğinde Stanley metodu, Pure Pursuit'e göre daha konforludur [11]. Pure pursuit ayrıca, manevralarda Stanley'e göre kararsızdır. Model tabanlı metotlar, aracın matematiksel modelini kullanırlar ve aracın matematiksel modeline göre bir kontrol stratejisi oluştururlar. Aracın matematiksel modeli, kinematik, doğrusal dinamik ve doğrusal olmayan dinamik model olabilir. Hareket kontrol algoritmalarında bulanık mantık da kullanılabilir. [11]'e göre yol takip hatası, PI kontrolörü tarafından, savrulma açısı hatası da bulanık mantık kontrolör tarafından giderilmeyi önermektedir. Model Kestirim Kontrolü (Model Predictive Control - MPC) de hareket kontrol algoritmaları için sıkça kullanılmaktadır. Bölüm 2'de kinematik ve dinamik bisiklet araç modelleri tanıtılmaktadır ve bu modellerin hangi amaçlarda nasıl kullanıldığı açıklanmaktadır. Bu tezde dinamik bisiklet araç modelinin neden seçildiğini ve nasıl kullanıldığı da anlatılmaktadır. Bölüm 3'te engelden kaçma algoritmalarının ne olduğu, nasıl çalıştıkları ve hangi alanlarda kullanıldıkları açıklanmıştır. Özellikle Boşluğu Takip

Et metodunun çalışma prensibi ve engelden kaçma algoritmasının yanı sıra bir otonom sollama manevrasına da uygunluğu vurgulanmaktadır. Bölüm 4'te hareket planlama algoritmalarının ne olduğu nasıl kullanıldığı ve literatürde bulunan bazı hareket planlama algoritmalarının nasıl çalıştığı anlatılmaktadır. Bölüm 5'te hareket kontrol algoritmalarının açıklaması yapılmaktadır. Pure Pursuit ve Stanley metodunun nasıl çalıştığı açıklanmaktadır. Bölüm 6'da otonom sollama manevrasındaki problemlerden bahsedilmektedir ve bu tezde otonom sollama manevrasının problemleri için nasıl bir çözüm tasarlandığı anlatılmaktadır. Bölüm 7'de bu tezin gerçekleştirildiği yazılım ve bu yazılımın hangi koşullar için uygulandığı anlatılmaktadır. Yaklaşımların sonuçları ve elde edilen çıktılar ayrıntılarıyla birlikte sunulmaktadır. Son olarak gelecekte yapılabilecek çalışmalar bölüm 8'da anlatılmaktadır.



2. ARAÇ MODELİ

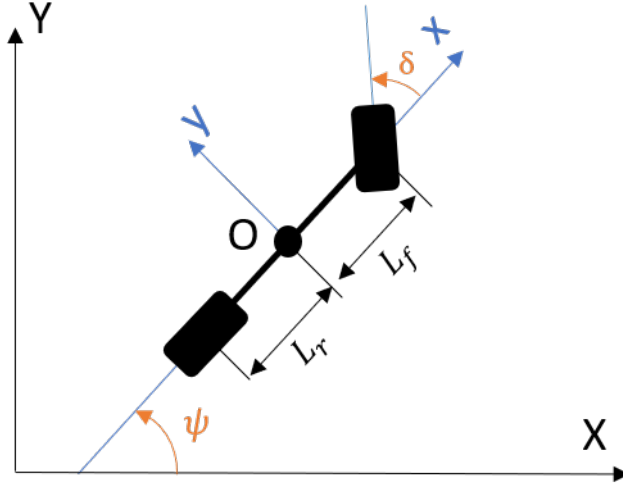
Bu bölümde hangi araç modellerinin kullanıldığını ve bu araç modellerin hangi uygulamalarda kullanıldığını anlatılacaktır. Araç modeli türleri anlatılıp, avantajlar ve dezavantajlar karşılaştırıp hangi araç modelinin neden seçildiği açıklanacaktır. Bir araçta yönü direksiyon tayin eder, araç direksiyonun hareketine göre yol alır. Bu tez çalışmasında otonom sollama manevrası gerçekleştirileceğinden, aracın direksiyon açısı kontrol edilmektedir. Bu sebeple, aracın boylamsal hızı sabit kabul edilmiştir. Aracın direksiyon açısını kontrol edebilmek için yaygın olarak, kinematik ve dinamik araç modelleri kullanılır.

2.1 Kinematik Araç Modeli

Kinematik araç modeli, aracın hareketini tamamen geometri tabanlı matematiksel ifadelerle tanımlar [12]. Bu modelde harekete etkiyen kuvvetler hesaplara katılmaz. Şekil 2.1, bisiklet araç modelini gösterir, ön tekerlekler bir tekerlek ve arka tekerlekler tek teker olarak düşünülmüştür. Arka tekerleğin seyir açısını δ_r ve ön tekerleğin seyir açısını δ_f temsil eder. Önden çekişli araçlar için δ_r sıfır olarak kabul edilebilir. Aracın ağırlık merkezi O ile gösterilmiştir. Ön teker ile ağırlık merkezi arasındaki uzaklık l_f , arka teker ile ağırlık merkezi arasındaki uzaklık l_r ile gösterilmiştir. Aracın uzunluğu L , l_f ile l_r 'nin toplanmasıyla bulunur. Aracın global pozisyonunu temsil etmek için, X ve Y kullanılmış, aracın lokal koordinat düzlemi temsili için de x ve y kullanılmıştır. Aracın savrulma açısı ψ ve ağırlık merkezindeki hızı v ile gösterilmiştir. Kinematik modelde genelde kayma olmadığını kabul edilir ve önden çekişli olarak hareket ettiği varsayılır. Buna istinaden ön ve arka tekerleklerdeki hareket denklemi (2.2) ve (2.1)'deki gibi ifade edilir. Bu denklemlerden ön tekerin x ve y deki pozisyonları (2.3) ve (2.4) denklemleri ile bulunur.

$$\dot{x}_f \sin(\delta + \psi) - \dot{y}_f \cos(\delta + \psi) = 0 \quad (2.1)$$

$$\dot{x}_r \sin(\psi) - \dot{y}_r \cos(\psi) = 0 \quad (2.2)$$



Şekil 2.1 : Kinematik Araç Modeli.

$$x_f = x + L \cos(\psi) \quad (2.3)$$

$$y_f = x + L \sin(\psi) \quad (2.4)$$

Denklemler (2.3) ve (2.4)'ü, (2.1) denkleminin içerisinde kullanılıp x_f ve y_f 'ler elendiğinde,

$$\frac{\delta(x_r + L \cos(\psi))}{\delta t} \sin(\delta + \psi) - \frac{\delta(y_r + L \sin(\psi))}{\delta t} \cos(\delta + \psi) = 0 \quad (2.5)$$

$$(\dot{x}_r - \dot{\psi} L \sin(\psi)) \sin(\delta + \psi) - (\dot{y}_r + \dot{\psi} L \cos(\psi)) \cos(\delta + \psi) = 0 \quad (2.6)$$

$$\dot{x}_r \sin(\delta + \psi) - \dot{y}_r \cos(\delta + \psi) - \dot{\psi} L \sin(\psi) \sin(\psi + \delta) - \dot{\psi} L \cos(\delta + \psi) = 0 \quad (2.7)$$

(2.5), (2.6) ve (2.7) denklemlerindeki trigonometrik ifadelerde açılım yapıldığında (2.8), (2.9), (2.10), (2.11) ve (2.12) denklemleri elde edilir.

$$\dot{x} \sin(\delta + \psi) - \dot{y} \cos(\delta + \psi) - \dot{\psi} L \sin(\psi) (\sin(\psi) \cos(\delta) + \sin(\delta) \cos(\psi)) - \dot{\psi} L \cos(\psi) (\cos(\psi) \cos(\delta) - \sin(\delta) \sin(\psi)) = 0 \quad (2.8)$$

$$\dot{x} \sin(\delta + \psi) - \dot{y} \cos(\delta + \psi) - \dot{\psi} L (\sin^2(\psi) \cos(\delta) + \sin(\psi) \sin(\delta) \cos(\psi)) - \dot{\psi} L (\cos^2(\psi) \cos(\delta) - \cos(\psi) \sin(\delta) \sin(\psi)) = 0 \quad (2.9)$$

$$\dot{x} \sin(\delta + \psi) - \dot{y} \cos(\delta + \psi) - \dot{\psi} L (\sin^2(\psi) \cos(\delta) - \dot{\psi} L (\cos^2(\psi) \cos(\delta) - \dot{\psi} L \sin(\psi) \sin(\delta) \cos(\psi)) + \dot{\psi} \cos(\psi) \sin(\delta) \sin(\psi)) = 0 \quad (2.10)$$

$$\dot{x} \sin(\delta + \psi) - \dot{y} \cos(\delta + \psi) - \dot{\psi} L \cos(\delta) (\sin^2(\psi) + \cos^2(\psi)) = 0 \quad (2.11)$$

$$\dot{x}\sin(\delta + \psi) - \dot{y}\cos(\delta + \psi) - \dot{\psi}L\cos(\delta) = 0 \quad (2.12)$$

denklemleri elde edilir. \dot{x} yerine $\cos(\psi)$ ve \dot{y} yerinde $\sin(\psi)$ kullanıldığında, v_x ve v_y , (2.13) ve (2.14) denklemlerindeki gibi ifade edilir ve $\dot{\psi}$, savrulma hızı bu denklemlerden (2.15) denklemindeki gibi ifade edilir.

$$\dot{x}_r = v_x \cos(\psi) \quad (2.13)$$

$$\dot{y}_r = v_x \sin(\psi) \quad (2.14)$$

Denklem (2.15)'te bulunan trigonometrik açılımlar yapıldığında, (2.16), (2.17) ve (2.18) denklemleri elde edilir.

$$\dot{\psi} = \frac{\dot{x}\sin(\delta + \psi) - \dot{y}\cos(\delta + \psi)}{L\cos(\delta)} \quad (2.15)$$

$$\dot{\psi} = \frac{v_x \cos(\psi)(\sin(\psi)\cos(\delta) + \cos(\psi) + \sin(\delta))}{L\cos(\delta)} - \frac{v_x \sin(\psi)(\cos(\psi)\cos(\delta) - \sin(\psi)\sin(\delta))}{L\cos(\delta)} \quad (2.16)$$

$$\dot{\psi} = \frac{v_x \sin(\psi)(\sin(\psi)\cos(\delta)\cos(\psi) + \cos^2(\psi)\sin(\delta))}{L\cos(\delta)} - \frac{v_x \cos(\psi)(\sin(\psi)\cos(\delta)\cos(\psi) - \sin^2(\delta)\sin(\delta))}{L\cos(\delta)} \quad (2.17)$$

$$\dot{\psi} = \frac{v_x((\sin^2(\psi) + \cos^2(\psi))\sin(\delta))}{L\cos(\delta)} \quad (2.18)$$

$$\dot{\psi} = \frac{v_x \sin(\delta)}{L\cos(\delta)} = \frac{v_x \tan(\delta)}{L} \quad (2.19)$$

Özetlendiğinde (2.20), (2.21) denklemleri elde edilir. Bu denklemlerden x ve y yönündeki hızlar elde edilir ve araç modeli ihtiyacına göre gerçekleştirilebilir.

$$\dot{x} = v_x \cos(\psi) \quad (2.20)$$

$$\dot{y} = v_x \sin(\psi) \quad (2.21)$$

Denklem (2.22) ile kinematik araç modelinin yalpa oranı bulunmaktadır. Simülasyonda direk olarak bu formül kullanılarak kinematik araç modeli gerçekleştirilebilir. Kinematik araç modeli düşük hızlı uygulamalarda kullanılabilen son derece yaygın uygulanan bir modeldir. Düşük hızlarda tekerlere etkiyen kuvvetin daha az olması

dolayısı ile gerçek araç ve kinematik araç modeli benzer sonuç vermektedir. Uygulanması kolaydır hızlı sonuç alabilmek için kullanılabilir.

$$\psi = \frac{v_x \tan(\delta)}{L} \quad (2.22)$$

2.2 Dinamik Araç Modeli

Daha yüksek hızlarda, kuvvetlerin tekerlekler üzerindeki etkisi arttığından kinematik model doğru sonuç vermeyebilir [12]. Bunun için kuvvetlerin de göz önünde bulundurulduğu dinamik araç modeli kullanılmalıdır. Boylamsal düzlemdeki hızı bu tez için sabit alındığından, iki serbestlik dereceli denklemler kullanılır. Bu denklemler yanal pozisyon ve savrulma açısına ilişkindir. Savrulma açısı global koordinat düzlemindeki X eksenine göre ölçülür. Bu tezde boylamsal hız sabit kabul edilmiştir ve $v_x = 0$ varsayılmıştır. Bunun sonucunda F_{xf} , (2.23) denkleminde olduğu gibi kullanılmıştır. Şekil 2.2’de, dinamik araç modelinin bir gösterimidir, ön ve arka tekerleğe uygulanan x eksenini yönünde bir kuvvet olmadığı varsayılmıştır. Yanal kuvvetler F_{yf} ve F_{yr} ile gösterilmiştir.

$$F_{xf} = 0 \quad (2.23)$$

Kuvvet denklemi (2.24) denklemindeki gibi oluşturulur.

$$m\dot{v}_y = F_{yf} \cos(\delta) + F_{yr} + mv_x \dot{\psi} \quad (2.24)$$

Atalet momentini denklemi ise (2.25) denklemindeki gibi oluşturulmuştur.

$$I_z \ddot{\psi} = L_f F_{yf} \cos(\delta) - L_r F_{yr} \quad (2.25)$$

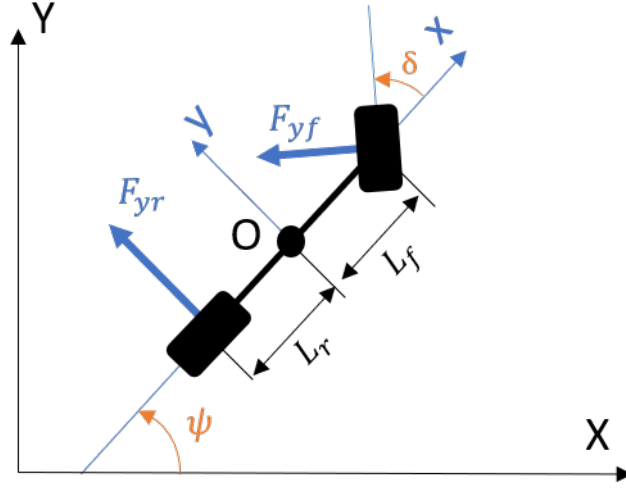
$\dot{\psi}$, z eksenindeki açısal savrulma ivmesini temsil eder. Tekerleklerdeki kayma, (2.26) ve (2.27) denklemlerinde ifade edilmiştir.

