

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SÜRÜCÜ POSTÜRÜNÜN TANIMLANMASI VE MAKİNE
ÖĞRENMESİ TABANLI TAHMİNİ

Ceyhun IBOLAR

YÜKSEK LİSANS TEZİ
Mekatronik Mühendisliği Anabilim Dalı
Mekatronik Mühendisliği Programı

Danışman
Doç. Dr. Erhan AKDOĞAN

Eş-Danışman
Dr. A. Emre ÇETİN

Temmuz, 2020

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**SÜRÜCÜ POSTÜRÜNÜN TANIMLANMASI VE MAKİNE ÖĞRENMESİ
TABANLI TAHMİNİ**

Ceyhun IBOLAR tarafından hazırlanan tez çalışması 20.07.2020 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Anabilim Dalı Mekatronik Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Doç. Dr. Erhan AKDOĞAN
Yıldız Teknik Üniversitesi
Danışman

Dr. A. Emre ÇETİN
Altınay Teknolojileri
Eş-Danışman

Jüri Üyeleri

Doç. Dr. Erhan AKDOĞAN, Danışman
Yıldız Teknik Üniversitesi

Prof. Dr. Vasfi Emre ÖMÜRLÜ, Üye
Yıldız Teknik Üniversitesi

Doç. Dr. Alihsan KOCA, Üye
Fatih Sultan Mehmet Vakıf Üniversitesi

Danışmanım Doç. Dr. Erhan AKDOĞAN sorumluluğunda tarafımca hazırlanan Sürücü Postürünün Tanımlanması ve Makine Öğrenmesi Tabanlı Tahmini başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Ceyhun IBOLAR

İmza

*Değerli aileme
ve hayat arkadaşına*



TEŞEKKÜR

Bu çalışmanın yürütülmesi sırasında desteğini esirgemeyen danışmanım Doç. Dr. Erhan Akdoğan'a, çalışmalarım sırasında destek oldukları için çok yakın arkadaşlarım Doruk ve Seher'e, sistemin mekanik tasarımının üretiminde destek veren Altınay Teknolojileri'ne, özellikle eş danışmanım Dr. A. Emre Çetin'e ve çalışmam sırasında küçük veya büyük yardımını esirgemeyen herkese teşekkür ederim.

Ceyhun IBOLAR

İÇİNDEKİLER

SİMGE LİSTESİ	vii
KISALTMA LİSTESİ	x
ŞEKİL LİSTESİ	xi
TABLO LİSTESİ	xiii
ÖZET	xiv
ABSTRACT	xvi
1 GİRİŞ	1
1.1 Literatür Özeti	2
1.2 Tezin Amacı	8
1.3 Orjinal Katkı	8
1.4 Yöntem	10
1.4.1 Tensör Matrisi	11
1.4.2 Makine Öğrenmesi	13
1.5 İçerik	14
2 TEZ ÇALIŞMALARI	15
2.1 Sürücü Postürü	15
2.2 Sistemin Modellenmesi	15
2.2.1 Direksiyon Sistemi	15
2.2.2 Sürücü Sistemi Analiz Algoritmaları	18
2.3 Deney Düzenegi	23
2.4 Sistem Arayüzü ve Haberleşme Protokolü	24
2.4.1 Sistem Arayüzü	24
2.4.2 Haberleşme Protokolü	25
3 SİMÜLASYON VE DENEYSEL ÇALIŞMALAR	26
3.1 MATLAB ile Makine Öğrenmesi Algoritması	26
3.1.1 Veri Toplama	26

3.1.2 Verilerin Önişlenmesi	27
3.1.3 Öngörülü Modelin Geliştirilmesi	28
3.1.4 Analitik Çözümün Sistemlere Entegre Edilmesi	32
3.2 Sistemin Gerçek Zamanlı Çalışması	32
3.3 Sürücü Postür Algoritması Analiz Sonuçları	34
3.4 Makine Öğrenmesi Algoritması Analiz Sonuçları	52
4 SONUÇ VE ÖNERİLER	55
KAYNAKÇA	57
TEZDEN ÜRETİLMİŞ YAYINLAR	60