$$\alpha_f = \tan^{-1} \left(\frac{v_y + L_f \dot{\psi}}{v_x} \right) - \delta \quad (2.26)$$

$$\alpha_r = \tan^{-1} \left(\frac{v_y - L_r \dot{\psi}}{v_x} \right) \quad (2.27)$$

Tekerleklerdeki kaymaların küçük olduğu varsayılmıştır, dolayısıyla kuvvetlere etkileyen yanal kuvvetler (2.28) ve (2.29) denklemlerindeki gibi olur.

$$F_{yf} = -C_f \alpha_f \quad (2.28)$$



Şekil 2.2 : Dinamik Araç Modeli.

$$F_{yr} = -C_r \alpha_r \quad (2.29)$$

C_f ve C_r ön ve arka tekerleğin dönüş dirençleridir. (2.24) ve (2.25) denklemlerinde F_{yf} ve F_{yr} yerlerine konursa, (2.30) ve (2.31) elde edilir.

$$m\dot{v}_y = -C_f \left[\tan^{-1} \frac{v_y + L_f \dot{\psi}}{v_x} - \delta \right] \cos(\delta) - C_r \left[\tan^{-1} \frac{v_y - L_r \dot{\psi}}{v_x} \right] + m v_x \dot{\psi} \quad (2.30)$$

$$I_z \ddot{\psi} = -L_r C_f \left[\tan^{-1} \frac{v_y + L_f \dot{\psi}}{v_x} - \delta \right] \cos(\delta) + L_r C_r \left[\tan^{-1} \frac{v_y - L_r \dot{\psi}}{v_x} \right] \quad (2.31)$$

Bu denklemler serisinin sonucunda, arabanın pozisyonu (2.32) ve (2.33) denklemleriyle çıkartılmış olur. Bu denklemler lineer olmayan denklemlerdir.

$$\dot{x} = v_x \cos(\psi) - v_y \sin(\psi) \quad (2.32)$$

$$\dot{y} = v_x \sin(\psi) + v_y \cos(\psi) \quad (2.33)$$

Denklem (2.34) ve denklem (2.35)'te lineerleştirilmiş dinamik denklemler elde edilmiştir. Lineer denklemler bazı uygulamalar için uygun olabilir. Bu tezde, lineerleştirilmiş kuvvet denklemleri kullanılmıştır.

$$m\dot{v}_y = \frac{-C_f v_y - C_f L_f \dot{\psi}}{v_x} + C_f \delta + \frac{-C_r v_y + C_r L_r \dot{\psi}}{v_x} - m v_x \dot{\psi} \quad (2.34)$$

$$I_z \ddot{\psi} = \frac{-L_f C_f v_y - L_f^2 C_f \dot{\psi}}{v_x} + L_f C_f \delta + \frac{L_r C_r v_y - L_r^2 C_r \dot{\psi}}{v_x} \quad (2.35)$$

Çizelge 2.1, aracın kütlesi, ön ve arka tekerlek dönüş dirençleri, atalet momentini ve ağırlık merkezinden ön ve arka tekerleklere olan uzaklıkları içermektedir. Bu

Çizelge 2.1 : Dinamik Bisiklet Modeli parametreleri.

Parametre	Değer	Birim
m	2220	kg
C_f	600	N/deg
C_r	600	N/deg
I_z	1549	kgm^2
L_f	1.425	m
L_r	1.425	m

görülen parametreler araç modelini oluştururken kullanılan sabitlerdir. Araç modeli gerçekleşmesi istendiğinde eğer bu parametreler kullanılırsa, simülasyon ortamında kararlı çalışan bir dinamik araç modeli elde edilmektedir. Bu araç modeli farklı simülasyonlar için kullanılabilir.

3. ENGELDEN KAÇMA ALGORİTMALARI

Engelden kaçma sistemleri; mesafe ölçme, haritalama, konum gibi verileri elde edecek sensörleri kullanarak, olası bir çarpışma durumunda çarpışma önleme veya hasarı azaltma amacı ile otomotiv sektöründe adı geçen sistemlerdir. Bu sistemlere her geçen gün yeni yeni fonksiyonlar eklenerek bu sistemlerin otomotiv sektöründe gelişimleri devam etmektedir. Bu tezde engelden kaçma algoritmalarından Boşluğu Takip Et metodu detaylı bir biçimde incelenmiş ve otonom sollama manevrasına uygunluğu sunulmuştur. Engelden kaçma algoritmalarından böcek algoritmaları, yapay potansiyel alan metodu ve Boşluğu Takip Et metodu araştırılmıştır.

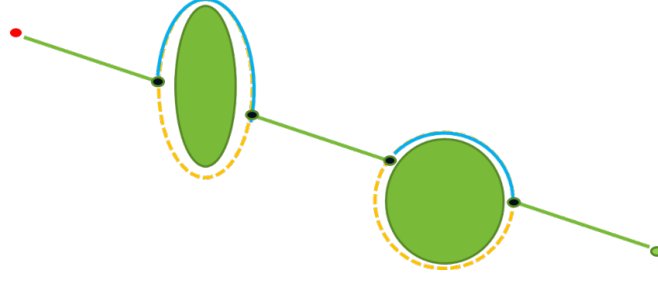
3.1 Böcek Algoritmaları

Böcek algoritmaları ilk olarak bir böceğin hareketinden yola çıkılarak, hedefine doğru ilerleyen bir böceğin hareketine benzetilerek oluşturulmuştur [13]. Farklı varyasyonları bulunmaktadır. Bu varyasyonların karşılaştırılması [14]'te yapılmıştır. Böcek algoritmaları genel hatlarıyla basittir ve işlemciye yük olmazlar. Eski bir algoritma olması sebebi ile bir çok yöntem için temel oluşturabilmektedir. Ayrıca lokal minima problemi yaşayan algoritmalar böcek algoritması ile birleştirilerek lokal minima problemi yaşamadan hedefe varabilmektedir. Bu böcek algoritması kadar uzun olmayan ve lokal minima problemi olmayan hibrit bir engelden kaçma algoritması elde etmeyi sağlamaktadır. Bu özelliği ile iyi bir yardımcı algoritma olduğu söylenmektedir.

3.1.1 Böcek-1 algoritması

Böcek-1 algoritması, robot hedefine doğru hiç bir kontrol yapmadan ilerler, eğer önüne bir engel çıkarsa o engelin etrafından dolanmaya başlar, etrafını tam tur çevrelerken hedefine en yakın olan noktayı bulmaya çalışır, hedefine en yakın olan noktayı tutar ve ikinci tura başlar. Tuttuğu noktaya geldiğinde ise, hedefine doğru hareket edecek şekilde ayrılır. Böcek-1 algoritması Şekil 3.1'de görüldüğü gibidir. Kırmızı nokta

başlangıç noktasını, yeşil nokta hedef noktasını temsil eder. Robotun engelin çevresini dolanması sarı kesikli çizgi ile gösterilmiştir. Böcek-1 algoritmasında hedefe doğru

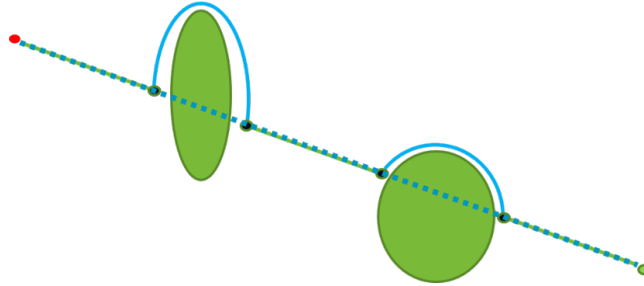


Şekil 3.1 : Böcek-1 algoritmasının gösterimi.

giderken, engelin etrafında bir kez tur atması yetmediğinden yol uzun sürer. Bu da verimsiz bir rotaya neden olur, verimsiz rota zamanda kayıplara sebep olur.

3.1.2 Böcek-2 algoritması

Böcek-1 algoritmasının yol uzunluğu çok fazladır. Bunu kısaltmak için [15]'de da bahsedilen böcek-2 algoritması geliştirilmiştir. Böcek-2 algoritmasında robotun başlangıç ve hedef noktası arasında düz bir çizgi olduğu varsayılmıştır böylece robot hedefe doğru olan hareketinde engelle karşılaştığında engelin çevresindeki en yakın noktayı bulmak için tüm çevreyi dolanmak yerine varsayılan çizgi üzerine denk gelen noktaya kadar ilerleyip oradan ayrılmaktadır. Şekil 3.2'de, Böcek-2 algoritması gösterilmiştir. Kırmızı nokta başlangıç noktası, yeşil nokta hedef noktasıdır. Kesikli mavi çizgi hedef olan hayali çizgiyi temsil eder. Böcek-2 algoritmasındaki yol



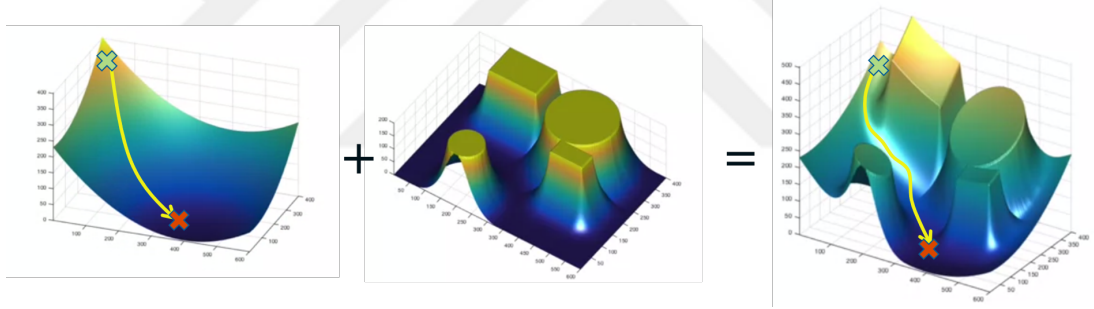
Şekil 3.2 : Böcek-2 algoritmasının gösterimi.

uzunluğu Böcek-1 algoritmasındaki yol uzunluğuna göre daha kısa olmasına rağmen, her iki böcek algoritmalarında, robot şeklin etrafını dolaştığından yol çok uzundur ve optimal değildir. Avantaj olarak da etrafını dolaştığından bir çıkış noktası bulur ve lokal minimada takılma problemi yaşamaz. Araç için bir sürü sınırlar bulunmaktadır. Araç

her zaman engelin etrafında tam tur atacak dinamiğe sahip olamayabilir dolayısıyla otomotiv sektöründe kullanılması verimli olmayacaktır.

3.2 Yapay Potansiyel Alan Metodu

Yapay potansiyel alan metodu engelden kaçma algoritması olarak kullanılır [16]. Burada başlangıç noktası, hedef noktası ve objeleri yarattığı bir yapay potansiyel alan varlığından söz edilir. Şekil 3.3, yapay potansiyel alan metodunun bir gösterimidir. Solda bulunan resim eğer herhangi bir engel olmasaydı potansiyel alanın nasıl görüldüğü ve yolun nasıl olacağını göstermektedir. Ortadaki resim engellerin potansiyel alanlarını gösterir. En sağdaki resimde de başlangıç ve bitiş noktalarının yapay potansiyel alanlarına, engellerin yapay potansiyel alanlarının eklenmesi ile planlanacak yol gösterilmiştir. Kırmızı nokta başlangıç noktası ve yeşil nokta hedef noktasıdır. Sarı yol engelli ve engelsiz iki durum için de optimal yolu gösterir. YPA'da başlangıç noktası ve engeller itme alanlarını oluşturur. Hedef ise çekme alanını



Şekil 3.3 : Yapay Potansiyel Alan metodunun gösterimi [3].

oluşturur. İtme kuvveti, engel ile arasındaki mesafe ile ters orantılıdır. Denklem (3.1) itme alanının nasıl hesaplandığını gösterir ve itme alanı V_r ile ifade edilir. İtme alanından x yönünde F_{r_x} denklem (3.2)'deki gibi ve y yönünde F_{r_y} denklem (3.3)'teki gibi hesaplanır.

$$V_r(x,y) = \frac{K_r}{\sqrt{(x-x_0)^2 + (y-y_0)^2}} \quad (3.1)$$

$$F_{r_x} = -\frac{\nabla V_r}{\nabla x} \quad (3.2)$$

$$F_{r_y} = -\frac{\nabla V_r}{\nabla y} \quad (3.3)$$

K_r kalibrasyon katsayısıdır, uygulamaya göre ayarlanabilir. İtici kuvvet, engelle yaklaştıkça etkisi artar böylece robot engelden kaçma hareketi yapar. Başlangıç

noktasında da itici kuvvetin değeri çok yüksektir başlangıç noktasından uzaklaştıkça etkisi azalır. Çekme alanı, V_a ile gösterilir ve denklem (3.4)'teki gibi hesaplanır. V_a 'dan x yönündeki çekici kuvvet F_{a_x} denklem (3.5) ile y yönündeki çekme kuvveti F_{a_y} denklem (3.6) ile hesaplanmıştır.

$$V_a(x,y) = \frac{K_a}{\sqrt{(x-x_0)^2 + (y-y_0)^2}} \quad (3.4)$$

$$F_{a_x} = -\frac{\nabla V_a}{\nabla x} \quad (3.5)$$

$$F_{a_y} = -\frac{\nabla V_a}{\nabla y} \quad (3.6)$$

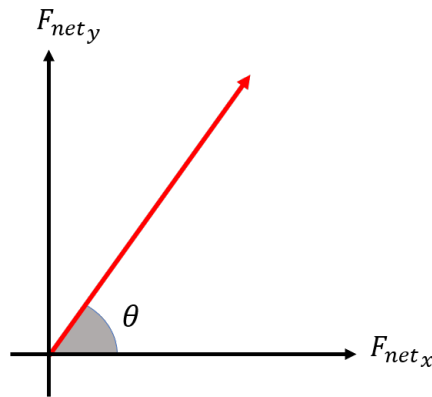
Robotun üzerine etkiye net kuvvet x yönünde F_{net_x} ve y yönünde F_{net_y} ile ifade edilir. Denklem (3.7), F_{net_x} 'in elde edilmesini gösterir. Denklem (3.8), F_{net_y} 'nin elde edilmesini gösterir.

$$F_{net_x} = F_{a_x} - (F_{ri_x} + F_{ro_x}) \quad (3.7)$$

F_{a_x} hedef noktasının x yönündeki çekme kuvvetini, F_{ri_x} i. engelin x yönündeki itme kuvvetini ve F_{ro_x} başlangıç noktasının x yönündeki itme kuvvetini temsil eder.

$$F_{net_y} = F_{a_y} - (F_{ri_y} + F_{ro_y}) \quad (3.8)$$

F_{a_y} hedef noktasının y yönündeki çekme kuvvetini, F_{ri_y} i. engelin y yönündeki itme kuvvetini ve F_{ro_y} başlangıç noktasının y yönündeki itme kuvvetini temsil eder. Robotun gitmesi istenen direksiyon açısı θ , bu kuvvetlerin arasındaki açı ile (3.9) denklemindeki gibi hesaplanır. Şekil 3.4'teki gibi gösterilir. Bu kuvvetlerin toplamı



Şekil 3.4 : θ açısının gösterimi.

yapay potansiyel alanı oluşturur ve ortaya çıkan yol optimaldir. Öte yandan, lokal minimada takılabilir.

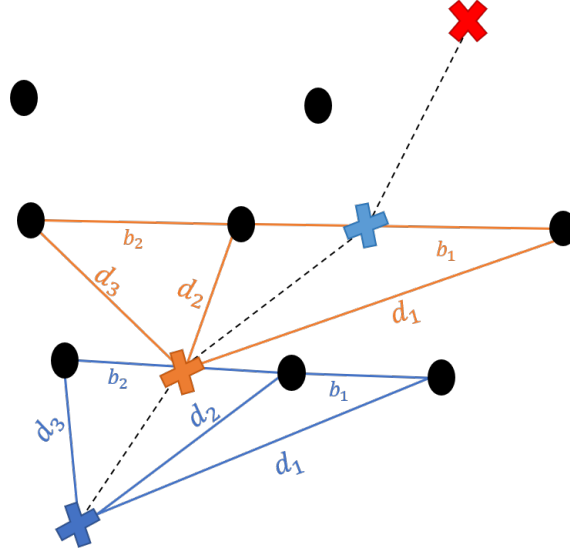
$$\theta = \tan^{-1} \left(\frac{F_{net_y}}{F_{net_x}} \right) \quad (3.9)$$

Ayrıca kalibre edilebilir katsayılar fazla olduğundan, her koşul için değiştirilmesi gerekebilir bu da kullanım zorluğuna neden olur.

3.3 Boşluğu Takip Et Metodu

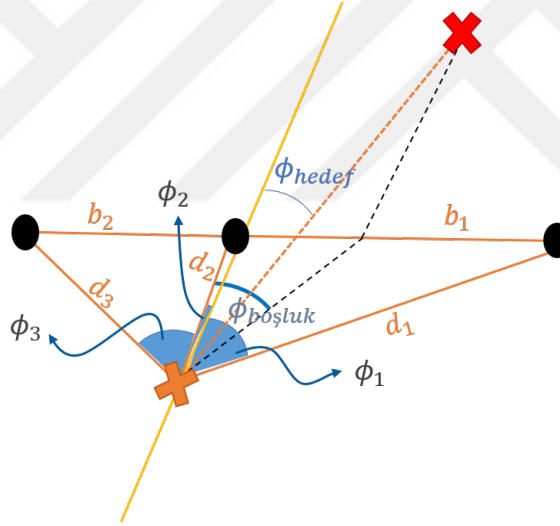
Boşluğu takip et metodu ilk olarak [17]'te yeni geliştirilmiş bir algoritma olarak tanıtılmış ve literatürdeki yerini almıştır. Bu metot, bulunduğu konuma göre çevre-
sindeki objeleri hesaba katarak bu objelerin arasındaki boşlukları hesaplamaktadır. En
yüksek genişliğe sahip boşluktan, hedef noktasını hesaba katarak, en optimal şekilde
geçecek bir direksiyon açısı oluşturmaktadır. Boşluğu Takip Et metodu için iki adet
geliştirme yapılmıştır [18]. Bunlardan bir tanesi; en yüksek genişliğe sahip boşluğun
değişme sıklığı arttığın ortaya çıkan zigzag hareketidir. Bir diğeri ise, bazen en geniş
boşluk en iyi çözüm olmayabilir, hedefe gidiş yolunu çok uzatabilir. Bunun için en iyi
boşluk yerine en iyi boşluk seçimi yapılmaktadır. Boşluğu Takip Et metodunun her iki
uygulamasında boylamsal hız sabit kabul edilmiştir. Açısal ve lineer hızların hesaba
katıldığı diğeri bir çalışma ise [19]'da yapılmıştır. Burada Dinamik Pencere yaklaşımı
(DWA) ile Boşluğu Takip Et metodunun güçlü yanları birleştirilmiştir.

Şekil 3.5'te Boşluğu Takip Et metodu genel yapısıyla gösterilmiştir. Burada
siyah noktalar objeleri temsil eder. Mavi x başlangıç noktasını gösterir. Başlangıç
noktasındaki robot kendine en yakın objeleri ve bu objelere olan uzaklıklarını (d_1 ,
 d_2 ve d_3) tayin eder. Uzaklıklara göre seçtiği objelerin arasındaki boşluk mesafelerini
(b_1 , b_2) hesaplar. Aralarından en geniş boşluğu seçip, hedefe ulaşması gereken açı
ile ağırlıklandırarak bir direksiyon açısı oluşturur ve turuncu x'ye gelir. Turuncu
x'e geldikten sonra aynı işlemi tekrarlar ve açık mavi x'e gelir en sonunda sadece
hedef noktası kaldığından direk hedefe doğru ilerler. Şekil 3.6 açılarının gösterimini
daha detaylı bir şekilde gösterir. Burada turuncu x başlangıç noktası, kırmızı x hedef
noktasıdır. Turuncu x noktasından bakan bir robot üç adet engel görecektir. Bu engeller
ile olan uzaklıklarını ($d_1, d_2, d_3 \dots d_n$) bulur, her engele olan açılarını ($\phi_1, \phi_2, \phi_3 \dots$
 ϕ_n) bulur. ϕ_1, ϕ_2 ve ϕ_3 'e göre hangi iki engel arasındaki boşluğun en geniş olduğunu
hesaplar. Bulduğu boşluktan geçmek üzere ϕ_{bosluk} 'u hesaplar. Hedefle arasındaki açı,
 ϕ_{hedef} ile gösterilmiştir ve bu da hedefin yeri bilindiğinden bilinmektedir. Savrulma
açısı ϕ_{final} ile gösterilmektedir, ϕ_{hedef} ve ϕ_{bosluk} ağırlıklandırılarak bulunur. Sarı çizgi
robotun o andaki savrulma açısını gösterir. Kesikli siyah çizgi boşluk açısı gösterir



Şekil 3.5 : Boşluğu Takip Et metodunun genel gösterimi.

ve kesikli turuncu çizgi de hedef açısını göstermektedir. Hesaplanan direksiyon açısı, θ , kesikli turuncu çizgi ve kesikli siyah çizgi arasında olacaktır. ϕ_{bosluk} , dekleme



Şekil 3.6 : Boşluğu Takip Et metodunda açılarının gösterimi.

(3.10)'daki hesaplanmaktadır.

$$\phi_{bosluk} = \arccos\left(\frac{d_1 + d_2 \cos(\phi_1 + \phi_2)}{\sqrt{d_1^2 + d_2^2 + 2d_1 d_2 \cos(\phi_1 + \phi_2)}}\right) \quad (3.10)$$

Denklem (3.11), ϕ_{final} 'in nasıl hesaplandığını göstermektedir.

$$\phi_{final} = \frac{\frac{\alpha}{d_{min}} \phi_{bosluk} + \phi_{hedef}}{\frac{\alpha}{d_{min}} + 1} \quad (3.11)$$

d_{min} , en yakın engele olan mesafedir, α değiştirilebilir katsayıdır. d_{min} ne kadar yakın olursa ϕ_{bosluk} 'un ağırlığı artacak, ϕ_{final} engelden kaçmaya daha meyilli olacaktır.

Eğer engel çok uzaktaysa ϕ_{final} , ϕ_{hedef} değerine daha yakın çıkacak böylece hedefe doğru yönelmiş bir hareket elde edilecektir. Bu tezde uygulanabilirliğindeki kolaylık ve tek bir parametre ile istenen sonuç elde edilebildiğinden otonom sollama manevra algoritması olarak Boşluğu Takip Et metodu seçilmiştir. Bölüm 6'de Boşluğu Takip Et metodunun otonom sollamaya nasıl uyarlandığı anlatılmaktadır.





4. HAREKET PLANLAMA ALGORİTMALARI

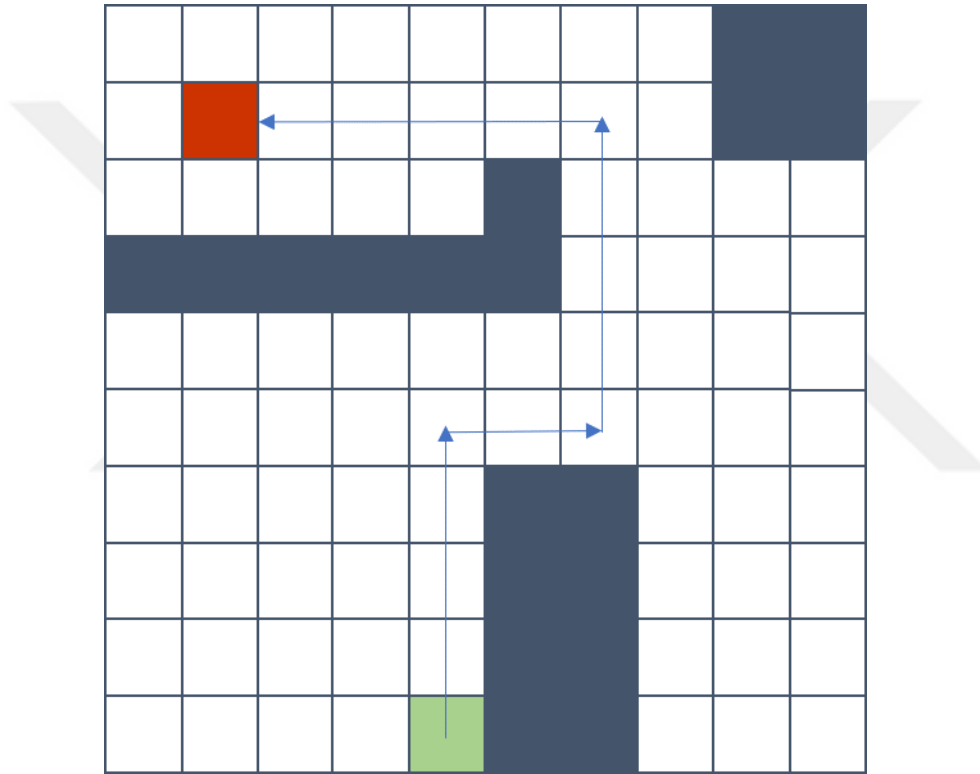
Hareket planlama algoritmaları robotun yapması istenilen hareketleri ayırık hareketler olarak ifade eden fonksiyonlardır. [20]'e göre otonom araçlar için hareket planlaması 4 ana gruba ayrılmaktadır: (1) güzergah planlaması, (2), yol planlanması, (3) manevra seçimi ve (4) Yörünge planlanması. Güzergah planlanmasında global dünyada en optimal rota seçilir. Bu rota aracın dinamiklerinden bağımsızdır ve aracın başaramayacağı manevraları da içerebilir. Yol planlama, manevra seçimi ve yörünge planlaması için aracın dinamikleri hesaba katılır ve gidilebilir bir yol planlanmaktadır. Bu yola uygun hız profilleri oluşturularak hareket kontrolü için gerekli çıktılar hesaplanmaktadır.

Bu bölümde, hareket planlama algoritmalarına genel bir bakış sağlamak amacı ile literatürde bulunan algoritmaların nasıl çalıştıkları, hangi amaçla ve hangi alanlarda kullanıldığı açıklanmıştır.