SİMGE LİSTESİ

e_1	Birinci Koordinat Sistemi
X_l	Birinci Koordinat Sisteminde X Eksen
Y_1	Birinci Koordinat Sisteminde Y Eksen
Z_1	Birinci Koordinat Sisteminde Z Eksen
θ_{sw}	Direksiyon Açısal Konumu
r_{sw}	Direksiyon Yarıçapı
θ_{hand}	Elin Açısal Konumu
Φ_{1rh}	Hesaplanan Sağ El Kuvvet/Tork Değer Matrisi
Φ_{1lh}	Hesaplanan Sol El Kuvvet/Tork Değer Matrisi
e_0	Orjin Koordinat Sistemi
X_0	Orjin Koordinat Sisteminde X Eksen
Y_0	Orjin Koordinat Sisteminde Y Eksen
Z_0	Orjin Koordinat Sisteminde Z Eksen
Q	Ortogonal Matris
Φ_1	Ölçülen Kuvvet/Tork Değer Matrisi
θ_{rh}	Sağ El Açısal Konumu
e_{rh}	Sağ El Koordinat Sistemi
X_{rh}	Sağ El Koordinat Sisteminde X Eksen
Y_{rh}	Sağ El Koordinat Sisteminde Y Eksen
Z_{rh}	Sağ El Koordinat Sisteminde Z Eksen
f_{xrh}	Sağ El Koordinat Sistemindeki X Ekseninde Hesaplanan Kuvvet
$f_{y rh}$	Sağ El Koordinat Sistemindeki Y Ekseninde Hesaplanan Kuvvet
$f_{z rh}$	Sağ El Koordinat Sistemindeki Z Ekseninde Hesaplanan Kuvvet

m_{xrh}	Sağ El Koordinat Sistemindeki X Ekseninde Hesaplanan Tork
m_{yrh}	Sağ El Koordinat Sistemindeki Y Ekseninde Hesaplanan Tork
m_{zrh}	Sağ El Koordinat Sistemindeki Z Ekseninde Hesaplanan Tork
θ_{lh}	Sol El Açısai Konumu
R_{1lh}	Sol El Dönme Matrisi
H_{1lh}	Sol El Dönüşüm Matrisi
e_{lh}	Sol El Koordinat Sistemi
X_{lh}	Sol El Koordinat Sisteminde X Eksenii
f_{xlh}	Sol El Koordinat Sistemindeki X Ekseninde Hesaplanan Kuvvet
m_{xlh}	Sol El Koordinat Sistemindeki X Ekseninde Hesaplanan Tork
Y_{lh}	Sol El Koordinat Sisteminde Y Eksenii
f_{ylh}	Sol El Koordinat Sistemindeki Y Ekseninde Hesaplanan Kuvvet
m_{ylh}	Sol El Koordinat Sistemindeki Y Ekseninde Hesaplanan Tork
Z_{lh}	Sol El Koordinat Sisteminde Z Eksenii
f_{zlh}	Sol El Koordinat Sistemindeki Z Ekseninde Hesaplanan Kuvvet
m_{zlh}	Sol El Koordinat Sistemindeki Z Ekseninde Hesaplanan Tork
T_{1lh}	Sol El Öteleme Matrisi
T	Tensör Matrisi
x_1	Vektörel Yön
x_2	Vektörel Yön
x_3	Vektörel Yön
x'_1	Vektörel Yön
x'_2	Vektörel Yön
x'_3	Vektörel Yön
n_1	x_1 Yönünde Birim Vektörü
n_2	x_2 Yönünde Birim Vektörü
n_3	x_3 Yönünde Birim Vektörü
n'_1	x'_1 Yönünde Birim Vektörü
n'_2	x'_2 Yönünde Birim Vektörü

n'_3	x'_3 Yönünde Birim Vektörü
f_x	X Ekseninde Ölçülen Kuvvet
m_x	X Ekseninde Ölçülen Tork
f_y	Y Ekseninde Ölçülen Kuvvet
m_y	Y Ekseninde Ölçülen Tork
t	Zaman
f_z	Z Ekseninde Ölçülen Kuvvet
m_z	Z Ekseninde Ölçülen Tork



KISALTMA LİSTESİ

Adaboost	Adaptive Boosting
AUC	Area Under The Curve
CNN	Convolutional Neural Network
DAQ	Data Acquisition
DC	Direct Current
EEG	Electroencephalogram
EMG	Electromyography
FFNN	Feedforward Neural Network
F/T	Force/Torque
IP	Internet Protocol
IR	Infrared
KNN	K-Nearest Neighbour
LSTM	Long Short Term Memory
MÖ	Makine Öğrenmesi
N	Newton
NI	National Instruments
ROC	Receiver Operating Characteristic
RPN	Region Proposal Network0
SVM	Support Vector Machine
TÜİK	Türkiye İstatistik Kurumu
UDP	User Datagram Protocol
V	Voltage
YSA	Yapay Sinir Ağları

ŞEKİL LİSTESİ

Şekil 1.1	Literatürde göz tanıma algoritması	2
Şekil 1.2	Literatürde göz tanıma resmi	3
Şekil 1.3	Literatürde sürücü dikkatsizlik senaryoları	3
Şekil 1.4	Literatürde telefon kullanımı	5
Şekil 1.5	Literatürde sürücü dikkatsizlik durumları	5
Şekil 1.6	Literatürde el konum tespiti	5
Şekil 1.7	Literatürde kinect sensörü	6
Şekil 1.8	Literatürde kinect sensörü 2	7
Şekil 1.9	Literatürde biyolojik ölçüm	8
Şekil 2.1	Direksiyon sistemi	16
Şekil 2.2	ATI S-330-30, kuvvet/tork sensörü	16
Şekil 2.3	NI-USB 6221, veri toplama sistemi	17
Şekil 2.4	Kübler, artımsal enkoder	17
Şekil 2.5	Direksiyon sisteminin serbest cisim diyagramı	18
Şekil 2.6	Direksiyon sistemi	23
Şekil 2.7	Deney düzeneği	23
Şekil 2.8	Unity arayüzü	24
Şekil 2.9	Simulink modelinde UDP gönderici bloğu	25
Şekil 3.1	Classification learner SVM doğruluk oranı	29
Şekil 3.2	Destek vektör makinesi 2D hiper düzlem örneği	30
Şekil 3.3	Destek vektör makinesi 3D örneği	30
Şekil 3.4	Destek vektör makinesi 3D hiper düzlem örneği	30
Şekil 3.5	Makine öğrenmesi algoritması hata matrisi	31
Şekil 3.6	ROC eğrisi	32
Şekil 3.7	Sistemin simulink modeli	33
Şekil 3.8	Senaryo 1, sağ el ile sürüş	34
Şekil 3.9	Senaryo 1, sağ el parametreleri grafikleri	35
Şekil 3.10	Senaryo 1, direksiyon açısı ve sağ el konum grafikleri	35
Şekil 3.11	Senaryo 1, sağ el konum hata grafiği	36
Şekil 3.12	Direksiyonu tutma aralığı	36
Şekil 3.13	Senaryo 2, sağ el ile sürüş	37