4.1 Grafik Arama Tabanlı Algoritmalar

Grafik arama tabanlı algoritmalar global hareket planlama algoritmalarından biridir. Bu algoritmalarda bir başlangıç ve bir bitiş noktası olmalıdır. Bu başlangıç ve bitiş noktası arasındaki en kısa mesafeyi bulmayı amaçlar. Bunu gerçekleştirebilmek için çevrenin modeline ihtiyaç duyar. Bu algoritmalara örnek olarak Dijkstra, A*, D* algoritmaları verilebilir. Dijkstra, 1959 yılında geliştirilen bir grafik arama tabanlı algoritmadır [21]. Dijkstra, A* ve D* gibi popüler grafik arama tabanlı algoritmalarının temeli olarak bilinmektedir. Bir arama alanı vardır ve düğümlerle ayrılmıştır. Başlangıç noktası ve bitiş noktası bilinmektedir. Başlangıç ve bitiş noktası arasındaki her düğümle olan bağlantı maliyetlendirilir. Bütün arama alanı tarandığında, en düşük maliyeti olan yol seçilir. Birbiriyle bağlantısı olan her düğüm tarandığından işlem hızı yavaştır. Bu sebeple zamanın önemli olduğu uygulamalar için verimsizdir. [22]'te bahsedildiği üzere, A* verilen bir arama alanı içerisinde uygun yolu bulan verimli

algoritmaların en popülerlerinden biridir. A* algoritmasının her hareketi için bir arama alanına ve bir maliyet fonksiyonuna ihtiyaç vardır. Arama alanı ve maliyet fonksiyonu, alanın en verimli bir şekilde taranıp optimal yolun bulunabilmesi için kullanılmaktadır. Dijkstra algoritmasından farkı A* algoritmasında buluşsal fonksiyonun olmasıdır. Diğer bir deyişle, bağlantısı olan her düğümden yol planlanmaz. Buluşsal fonksiyon ile seçilen düğüme giderek yol planlanmaktadır. Şekil 4.1’de, A* algoritmasının bir uygulaması gösterilmektedir. Bu 10x10’luk bir alandır. Her bir kare bir düğüm noktası demektir. Yeşil kare, başlangıç düğümüdür. Kırmızı kare, hedef düğümüdür ve gri alan ise engelleri temsil etmektedir. A* algoritması kullanılırken her düğüm için



Şekil 4.1 : A* algoritmasının gösterimi.

denklem (4.1)’deki gibi bir maliyet fonksiyonuna ihtiyaç vardır. $g(t)$, bir önceki düğüm ile olan ilişkinin fonksiyonunu temsil etmektedir. $h(t)$ ise buluşsal fonksiyon olarak tanımlanmaktadır ve hedef düğümü ile o anki düğüm ile olan ilişkinin fonksiyonudur. $g(t)$ ve $h(t)$ fonksiyonları genel olarak mesafe seçilebilmektedir. Bu iki fonksiyonun toplamı da o düğümün maliyeti olur.

$$f(t) = g(t) + h(t) \quad (4.1)$$

Örneğin yeşil noktadan yani başlangıç düğümünden başladığı durumda komşu düğümler taranır, engeller taranmaz. Buradan etrafındaki düğümlerin maliyetleri

hesaplanır ve maliyeti düşük olan düğüm seçilir. Maliyeti düşük olan düğüm, komşularının maliyetini hesaplar ve hangi düğümden geldiyse onun bilgisini de ebeveyn düğüm olarak tutar. Böylece şekil 4.1'deki mavi çizgi diğer bir anlamda yol oluşmaktadır. A* algoritmasının kodsusu EK A'da gösterilmektedir. A* algoritması geliştirilirken açık ve kapalı adı altında iki adet liste tutulmaktadır. Açık liste daha ziyaret edilmemiş düğüm bilgilerini saklı tutar. Kapalı liste ise daha önceden ziyaret edilen düğümleri tutmaktadır. A* algoritmasında engeller sabittir. Her yeni adımda bütün maliyetler tekrar hesaplanmaktadır. Bu belleğe fazlaca yük bindirmektedir. Bunun önüne geçebilmek için D* algoritması geliştirilmiştir [23]. D* algoritmasında engellerin hareketleri de hafızaya eklenmektedir ve engellerin hareketlerine göre yol oluşturulmaktadır. Eğer görüş alanına yeni bir engel eklenmezse veya var olan engel görüş alanından çıkmazsa yol aynı kalır. Bu gibi algoritmalar robotta kesikli hareket meydana getirebilir ve aracın dinamiğini hesaba katmadan direk rota oluşturmaktadırlar. Dönüşleri keskindir ve herhangi bir yol yumuşatma fonksiyonu kullanılmadan aracta uygulama yapmak zordur. sollama manevrası için hareketlerin hem pürüzsüz olması hem de hesaplanması kolay olan bir yöntem olması gerekmektedir. A*'da her yeni adımda tüm harita baştan ağırlıklandırılıp, rota oluşturulacağından hesaplama maliyeti de yüksektir.

4.2 Örneklem Tabanlı Algoritmalar

Bu algoritmalar, araç dinamiğine daha uygundur. Araç dinamiğine göre bir yol oluşturmak mümkündür. Olasılıklı Yol Haritası (Probabilistic Roadmap- PRM) algoritması, ilk örneklem tabanlı algoritma olarak gösterilir [24]. Bu algoritmada, konfigürasyon uzayına rastgele noktalar atılır. Böylece, konfigürasyon uzayı örneklenir. Bu rastgele noktalar arasından herhangi bir engele çarpmayacak şekilde bir grafik oluşturmaya çalışır. Her bir tekrarlama, doğru yolu bulma ihtimali daha da artmış olur. Bu tekrarlama sayısı, uygulamadan uygulamaya ve işlemci hızına göre değişebilmektedir. Bir başka örneklem tabanlı algoritma örneği olarak Rastgele Ağacı Hızlı Keşif(Rapidly exploring Random Tree - RRT) metodu söylenebilir [25]. Bu algoritmada, başlangıç noktasından rastgele dallar çizilir, daha sonra bu dallardan hedefe uygun olan dal seçilir ve o seçilen daldan tekrar dallar uzanır. Bu işlem hedefe varıncaya kadar devam eder. Hedefe varıldığında ise seçilen dallar aslında o hareketin

yolu olmuş olur. Bu algoritmanın ayırık zamanda sürekli olarak çalışabilmesi için iyi bir işlemciye ihtiyaç vardır. Algoritmada bahsedilen dallar araç dinamiğine göre oluşturulabilir böylece aracın gidebileceği bir hareket planlaması olmaktadır.

4.3 Matematiksel Fonksiyon Tabanlı Algoritmalar

Bu yaklaşımın amacı, kesik çalışan ve çok işlemci harcayan algoritmaların tersine, hem işlem yükünü azaltmak hem de konforlu bir sürüş sağlamaktır. Burada bir hareketin niteliğine göre bir matematiksel fonksiyon kullanılmaktadır. Bunun için en yaygın olarak kullanılan fonksiyonlar polinomlardır. Polinomlar arasında spline (polinomsal parametrik eğri parçalar) olarak da adlandırılan yörüngeler oluşturulmaktadır. Splinelar robotun kolay takip edebileceği pürüzsüzlükte yol sağlarlar. Başlangıç noktası ve hedef noktası arasında bir sürü seçenekten optimal olanı seçmekle uğraşmak yerine, düzgün bir yörünge oluşturur. Bu yörünge güvenliği ve konforu düşünerek oluşturulur. Konfor ivmenin türeyle orantılıdır. İvmenin türevi yol boyunca düşük değerdeki ivmenin o kadar konforlu bir sürüş gerçekleşmektedir [26]. Bölüm 4.1’de bahsedilen yaklaşımlardan en büyük farklılığı bir alanı taramamasıdır. Denklem (4.2)’de görülen bir polinom ile ifade edilebilirler. Ara değer hesaplama yöntemlerine göre farklı denklemlere sahip olabilirler. Çoğunlukla 4. derece üstü olurlar.

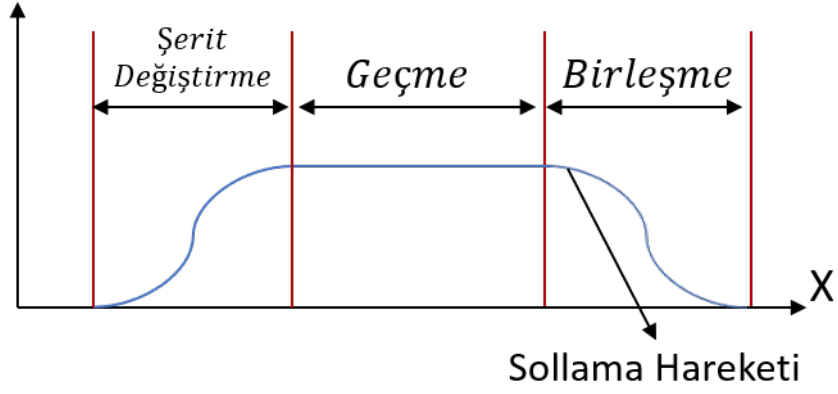
$$y(x) = ax^3 + bx^2 + cx + d \quad (4.2)$$

Bu tezde geometri tabanlı algoritmalarından, otonom sollama manevrası için özel olarak geliştirilen X-sin fonksiyonu kullanılmaktadır. İlk olarak [27]’de bahsedilmiştir. Bu fonksiyon, aracın dinamiğini düşünmektedir aynı zamanda aracın başarılı manevralar yapabilmesini sağlamaktadır. X-sin fonksiyonu ile sollama manevrası, parçalı bir fonksiyon ile ifade edilmektedir. Şekil 4.2’de gösterilen fazlalarla ifade edilmektedir. Şekil 4.3, x-sin fonksiyonunun parametreler ile gösterimidir. Sollama manevrasında şerit değiştirme fazı denklem (4.3)’teki gibi hesaplanmaktadır. L_d sollama fazının boylamsal uzunluğudur.

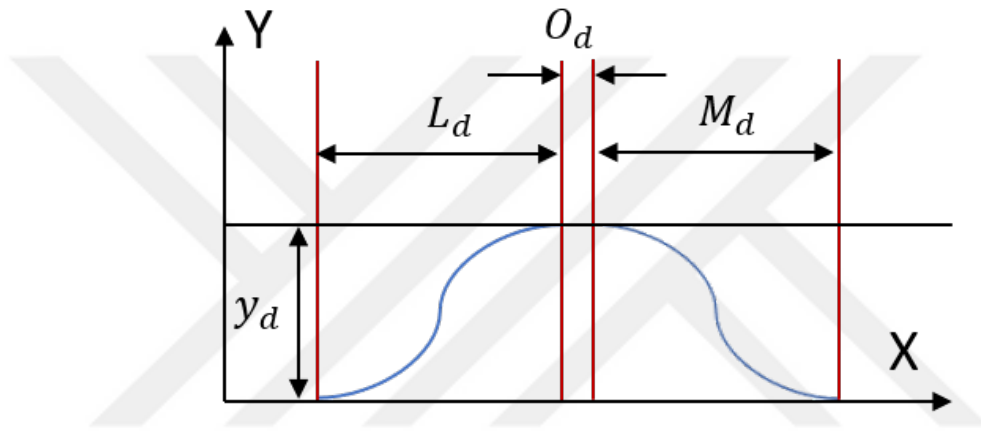
$$y(x) = y_d \left[\frac{2\pi}{L_d}x - \sin \left(\frac{2\pi}{L_d}x \right) \right] x \in [0, L_d] \quad (4.3)$$

Geçme fazı ise, (4.4) denklemindeki gibi kullanılmaktadır.

$$y(x) = y_d x \in [L_d, L_d + O_d] \quad (4.4)$$



Şekil 4.2 : X-sin fonksiyonu ile hareket planı.



Şekil 4.3 : X-sin fonksiyonunun parametreler ile gösterimi.

Birleşme fazında, araç sollama şeridinden hareketi başladığı şeride döner bu hareket ise, (4.5) denkleminde hesaplanmaktadır.

$$y(x) = y_d - y_d \left[\frac{2\pi}{L_d}x - \sin \left(\frac{2\pi}{L_d}x \right) \right] x \in [L_d + O_d, L_d + O_d + M_d] \quad (4.5)$$

Bu tezde X-sin fonksiyonu hareket planlama algoritması olarak X-sin fonksiyonu seçilmiştir. Bu fonksiyon sollama için özel olarak tasarlanan bir fonksiyon olduğu için, diğer yöntemlere göre avantajı vardır. Otonom sollama manevrasının tasarımı bölüm 6.1.1'de detaylı olarak açıklanmıştır ve bölüm 7'da karşılaştırmalı sonuçlar sunulmuştur.



5. HAREKET KONTROL ALGORİTMALARI

Hareket kontrol algoritmaları, robotun istenilen yere gidebilmesi için kendi hareketini ona göre ayarlayan fonksiyonlardır. Bölüm 1'deki araştırmalardan sonra, bu tezde uygulama kolaylığı ve manevralardaki performansı göz önünde bulundurulduğunda iki hareket kontrol algoritması öne çıkmaktadır. Bu bölümde Pure Pursuit ve Stanley metotlarının nasıl çalıştığı, avantajları ve dezavantajları anlatılmaktadır.

5.1 Pure Pursuit Metodu

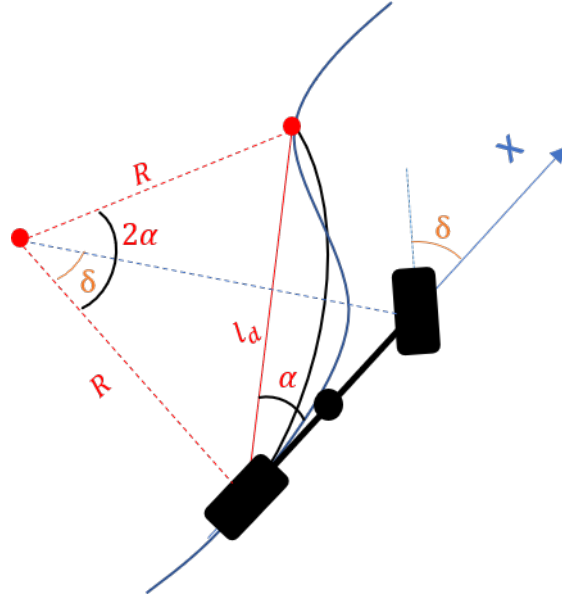
Pure Pursuit, basitliğiyle ve iyi performansıya en çok kullanılan hareket kontrol algoritmalarından biridir. Pure Pursuit metodunun temeli 'ileri bakma' hareketidir [28]. Bu metot her zaman bulunduğu yoldan ileri bakarak uygun direksiyon açısını hesaplamaktadır. Bu 'ileri bakma' yaklaşımı, Pure Pursuit metodunu hareket kontrolü probleminden çıkartıp basit bir geometri problemine dönüştürmektedir. Tekerleklerin kayması ihmal edilmektedir. Şekil 5.1'de Pure Pursuit metodu gösterilmektedir. Pure Pursuit metodunun elde ettiği direksiyon açısı temelde sinüs açılımına dayanmaktadır ve denklem (5.1)'deki gibi hesaplanır. L aracın uzunluğu, δ direksiyon açısı, l_d 'ileri bakma' mesafesi, $\alpha(t)$ l_d kadar uzaktaki yolun noktası ile arka tekerleği bağlayan yay ile aracın savrulma açısı arasındaki açıdır. R aracın dönüş yarıçapıdır.

$$\delta_{PP}(t) = \tan^{-1} \left(\frac{2L \sin(\alpha(t))}{l_d} \right) \quad (5.1)$$

İlk önce aracın global koordinatlardaki yeri ve oryantasyonu bulunur. Daha sonra arka tekerleğin yola olan en yakın mesafesi bulunur. Yolun üzerindeki l_d uzaklıkta bir hedef noktası aranır. Hedef noktası global koordinat düzleminde aracın lokal koordinat düzlemine dönüştürülür. Buradan denklem 5.2 ile $\alpha(t)$ bulunur.

$$R = \frac{l_d}{2 \sin(\alpha)} \quad (5.2)$$

l_d , PID kontrolcüsündeki p kazancı gibi davranır. Eğer l_d değeri düşürülürse, araç yolu daha kusursuz takip etmeye niyetlenir fakat bu kontrol sinyalinin sürekli değişmesi



Şekil 5.1 : Pure Pursuit metodunun gösterimi.