Şekil 3.14 Senaryo 2, sağ el parametreleri grafikleri	37
Şekil 3.15 Senaryo 2, direksiyon açısı ve sağ el konum grafikleri	38
Şekil 3.16 Senaryo 2, sağ el konum hata grafiği	38
Şekil 3.17 Senaryo 3, sağ el ile sürüş	39
Şekil 3.18 Senaryo 3, sağ el parametreleri grafikleri	39
Şekil 3.19 Senaryo 3, direksiyon açısı ve sağ el konum grafikleri	40
Şekil 3.20 Senaryo 3, sağ el konum hata grafiği	40
Şekil 3.21 Senaryo 4, sağ el ile sürüş	40
Şekil 3.22 Senaryo 4, sağ el parametreleri grafikleri	41
Şekil 3.23 Senaryo 4, direksiyon açısı ve sağ el konum grafikleri	41
Şekil 3.24 Senaryo 4, sağ el konum hata grafiği	42
Şekil 3.25 Senaryo 5, sol el ile sürüş	42
Şekil 3.26 Senaryo 5, sol el parametreleri grafikleri	43
Şekil 3.27 Senaryo 5, direksiyon açısı ve sol el konum grafikleri	43
Şekil 3.28 Senaryo 5, sol el konum hata grafiği	43
Şekil 3.29 Senaryo 6, sol el ile sürüş	44
Şekil 3.30 Senaryo 6, sol el parametreleri grafikleri	44
Şekil 3.31 Senaryo 6, direksiyon açısı ve sol el konum grafikleri	45
Şekil 3.32 Senaryo 6, sol el konum hata grafiği	45
Şekil 3.33 Senaryo 7, sol el ile sürüş	46
Şekil 3.34 Senaryo 7, sol el parametreleri grafikleri	46
Şekil 3.35 Senaryo 7, direksiyon açısı ve sol el konum grafikleri	47
Şekil 3.36 Senaryo 7, sol el konum hata grafiği	47
Şekil 3.37 Senaryo 8, sol el ile sürüş	48
Şekil 3.38 Senaryo 8, sol el parametreleri grafikleri	48
Şekil 3.39 Senaryo 8, direksiyon açısı ve sol el konum grafikleri	49
Şekil 3.40 Senaryo 8, sol el konum hata grafiği	49
Şekil 3.41 Senaryo 9, sağ el parametreleri grafikleri	50
Şekil 3.42 Senaryo 9, direksiyon açısı ve sağ el konum grafikleri	50
Şekil 3.43 Senaryo 9, sağ el konum hata grafiği	51
Şekil 3.44 Senaryo 10, sol el parametreleri grafikleri	51
Şekil 3.45 Senaryo 10, direksiyon açısı ve sol el konum grafikleri	52
Şekil 3.46 Senaryo 10, sol el konum hata grafiği	52
Şekil 3.47 Makine öğrenmesi birinci senaryo grafikleri	53
Şekil 3.48 Makine öğrenmesi ikinci senaryo grafikleri	54

TABLO LİSTESİ

Tablo 1.1	Literatürdeki sürücü postür analiz yöntemleri	10
Tablo 3.1	Gelecek tahmin tablosu	28
Tablo 3.2	Classification learner performans kıyaslaması	33
Tablo 3.3	Sürücü postür algoritması için sürüş senaryoları	34



Sürücü Postürünün Tanımlanması ve Makine Öğrenmesi Tabanlı Tahmini

Ceyhun IBOLAR

Mekatronik Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Doç. Dr. Erhan AKDOĞAN

Eş-Danışman: Dr. A. Emre ÇETİN

Trafik kazalarının çoğu sürücünün sürüş esnasında başka bir iş yapmasından veya sürücünün o anki fiziksel ve ruhî durumundan (uykulu, yorgun, kızgın, vb.) kaynaklanmaktadır. Literatürde sürücünün sürüş dikkatini takip eden ve buna bağlı olarak sürücüye geri bildirim yapan sistemlere dair birçok çalışma bulunmaktadır. Günümüzde, yeni nesil araçlarda kullanılan akıllı sürüş sistemlerinin çoğunda görüntü işleme yöntemi kullanılmaktadır. Buna bağlı olarak, sürücünün fiziksel davranışlarını tespit ederek yorumlayan algoritmaların geliştirilmesine ilişkin çalışmaların ivmelenerek sürüldüğü görülmektedir. Zamanla, sürücü davranışlarını yorumlayan bu algoritmaların geliştirilmesine olan ilgi sensörlerin ölçme hassasiyetlerinin artması ve maliyetlerinin azalması ile artmıştır. Bu algoritmalarda kullanılan önemli sensörlerden biri de kuvvet/tork sensörüdür. Kuvvet/tork sensörü ile direksiyona uygulanan kuvvetler incelenerek sürücünün davranışı tahmin edilebilir ve buna bağlı olarak çıkarımlar yapılabilir. Böylece el pozisyonu ve sürücünün uyguladığı kuvvet/tork değerleri kullanılarak, sürücünün duruşu ve o anki fiziksel durumu yorumlanabilir. Bu tez çalışmasında bu amaçla 6 eksenli kuvvet/tork sensörü ve bir enkoder içeren bir deney düzeneği tasarlanıp üretilmiştir. Bu düzenek ile sürücü postür analizi, tensör matrisinin kartezyen bileşenleri dönüşüm yasasını kullanarak yapılmıştır. Aynı zamanda deneysel olarak elde edilen veriler ile bir makine öğrenmesi algoritması geliştirilerek sürücünün sürüş esnasındaki bazı temel davranışları da incelenmiştir.