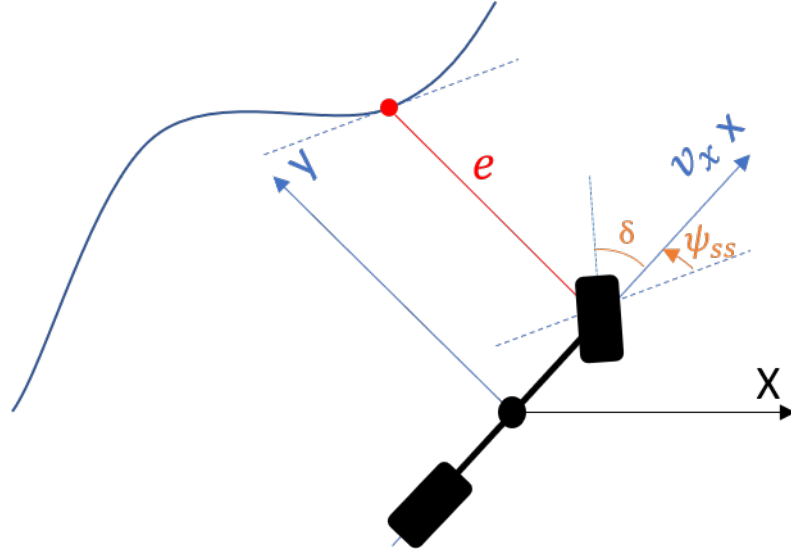
demek olduğundan osilasyon yaratabilir. l_d değeri büyüdükçe, hareket kontrolü daha pürüzsüz olmaktadır diğer yandan bu yolun köşelerini kaybedip kararsızlığa yol açabilir.

5.2 Stanley Metodu

Stanley metodu 2005 yılı DARPA Yarışması'nda kullanılmıştır. Metot, o yarışmada Stanley Üniversitesi tarafından Stanley ismini verdikleri otonom araçlarında kullanılmıştır [29]. Bu metotta, Pure Pursuit metodundaki 'ileri bak' yaklaşımı yerine, yoldaki en yakın noktaya olan yanal uzaklık yani yol takip hatası ve savrulma açısı hatası kullanılmıştır. Stanley metodu, şekil 5.2'de gösterilmiştir. Stanley metodunun elde ettiği direksiyon açısı denklem (5.3)'teki gibi bulunur.

$$\delta(t) = \psi(t) + \tan^{-1} \left(\frac{ke(t)}{v_x(t)} \right) \quad (5.3)$$

Buradaki ψ , savrulma açısı hatası, k kalibre edilebilir sabit, e , yol takip hatası ve v_x aracın boylamsal hızıdır. Stanley metodu, en verimli hareket planlama algoritmalarından biridir. Yol takip hatasının direk kullanımı, istenen yolun elde edilmesini sağlar. Denklem (5.3)'te yol takip hatası boylamsal araç hızı ile ölçeklendirilir. Stanley metodu kararlıdır ve bu da onu diğer yöntemlerden ayıran güçlü bir özelliktir. Stanley metodu sürekli yan tarafında bir yol arayıp oradan bir direksiyon açısı oluşturması gerektiğinden kesikli yollarda çok iyi sonuç vermez. Ayrık elemanları



Şekil 5.2 : Stanley metodunun gösterimi.

olan bir yolda, Stanley metodunda 'ileri bak' yaklaşımı olmadığından yol takip hatası çok büyük olmaktadır. Bu, araca bir direksiyon açısı oluşturmayacağından, yanında bir köşeye gelene kadar yoldan sapar.

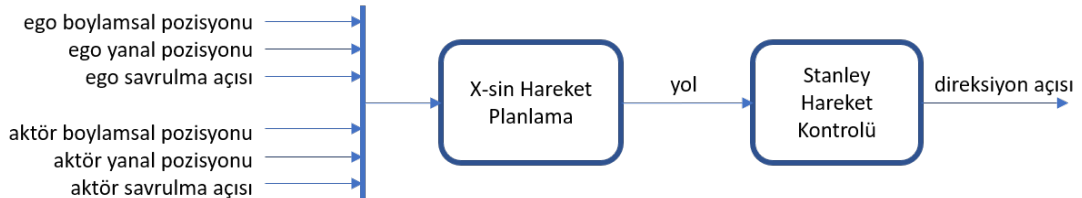


6. OTONOM SOLLAMA MANEVRASI TASARIMI

Trafik kazalarının %60'ı şerit değiştirirken meydana gelmektedir [30]. Bu kontrol edilmesi gereken alanın genişlemesinden kaynaklanır. Sollama manevrası sürücünün dikkatsizlik oranının arttığı bir harekettir. Literatürde, Karışık Gözlemlenebilir Markov Karar Süreci (MOMDP) sollama manevrası problemi için bir çözüm üretmiştir [31]. Fakat bu çözüm hesaplama karmaşıklığından ötürü, gerçek hayattaki uygulamalar için uygun değildir. Ayrıca ayrık durum uzayı gösterimi yöntemin çalışmasını kısıtlar. Bu tezde otonom sollama manevrası tasarlanmıştır. Bu bölümde, Boşluğu Takip Et metodunda sollamaya nasıl adapte edildiği anlatılmaktadır. Ayrıca sollama manevrasının planlaması X-sin fonksiyonu ile yapıp, hareket kontrolü de Stanley metodu ile sağlanarak BTE ile karşılaştırması yapılmıştır. Sollama manevrası 3 ana bölüme ayrılmıştır. Birincisi şerit değiştirme, ikincisi geçme, sonuncusu da birleşme fazı olarak tanımlanmıştır. Sollama manevrasında, sollamayı yapacak olan araç ego araç olarak, sollanan araç aktör araç olarak isimlendirilmiştir.

6.1 Otonom Sollama Manevrası: Planlama ve Kontrol

Bu bölümde klasik yaklaşım ve önerilen yaklaşım ile gerçekleştirilmiş iki model anlatılmıştır. Klasik yaklaşımda, hareket planlanarak bir yol elde edilir. Daha sonra bu yolu takip edecek bir kontrolcü tasarlanır. Klasik yaklaşımın konsepti şekil 6.1'deki gibidir. Bu tezde, hareket planlama için X-sin fonksiyonu kullanılmaktadır. X-sin



Şekil 6.1 : Klasik yaklaşım ile sollama.

funksiyonu sollama manevrası tasarlar. Bu manevra Stanley kontrolcüsü tarafından gerçekleştirilir. Önerilen yaklaşımda, planlama ve kontrol bir arada yapılmaktadır.

Herhangi bir planlayıcıya ihtiyaç duymaksızın direksiyon açısı elde edilir. Bu yaklaşım için Boşluğu Takip Et metodu kullanılmıştır. Bu yaklaşımın konsepti şekil 6.2'deki gibidir. Normalde Boşluğu Takip Et metodu çıkış olarak istenen savrulma açısını



Şekil 6.2 : Önerilen yaklaşım ile sollama manevrası.

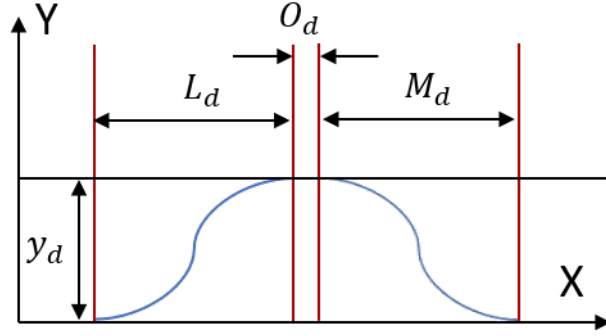
hesaplamaktadır. Savrulma açısı ile direksiyon açısı arasında kontrolcü eklenebilir. Fakat bu tezde, direk olarak direksiyon açısı olarak kullanılmıştır diğer bir anlamda P kontrolcüsünün değeri 1 seçilmiştir. Dolayısıyla savrulma açısı ile direksiyon açısı arasında bir kontrolcüye gerek duyulmadan model gerçekleşmiştir.

6.1.1 X-sin fonksiyonu ile otonom sollama manevrasının planlaması

X-sin fonksiyonu, sollama problemi için geliştirilmiş bir matematiksel formül olarak geliştirilmiştir [27]. X-sin fonksiyonu yanal ve boylamsal pozisyonları hesaplayan bir parçalı fonksiyondur. Şekil 6.3, sollama manevrasının rotasını göstermektedir. Sollama algoritması iki yaklaşımda da Şekil 6.4'te görülen akış kullanılmıştır. Hedef noktası aktör araç ile hareket etmektedir. Hedef noktası uzaklığı olarak birleşme fazının son elemanı olarak seçilmiştir. y_d şerit mesafesini yanal uzunluğunu, L_d şerit değiştirme fazındaki rotayı, O_d geçme fazındaki rotayı ve M_d birleşme fazındaki rotayı temsil eder. M_d , L_d ile aynı seçilmiştir. Denklem (6.1), X-sin parçalı fonksiyonun matematiksel formülünü gösterir. Bu formülde, x anlık x eksenindeki pozisyonu ve y o anlık y eksenindeki pozisyonunu gösterir.

$$y(x) = \begin{cases} y_d \left[\frac{2\pi}{L_d}x - \sin \left(\frac{2\pi}{L_d}x \right) \right] & x \in [0, L_d] \\ y_d & x \in [L_d, L_d + O_d] \\ y_d - y_d \left[\frac{2\pi}{L_d}x - \sin \left(\frac{2\pi}{L_d}x \right) \right] & x \in [L_d + O_d, L_d + O_d + M_d] \end{cases} \quad (6.1)$$

L_d parametresi seçilirken dikkat edilmesi gereken hususlar vardır. Aracın konforu ivmeye aracın ani hız değişimlerine yani ivmeye bağlıdır. Dolayısıyla ivmeye göre, L_d de değişmektedir. L_d 'nin formülü denklem (6.2)'de görülmektedir. Normalde [27]'de, konforlu bir sürüş için olabilmesi için ivme aralığı $1-4 \text{ m/s}^2$ olarak belirtilmektedir.



Şekil 6.3 : X-sin fonksiyonu gösterimi.

Bu tezde a_{max} , $3m/s^2$ seçilmiştir, $2.5m/s^2$ ve $3.5m/s^2$ için de denenip, simülasyon sonuçları eklenmiş ve karşılaştırmaları bölüm 7’de mevcuttur.

$$L_d = v_x \sqrt{\frac{2\pi y_d}{a_{max}}} \quad (6.2)$$

Her a_{max} değeri için v_x , 20, 25 ve 30 m/s olarak seçilmiş bu hızlardan farklı L_d ’ler elde edilmiştir. Bu sonuçlar da karşılaştırılıp bölüm 7’de sunulmaktadır.

6.1.2 Stanley metodu ile otonom sollama manevrasının kontrolü

Stanley ve Pure Pursuit metotları hareket kontrolü için kullanılan çok yaygın iki metottur [10]. Stanley metodunun kıvrımlı yollarda gösterdiği performans Pure Pursuit’e göre daha iyidir [11]. DARPA yarışmalarının araçlarında Stanley metodu daha yaygın olarak kullanılmaktadır [29]. Bu sebeplerden ötürü sollama manevrasının kontrolü için Stanley metodu seçilmiştir. Hareket planlama, X-sin fonksiyonu tarafından tamamlandıktan sonra, Stanley metodu o yolu araca takip ettirmek üzere bir direksiyon açısı üretmektedir. Bu direksiyon açısı temel olarak, aracın yol ile arasındaki en yakın noktaya olan mesafesini hata olarak kabul eder ve yolun üzerinde gitmek için bu hatayı sıfırlamaya çalışır. Bunu da hatayı boylamsal hıza bölüp bir k parametresi ile çarpıp arktanjanantını hesaplar. Buradan çıkan değeri de aracın savrulma açısı ve en yakın noktanın savrulma açısının farkıyla toplar. Direksiyon açısı, θ , hesaplanması denklem (5.3)’te gösterildiği gibidir. Şekil 5.2, Stanley metodunun gösterimidir. Yol takip hatası, aracın pozisyonu ile en yakın noktanın pozisyonu arasındaki farktır ve $e(t)$ ile gösterilmektedir. Yanal hata, $\psi(t)$ ile gösterilmektedir ve bu da aracın yol açısı ile en yakın noktanın savrulma açısı arasındaki farktır. v_x aracın boylamsal hızı. k ’da kalibrasyon değişkenidir ve bu tezde en iyi takibi sağlaması adına

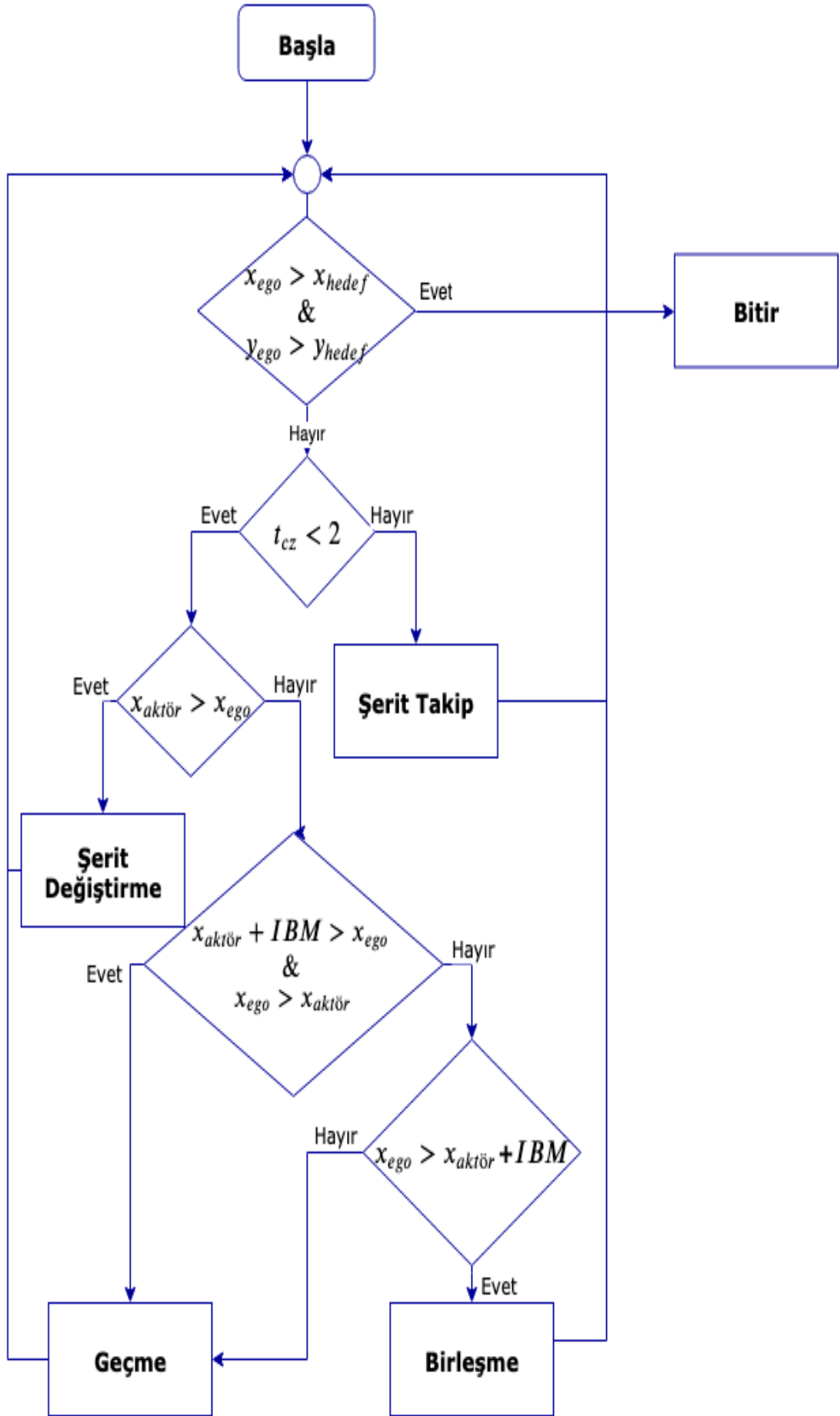
farklı değerler denenmiş ve değeri 10 olarak seçilmiştir. Bu değer ile en iki takibin yapıldığı gözlemlenmiştir.

6.1.3 Boşluğu Takip Et metodu ile otonom sollama

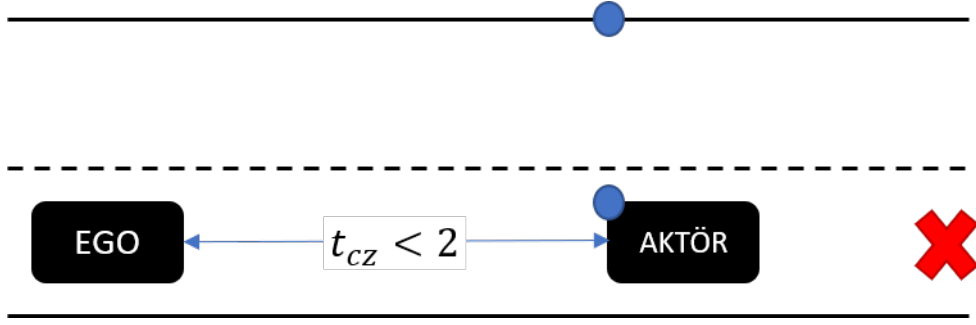
Boşluğu Takip Et metodu, bir adet kalibre edilebilir katsayısı (α) olması dolayısıyla kararlılığı yüksek bir metottur. Bu uygulamalar değişikçe kalibre edilmesini çok kolaylaştırır. Sollama, Boşluğu Takip Et metodunun engelden kaçması sırasında oluşan doğal bir manevra olmuştur. Adaptasyonu için, hedef noktası ego aktör arabanın belli bir mesafe önüne konumlandırılmıştır. Bu hedef noktası aktör aracın hareketine göre aktör araca olan göreceli mesafesi aynı olacak şekilde aynı hızla hareket etmektedir. Aktör araç, ego araç için engeldir ve hareketlidir. Şerit çizgileri de, ego araç için engeldir ve kaçması gerekir. Ego araç, bu hedef noktasına yol alırken, aktör araca da çarpılmaması gerekmektedir. BTE metodunun bu hareketi otonom sollamayı gerçekleştirmiş olur. BTE metodunun otonom sollama manevrasının akış şeması Şekil 6.4'te görüldüğü gibidir. Akış şemasında bulunan t_{cz} parametresi çarpışmaya olan zamanı temsil etmektedir ve denklem (6.3)'te görüldüğü gibi hesaplanmaktadır. Burada d_{ea} , ego araç ile aktör araç arasındaki mesafeyi temsil eder. Algoritma sürekli t_{cz} 'yi hesaplar ve 2 saniyeden daha küçük olup olmadığını kontrol eder. t_{cz} 'nin 2 saniyeden küçük olması eğer aynı koşullar devam ederse 2 saniye sonra çarpışma olacağı anlamını gelir. 2 değeri, deneysel sonuçlardan ve [27]'de verilen önerilerden hareketle seçilmiştir. t_{cz} 'nin 2 saniyeden küçük olduğu durumda, aktör araç, ego aracın önünde olup olmadığına bakılır. Eğer önünde ise, sollama manevrası başlar. Sollama manevrasının 'Şerit Değiştirme' fazı Şekil 6.5'te görüldüğü gibidir. Aktör aracın sol arka köşesi ve ona denk gelen şerit noktası iki engel olarak görülmektedir.

$$t_{cz} = \frac{d_{ea}}{|v_{ego} - v_{aktor}|} \quad (6.3)$$

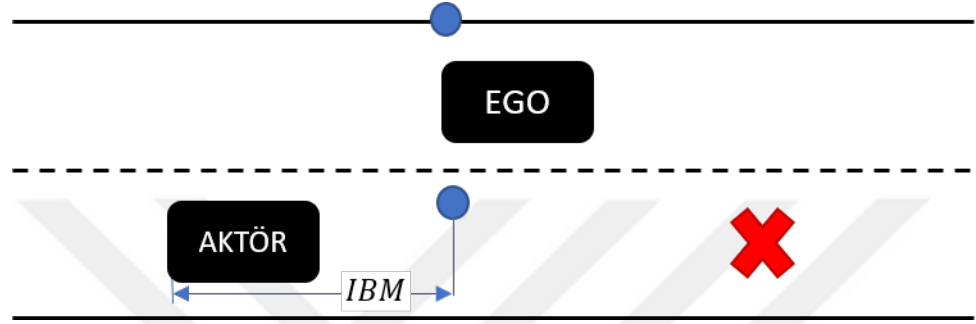
Ego araç aktör aracın x ekseninde arka tampon mesafesine geldiğinde, ego aracın sollama manevrasında 'Geçme' fazı başlar. Bu hareket şekil 6.7'da gösterilmektedir. Ego araç arka tamponun x mesafesinden başlar, aktör araca ileriye bakma mesafesi (*IBM*) kadar bir eşik değeri eklenip onu geçene kadar devam eder. *IBM*, ego araç aktör aracın önüne geçer geçmez birleşme fazına geçmesini önlemek için kullanılmıştır. Ego araç, x ekseninde aktör aracın *IBM* ötesine geldiğinde sollamanın 'birleşme' fazına geçer. Bu fazda, şekil 6.6'de görüldüğü üzere, ego araç eski şeridine dönerek



Şekil 6.4 : Boşluğu Takip Et metodunun sollama algoritması.

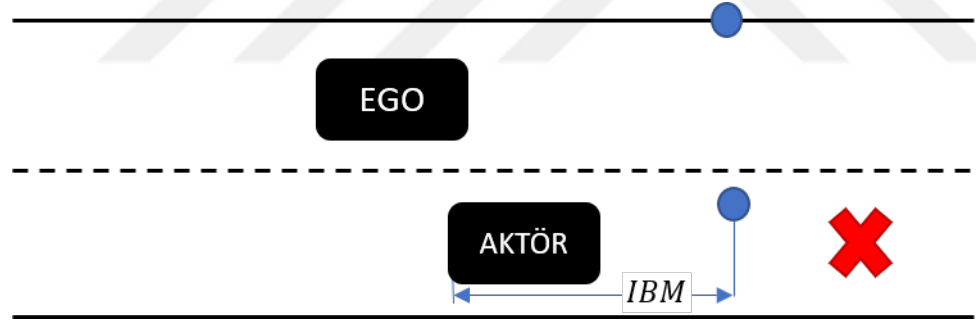


Şekil 6.5 : Şerit deęiřtirme fazı.



Şekil 6.6 : Birleřme fazı.

hedef noktasına ulařmaya çalıřır. Bu tezde Bořluęu Takip Et Metodu kullanılarak



Şekil 6.7 : Geçme fazı.

otonom sollama manevrasının tasarımı yapılmaktadır. Bu fazlar otonom sollama manevrasının tasarımı detaylı bir řekilde açıklanmaktadır ayrıca otonom sollama manevrası tasarımı sadece řerit deęiřtirme fazı kullanılarak řerit deęiřtirme manevrası tasarımı için de kullanılabilir. Bölüm 7’de bu tasarımı ve klasik yaklařımın sonuçları gösterilmektedir.

7. SONUÇLAR

Bu tezde, otonom sollama manevrasına yeni bir yaklaşım kazandırmak adına araştırmalar yapılmıştır. Bu araştırmalar sonucu iki yaklaşım ortaya çıkmıştır. Bu yaklaşımlardan bir tanesi bir engelden kaçma algoritması olan Boşluğu Takip Et metodunun hareket planlayıcısına ve kontrolcüsüne ihtiyaç duymadan her ikisini de verimli bir şekilde kendi başına yapabilmesidir. Otonom sollama manevrası Boşluğu Takip Et metodu için çift şeritli bir yolda önündeki engelden kaçmaya karar vermesiyle doğal olarak ortaya çıkmıştır. Bu yaklaşımın kararlılığını ve verimliliğini karşılaştırabilmek adına ikinci yaklaşım oluşturulmuş ve onunla karşılaştırılmıştır. İkinci yaklaşım literatürde genel olarak kullanılan yaklaşımdır. Dolayısıyla, hareket planlama için ayrı bir metot ve hatta otonom sollama manevrası için özel olarak geliştirilmiş X-sin fonksiyonu kullanılmıştır. Hareket kontrolü için, araştırmalar sonucu Stanley metodu kullanılmaya karar verilmiştir. X-sin fonksiyonu ve Stanley metodunu daha iyi irdelemek için, farklı a_{max} değerlerine göre de sonuçlar sunulmuştur. Her iki yaklaşım için, simulasyon ortamı olarak Matlab/Simulink seçilmiştir. Matlab/Simulink, dinamik bisiklet modeli olarak kullanılan araç modeli, yaklaşımların gerçekleşmesi, test ortamlarının oluşturulması ve iki yaklaşımın karşılaştırılması için kullanılmıştır. Test ortamı olarak otoban seçilmiştir ve otobanda koşulları [1]'nin verilerine göre hız sınırlarının $10 - 35m/s$ arasında olması gerekmektedir. Şerit genişliği ise [32] verilerine göre $3.5m$ olmalıdır. Simulasyonlar, başlangıç pozisyonları, başlangıç savrulma açıları, ego araç hızları, aktör araç hızları, hedefe olan mesafeleri, ortam ve sınırlamalar BTE ve XiS yaklaşımları için adil bir karşılaştırma olması adına aynı seçilerek yapılmıştır. Hedef noktası sollamanın birleşme fazının sonu yani aktör aracın boylamsal pozisyonundan M_d kadar ötesinde seçilmiştir. M_d de L_d ile aynı seçilmiştir ve denklem (6.2) ile hesaplanmıştır. Bu iki yaklaşım hızlar $20m/s$, $25m/s$ ve $30m/s$ olacak şekilde denenip sonuçları karşılaştırılıp, yorumlanmıştır. Karşılaştırma metrikleri olarak genel norm formülü kullanılmıştır. [18]'te kullanılan norm ifadesinin matematiksel gösterimi

denklem (7.1)'de ifade edilmiştir. Bu tezde p değeri 2 seçilerek, 2 normu üzerinden metrik tanımlaması gerçekleştirilmiştir. Bu karşılaştırma norm formülünü kullanarak emniyet metriği ve konfor metriği kullanılarak yapılmıştır. Ayrıca sollama manevrası sırasında toplam alınan yol karşılaştırma parametresi olarak eklenmiştir.

$$\|f\|_p = \left(\int |f(t)|^p dt \right)^{1/p} \quad (7.1)$$

Emniyet metriği için $f(t)$, (7.2)'deki gibi seçilmiştir ve ego araç ile aktör araç arasındaki mesafedir. Bu mesafenin yüksek olması istenmektedir. Çünkü bu mesafe ne kadar yüksek olursa, aslında ego araç aktör araca o kadar uzak yani emniyetli demektir.