Anahtar Kelimeler: Sürücü postürü, sürücü durumu tespiti, sürücü dikkatsizliği

tespiti, sürüş tanımlama, makina öğrenmesi



Driver Posture Recognition and Machine Learning-Based Posture Estimation

Ceyhun IBOLAR

Department of Mechatronics Engineering
Master of Science Thesis

Advisor: Assoc. Prof. Erhan AKDOĞAN

Co-advisor: Dr. A. Emre ÇETİN

Most of the traffic accidents are because of driver distraction that may be as a result of engagement in a secondary activity or the driver's state of mind (sleepy, tired, angry, etc.). There are many studies available in the literature about determining and monitoring the driver attention. Almost all academic studies and commercial products on some luxury intelligent cars on the market use vision based systems for this purpose. Furthermore, researches are ongoing to improve the algorithms that interpret the driver's physical measures. On the other hand, driver performance can be interpreted using the variables collected in intelligent cars by suitable algorithms. Some commercial products use the history of steering wheel torque and position variables to identify the driver distraction. With continually improving technology, sensors are evolving both in terms of precision and price. An example is the torque sensor which started to be used to measure the torque applied by the driver in order to determine the level of assistance of the steering system. Torque/force sensors can be used to interpret both the driver posture and the state of mind as the steering wheel is one of the most critical parts of the car where the driver interacts with. For instance, if the driver is maneuvering with single hand, the position of the driver hand can be determined analytically from the torques and forces that the driver exerts to the steering wheel. Hence, interpreting the driver's posture as well as state of mind may become possible using the hand position and the torques and forces that the driver exerts. In this study, a steering wheel jig is designed and produced equipped with a 6 axes torque/force sensor and an encoder in order to develop algorithms to determine

some basic driver's behavior with machine learning algorithms.

Keywords: Driver posture, driver state identification, determination of driver inattention, driving identification, machine learning



1 GİRİŞ

Günümüzde, trafik kazalarının çoğu sürücülerin dikkatsizliğinden kaynaklanmaktadır. TÜİK'in 2018 verilerine göre, 2018 yılında Türkiye'de 1,2 milyondan fazla trafik kazası gerçekleşmiş olup, bu kazaların %89'u sürücülerin dikkatsizliğinden dolayı meydana gelmiştir [1]. Sürüş esnasında sürücülere yardımcı olmak ve kazaların sayısını azaltmak için yeni geliştirilen araçlara akıllı sürüş sistemlerinin eklendiği gözlenmektedir. Akıllı sürüş sistemleri, sürüş esnasında sürücünün sergilediği davranışları incelemektedir. Bu tür sistemlerin temel amacı, sürücünün davranışlarını otomatik olarak yorumlayarak, sürüş esnasında ortaya çıkması istenmeyen davranışlara çeşitli uyarılarda bulunarak engel olmaktır. Sürüş esnasında telefon ile konuşmak, mesajlaşmak, uykusuz sürüş yapmak ve dikkatsiz bir biçimde radyo ile uğraşmak bazı istenmeyen davranışlara örnek olarak gösterilebilir. Akıllı sürüş sistemleri ile bu tarz davranışların tespiti yapılarak, sürücüye sesli veya titreşim yolu ile yapılan uyarılar yardımıyla trafik kazalarının meydana gelmesi önlenabilir. Son yıllarda yeni nesil araçlarda akıllı sürüş sistemleri kullanılmaya başlanmış olup bu sistemlere sürekli olarak yeni özellikler eklenerek geliştirildiği gözlenmektedir. Literatürde akıllı sürüş sistemleri ile ilgili çalışmaların neredeyse tamamında görüntü işleme tekniklerinin kullanıldığı görülmektedir. Görüntü işleme yöntemi ile sürücünün istenmeyen davranışlarını daha doğru tahmin edebilmek için bu tez çalışmasında bir **sürücü postür algoritması** geliştirilmiştir. Sürücü postür algoritmasında sürücünün duruşu, etkileşimde olduğu direksiyon üzerinden, direksiyona uygulanan kuvvet/tork ve direksiyon açısı ölçümleri alınarak yapılmakta ve bu ölçümler sonucunda sürücünün direksiyonu tutma şekline bağlı olarak sürücü postür analizi yapılmaktadır. Buna göre, sürücünün direksiyonu nereden tuttuğu, direksiyona ne kadar kuvvet/tork uyguladığı ve aracı hangi el ile sürdüğü geliştirilen algoritma tarafından ortaya konmaktadır. Bu sonuçlar ile görüntü işleme tekniklerinin hibrid olarak akıllı sürüş sistemlerinde kullanılması daha etkin ve yüksek performanslı sonuçlar ortaya koyacaktır.