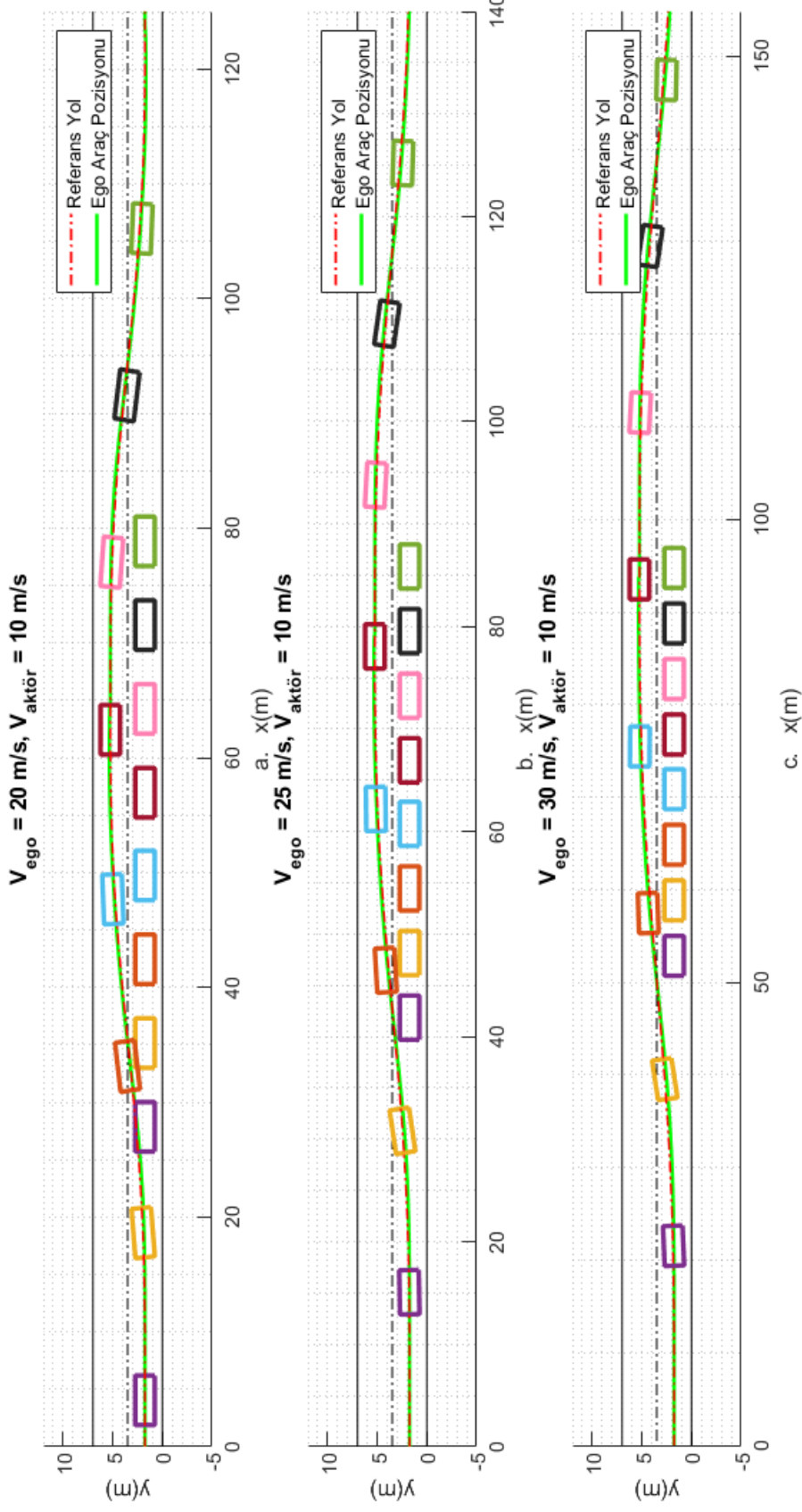
$$f(t) = \sqrt{(x_{ego} - x_{aktor})^2 + (y_{ego} - y_{aktor})^2} \quad (7.2)$$

Konfor metriği için aracın savrulmasının az olması beklenmektedir. Bu yüzden $f(t)$, ψ olarak kullanılmıştır. Konfor metriğinde değerin yüksek çıkması, aracın daha çok savrulduğu anlamına gelir. Dolayısıyla, konfor metriğinin değerinin daha az çıkması, yolcular için daha konforlu bir sollamanın gerçekleştiğini gösterir. Diğer bir karşılaştırma metriği olarak, alınan yol seçilmiştir. Alınan yol, en son yapılan hareketin toplam alınan mesafesi hesaplanmıştır. Burada da alınan yolun az olması asıl istenendir. Fakat eğer daha güvenli bir sollama olacaksa daha uzun bir yol gereksinimlere göre tercih edilebilir.

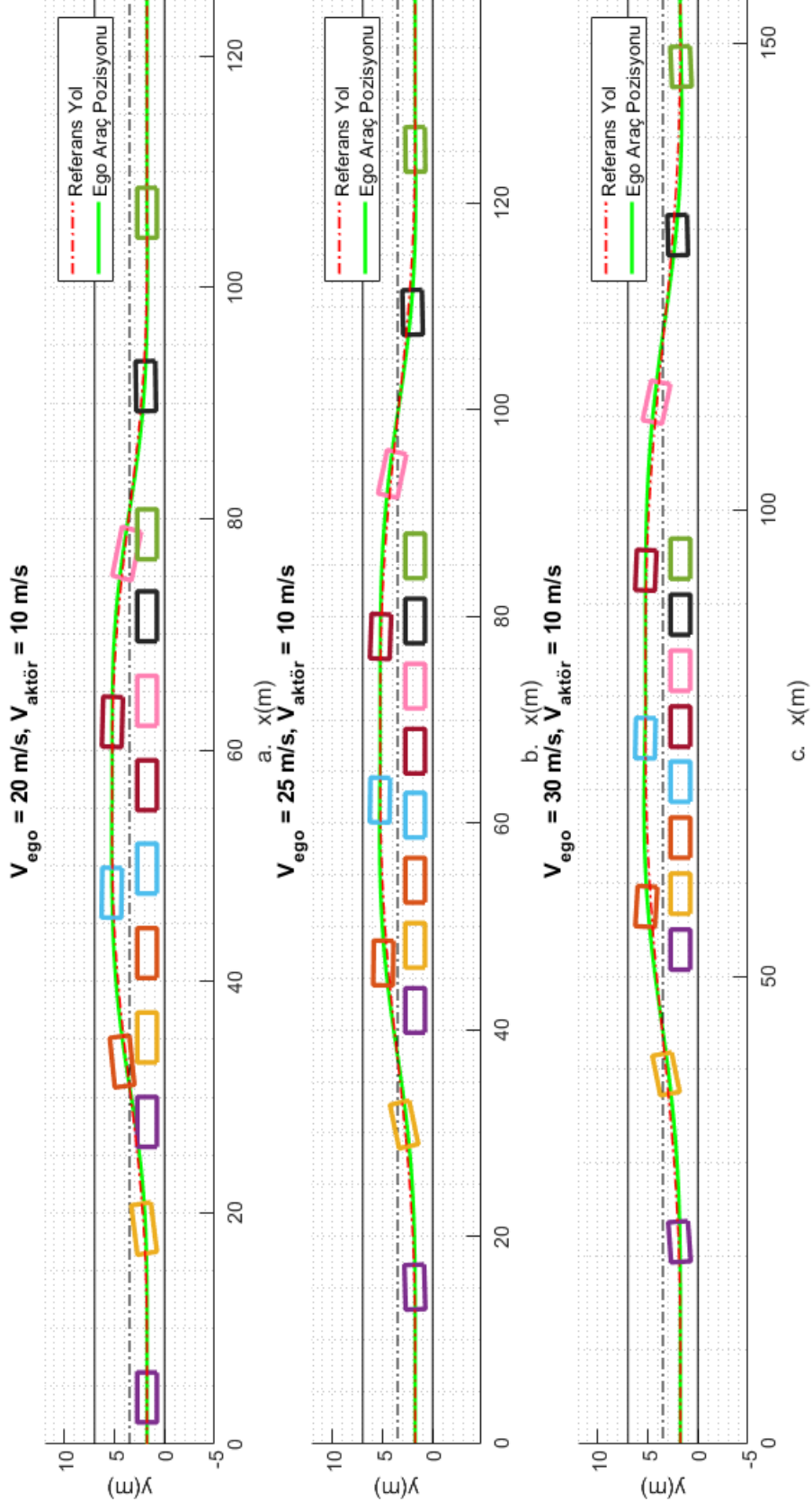
Çizelge 7.1'e göre, anlaşılıyor ki a_{max} 'tan bağımsız olarak, ego aracın hızının artması her durumda konforu azaltan bir etkidir. Diğer taraftan, v_x 'in artmasıyla birlikte, denklem (6.2)'de göre L_d 'nin de artması beklendiğinden, sollamanın daha önce başlaması ile birlikte emniyet metriği de artar. Ayrıca toplam alınan mesafe de artar. Aynı hızlardaki karşılaştırmaya bakılacak olursa, a_{max} 'ın artması ile birlikte L_d 'nin değeri, denklem (6.2)'ye göre düşer. L_d 'nin değerinin azalması ile birlikte yolcular için emniyet ve konfor azalır. Fakat, toplam alınan mesafe de azalır. Yüksek L_d değerleri şekil 7.1'de ve şekil 7.2'de de görüldüğü üzere alınan mesafenin uzamasına sebep olur. Yüksek v_x değerleri de yol takibinde aşımara neden olabilir.

Çizelge 7.1 : XiS'nin farklı a_{max} değerlerine göre karşılaştırma çizelgesi.

Metrikler	$a_{max} = 2.5m/s^2$			$a_{max} = 3m/s^2$			$a_{max} = 3.5m/s^2$		
	$v_{ego} = 20m/s$	$v_{ego} = 25m/s$	$v_{ego} = 30m/s$	$v_{ego} = 20m/s$	$v_{ego} = 25m/s$	$v_{ego} = 30m/s$	$v_{ego} = 20m/s$	$v_{ego} = 25m/s$	$v_{ego} = 30m/s$
Konfor Metriği	297.29	425.30	552.03	333.24	484.05	648.70	380.93	571.75	783.83
Emniyet Metriği	21.44	27.89	34.18	19.37	25.49	31.60	17.82	23.86	29.32
Toplam Yol [m]	145.82	163.98	185.04	135.53	153.76	174.87	127.64	146.55	165.63



Şekil 7.1 : $a_{max} = 2.5 \text{ m/s}^2$ iken XiS; a. $v_{ego} = 20 \text{ m/s}$, b. $v_{ego} = 25 \text{ m/s}$, c. $v_{ego} = 30 \text{ m/s}$.



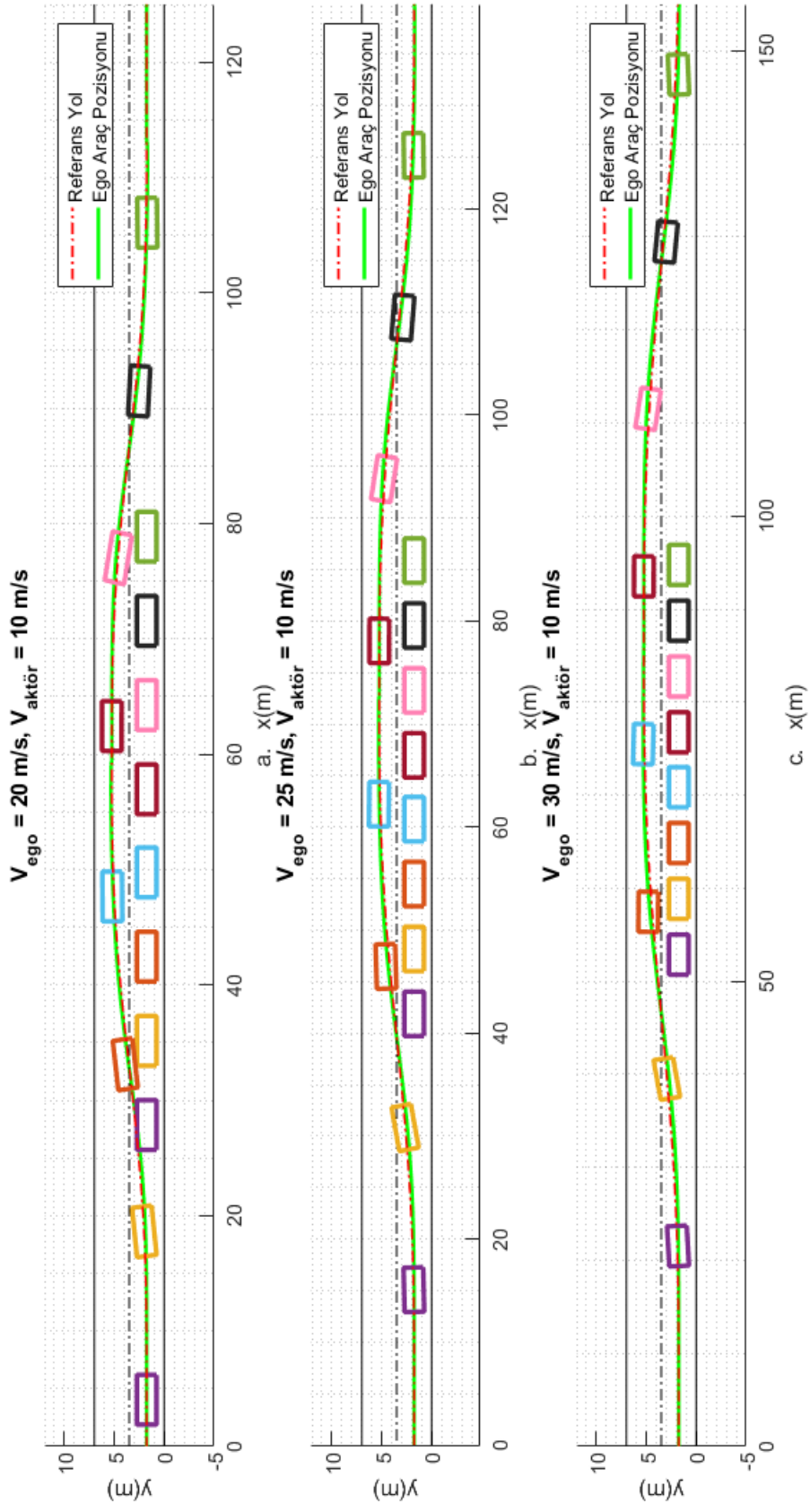
Şekil 7.2 : $a_{max} = 3.5 \text{ m/s}^2$ iken XiS; a. $v_{ego} = 20 \text{ m/s}$, b. $v_{ego} = 25 \text{ m/s}$, c. $v_{ego} = 30 \text{ m/s}$.

Çizelge 7.2 : BTE ve XiS karşılaştırma çizelgesi.

Hız	Yaklaşım	Konfor Metriği	Emniyet Metriği	Toplam Yol [m]
20 m/s	BTE	216.45	21.21	146.84
	XiS	333.24	19.37	135.54
25 m/s	BTE	282.34	30.65	161.30
	XiS	484.05	25.50	153.77
30 m/s	BTE	371.90	40.16	184.27
	XiS	648.70	31.60	174.88

Çizelge 7.2, BTE ile XiS aynı metrikler ile detaylı karşılaştırmasını içermektedir. İki yaklaşımda da, başlangıç pozisyonları aynıdır. Hedef noktası olarak da aynı nokta verilmiştir. Karşılaştırmanın doğruluğu açısından, karşılaştırmalar aynı boylamsal hızlara göre yapılmıştır. Aynı hızlar ele alındığında BTE'nin, XiS'ten daha konforlu olduğunu konfor metriğini daha düşük bir değeri olduğuna bakarak söylemek mümkündür. Emniyet metriğinin değeri BTE'de daha yüksektir bu da BTE'nin sollamayı daha güvenli bir şekilde gerçekleştirdiği anlamına gelir. Daha güvenli olması toplam alınan yolun da daha yüksek olmasına sebep olur. Buradaki daha uzun yol alma güvenlik ve konfor metriklerindeki avantajlar düşünüldüğünde göz ardı edilebilecek seviyededir. Şekil 7.3'te görüldüğü üzere BTE'deki toplam alınan yol, şekil 7.4'te görülen XiS'deki toplam alınan yoldan daha yüksektir. Toplam alınan yolun BTE'de daha yüksek olmasının sebebi, BTE'nin denklem (3.11)'de bulunan d_{min} parametresidir. Bu aktör araca yaklaştıkça ego araca ondan kaçması için oluşturulan direksiyon açısından kaynaklanır. Şu da belirtilmelidir ki BTE'de XiS gibi hareket planlayıcı ve hareket kontrolörüne ayrı ayrı ihtiyaç duymadan kendiliğinden planlamayı ve kontrolü yapar. Bu tezde yapılan çalışmaların doğruluğunu arttırmak ve daha genel bir çıkarım yapmak adına, Monte Carlo simülasyonları iki yaklaşım içinde gerçekleştirilmiştir. Bu simülasyonlar, aynı ortam için yapılarak BTE'nin ortalama başarı yüzdesi elde edilmiştir.

Bu tezde 1000 adet Monte Carlo simülasyonu yapılmıştır. Ego aracın hızı $10 - 20m/s$ arasında rastgele seçilmiştir. Aktör aracın hızı ise $8 - 12m/s$ hızları arasında rastgele atanmıştır. Her Monte Carlo simülasyonu BTE ve XiS'e aynı rastgele oluşturulmuş ego araç hızı ve aynı rastgele oluşturulmuş aktör araç hızı için yapılmıştır ve aynı şekilde bu rastgele ego ve aktör araç hızları için d_{ea} , L_d ve M_d hesaplanır ve simülasyon parametresi olarak eklenir. Çizelge 7.3'te 1000 adet Monte Carlo simülasyonlarının sonuçları ve başarı oranı verilmiştir. Sonuçlara göre ortalama konfor metriği BTE'de %41 daha iyidir. Emniyet metriğinin ortalaması da BTE'de XiS'e göre %13 daha yüksektir. Öte yandan toplam alınan yolun 1000 adet Monte Carlo simülasyonlarındaki ortalaması gösteriyor ki, BTE XiS'den %2.41 daha uzundur. Bu da daha önceden de belirtildiği gibi, emniyet ve konfordaki avantajları toplam alınan mesafedeki bu küçük farkı doğurmaktadır.



Şekil 7.4 : XiS'nin karşılaştırma sonuçları; a. $v_{ego} = 20m/s$, b. $v_{ego} = 25m/s$, c. $v_{ego} = 30m/s$.

Çizelge 7.3 : 1000 adet Monte Carlo simülasyon sonuçları.

Metot / Metrikler	Konfor Metriği	Emniyet Metriği	Toplam yol [m]
BTE	290.15	31.21	164.02
XiS	493.96	27.01	160.15
Başarı Oranı	41%	13%	-2.41%

8. ÖNERİLER VE GELECEK ÇALIŞMA

Bu tezde, BTE metodu otonom sollama manevrasında denenmiştir. Bu deneme ile XiS otonom sollama manevrası ile karşılaştırılmıştır. Simülasyonlarda dinamik bisiklet modeli kullanılmıştır. İki yaklaşım arasındaki ana fark BTE tek algoritmayla hem hareket planlamasını hem de harekte kontrolünü yapmaktayken, XiS için hareket planlama ve hareket kontrol algoritmaları ayrı ayrı gerçekleştirilmiştir. Monte Carlo simülasyonlarından da desteklendiği üzere BTE'nin XiS'e göre otonom sollama problemini çözme başarısı emniyette %13, konforda %41 daha üstündür. Öte yandan toplam alınan yol ise %2.41 daha uzundur. Bu sonuçlar BTE'nin engelden kaçma algoritması olmasının ötesinde aynı zamanda otonom sollama problemin çözümü için de çok uygun olduğunu göstermektedir. BTE bir engelden kaçma algoritması olmasına rağmen, otonom sollama manevrası için oluşturulmuş bir hareket planlama algoritmasından ve hareket kontrolü için oluşturulmuş bir metottan çok daha iyi sonuçlar vermiştir. Bundan sonraki çalışmalar için denenebilmesi mümkün olan başka çalışmalar da vardır. Otobandaki otonom sollama manevrası için seçilen *IBM* parametresinin değeri başka uygulamalar için değişmesi gerekebilir. BTE bir engelden kaçma algoritması olduğundan dolayı, aslında dinamik ortamlar için de çok uygundur. Bu sebeple şehir içi trafiğindeki hareketli objelere karşı kararlı bir hareket sergileyecektir. Aynı zamanda, şehir içi trafiğindeki kavşaklar, bağlantı yolları, döneller için de hareket planlaması ve kontrolü için BTE kullanılabilir.



KAYNAKÇA

- [1] **Url-1.** <<http://www.kgm.gov.tr/Sayfalar/KGM/SiteTr/Trafik/HizSinirlari.aspx>>, erişim tarihi 12.04.2019.
- [2] **Url-2.** <<http://www.kgm.gov.tr/SiteCollectionDocuments/KGMdocuments/Trafik/Trafik2017.pdf>>, erişim tarihi 12.04.2019.
- [3] **Url-3.** <<https://www.coursera.org/learn/robotics-motion-planning>>, erişim tarihi 20.04.2019.
- [4] **Toroyan, T.** (2018), Global Status Report on Road Safety.
- [5] **Schumacher, R. W., O., D., R., Wragg, R. ve Landau, F.H.**, (1995), Collision Warning System Technology.
- [6] **International, S.**, (2014), Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems.
- [7] **Shaout, A., Colella, D. ve Awad, S.** (2011). Advanced Driver Assistance Systems - Past, present and future, *Proc. 7th Int. Computer Engineering Conf.*
- [8] **Url-4.** <<http://www.tuik.gov.tr/PreHaberBultenleri.do?id=27668>>, erişim tarihi 25.03.2019.
- [9] **C Latombe, J.** (1991). *Robot Motion Planning*.
- [10] **Snider, J.M.** (2009). Automatic Steering Methods for Autonomous Automobile Path Tracking.
- [11] **Cibooglu, M., Karapinar, U. ve Soylemez, M.T.** (2017). Hybrid controller approach for an autonomous ground vehicle path tracking problem, *2017 25th Mediterranean Conference on Control and Automation (MED)*, s.583–588.
- [12] **Rajamani, R.** (2011). *Vehicle Dynamics and Control*, Mechanical Engineering Series, Springer US.
- [13] **Lumelsky, V. ve Stepanov, A.** (1986). Dynamic path planning for a mobile automaton with limited information on the environment, *IEEE Transactions on Automatic Control*, 31(11), 1058–1063.
- [14] **Yufka, A. ve Parlaktuna, O.** (2009). Performance Comparison of the BUG's Algorithms for Mobile Robots.
- [15] **Zohaib, M., Pasha, S.M., Javaid, N. ve Iqbal, J.** (2013). Intelligent Bug Algorithm (IBA): A Novel Strategy to Navigate Mobile Robots Autonomously, *CoRR*, *abs/1312.4552*.

- [16] **Zhu, Q., Yan, Y. ve Xing, Z.** (2006). Robot Path Planning Based on Artificial Potential Field Approach with Simulated Annealing, *Sixth International Conference on Intelligent Systems Design and Applications*, 2, 622–627.
- [17] **Sezer, V. ve Gokasan, M.** (2012). A Novel Obstacle Avoidance Algorithm: "Follow the Gap Method", *Robot. Auton. Syst.*, 60(9), 1123–1134.
- [18] **Demir, M. ve Sezer, V.** (2017). Improved Follow the Gap Method for obstacle avoidance, *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, s.1435–1440.
- [19] **Özdemir, A. ve Sezer, V.** (2018). Follow the Gap with Dynamic Window Approach, *International Journal of Semantic Computing*, 12, 43–57.
- [20] **Varaiya, P.** (1993). Smart cars on smart roads: problems of control, *IEEE Transactions on Automatic Control*, 38(2), 195–207.
- [21] **Wybe. Dijkstra, E.** (1959). A Note on Two Problems in Connexion With Graphs, *Numerische Mathematik*, 1, 269–271.
- [22] **Likhachev, M. ve Ferguson, D.** (2009). Planning Long Dynamically Feasible Maneuvers for Autonomous Vehicles, *The International Journal of Robotics Research*, 28(8), 933–945.
- [23] **Stentz, A.** (1994). Optimal and efficient path planning for partially-known environments, *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, s.3310–3317 vol.4.
- [24] **Kavraki, L., Svestka, P., claude Latombe, J. ve Overmars, M.** (1996). Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces, *IEEE International Conference on Robotics and Automation*, s.566–580.
- [25] **Lavalle, S.M.** (1998). Rapidly-Exploring Random Trees: A New Tool for Path Planning, **Teknik Rapor**.
- [26] **Werling, M., Ziegler, J., Kammel, S. ve Thrun, S.** (2010). Optimal trajectory generation for dynamic street scenarios in a Frenét Frame, *2010 IEEE International Conference on Robotics and Automation*, 987–993.
- [27] **Zhang, S., Liu, W., Li, B., Niu, R. ve Liang, H.** (2014). Trajectory planning of overtaking for intelligent vehicle based on X-Sin function, *2014 IEEE International Conference on Mechatronics and Automation*, s.618–622.
- [28] **Coulter, R.C.** (1992). Implementation of the Pure Pursuit Path Tracking Algorithm, **Teknik RaporCMU-RI-TR-92-01**, Carnegie Mellon University, Pittsburgh, PA.
- [29] **Hoffmann, G.M., Tomlin, C.J., Montemerlo, M. ve Thrun, S.** (2007). Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing, *2007 American Control Conference*, s.2296–2301.

- [30] **G. M. Fitch, S. E. Lee, S.K.J.H.J.S. ve Dingus, T.** (2009). *Analysis of Lane-Change Crashes and Near-Crashes*, National Highway Traffic Safety Administration.
- [31] **Sezer, V.** (2018). Intelligent decision making for overtaking maneuver using mixed observable Markov decision process, *J. Intellig. Transport. Systems*, 22(3), 201–217.
- [32] **Url-5.** <https://web.archive.org/web/20120311004223/http://www.eurotestmobility.net/eurotest.php?itemno=224&lang=EN>, Rovi Corporation, erişim tarihi 12.04.2019.





EKLER

EK A.1 : A* Algoritması

EK A.2 : Sollama Algoritması





EK A.1

Result: Başlangıç Düzümü ile Hedef Düzüm arasında yol

```
while Açık Listedeki tüm elemanlara bakılana kadar do
  Açık Listeden en iyi düğüm seçilir. ->  $n_{eniye}$ 
  if  $f(n_{eniye}) \leq f(n)$  then
    |  $n_{eniye}$  Açık Listeden Kapalı Listeye taşınır.
  else
    | if  $n_{eniye} == n_{hedef}$  then
      | Çıkış
    | else
      | if  $n_{eniye}$  açık listenin elemanı değilse then
        | Açık Listeye eklenir.
      | else
        |  $n$  düğümünün ebeveyn  $n_{eniye}$  düğümü olarak değiştirilir.
      | end
    | end
  | end
end
```

Algoritma 1: A* Algoritması



EK A.2

Result: Sollamanın Tamamlanması

$x_{ego_0}, y_{ego_0}, x_{aktor_0}, y_{aktor_0}, v_{ego}$ ve v_{aktor} tanımlanması

while *Hedef noktası kontrolü* **do**

```
    if  $t_{cz} < 2$  then
        if  $x_{aktor} > x_{ego}$  then
            | Şerit Değişirme
        else
            if  $x_{aktor} + IBM > x_{ego}$  &&  $x_{ego} > x_{aktor}$  then
                | Geçme
            else
                if  $x_{ego} > x_{aktor} + IBM$  then
                    | Birleştirme
                else
                    | Geçme
                end
            end
        end
    else
        | Şerit Takibi
    end
end
```

Algoritma 2: Sollama Algoritması



ÖZGEÇMİŞ

Ad Soyad: Münire Damla Demir

Doğum Tarihi ve Yeri: 19.02.1993, Kocaeli

E-Posta: damlademiir@gmail.com



ÖĞRENİM DURUMU: Üniversite

- **Lisans:** 2016, Yıldız Teknik Üniversitesi, Makine Fakültesi, Mekanik Mühendisliği
- **Y. Lisans:** 2019, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Mekanik Mühendisliği

MESLEKİ DENEYİMLER VE ÖDÜLLER:

- 2015-2016 yılları arasında Almanya’da Bosch Rexroth Şirketi Yazılım Departmanında kriptografi üzerine çalıştı.
- 2016 yılında İstanbul Teknik Üniversitesi Mekanik Mühendisliği Yüksek Lisans Programına kayıt oldu.
- 2016 yılında AVL Türkiye’de sistem mühendisi olarak çalışmaya başladı ve çeşitli uluslararası otomotiv projelerinde çalıştı.