1.1 Literatür Özeti

Literatürde sürücü postürü ile ilgili yapılan akademik ve bilimsel çalışmalarda görüntü işleme, biyolojik ölçüm ve sürücü-direksiyon ilişkisi algoritmalarına ilişkin çeşitli yöntemlerin geliştirildiği görülmekte olup, görüntü işleme tekniklerinin diğer yöntemlere göre daha fazla kullanıldığı göze çarpmaktadır. Görüntü işleme tekniklerinin yaygın kullanımının nedeni; alınan görüntüyü dijital veya analog hale getiren basit ve güvenilir bir yöntem olmasıdır. Biyolojik ölçümlere göre geliştirilen yöntemlerde ise, insanın sürüş esnasında sergilediği postür (sürücünün koltuğa oturma şekli, direksiyona olan uzaklığı, vb.) test amaçlı incelenir. Ancak bu yöntem, gerçek zamanlı olmayan bir sürücü postür analiz yöntemidir ve bu tür sistemlerin amacı sürüş esnasında sürücünün optimum fiziki postürünü belirlemektir. Sürücü postür algoritmaları ise şu ana kadar kendine literatürde kısıtlı alan bulabilmiştir. Ancak araştırma çalışmalarının bu konu üzerine odaklanmaya başladığı buna rağmen literatürde hala ciddi bir boşluk olduğu görülmektedir.

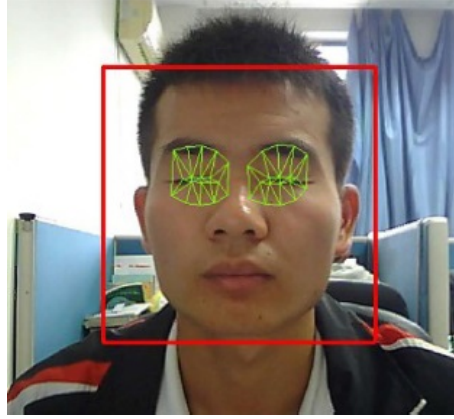


Şekil 1.1 Literatürde göz tanıma algoritması [2]

Zhang vd. [2], görüntü işleme yöntemini kullanarak yaptıkları çalışmada, sürüş esnasında sürücüde yorgunluk/uyuşukluk olup olmadığını tahminleyen bir algoritma geliştirmiştir. Bu algoritmada, sürücü duruşundaki değişiklikleri tespit edebilmek için Şekil 1.1'de görüldüğü gibi bir göz tanıma algoritması kullanılmıştır. Kullanılan kamera ile göz kapağı kapanma süresi, göz kapağının maksimum kapanma süresi, göz kırpma sıklığı, gözlerin ortalama açılma seviyesi, gözlerin açılma ve gözlerin kapanma hızı olarak toplam altı değer ölçülmüştür. Ölçülen değerler Fisher'ın doğrusal ayrıcı işlevi yöntemi kullanılarak işlenmiş ve buna göre sürücüde yorgunluk/uyuşukluk olup olmadığına dair bir sonuca varılmıştır.

Xie vd. [3], görüntü işleme yöntemini kullanarak yüz tespiti yapan bir algoritma geliştirmiştir. Şekil 1.2'de görüldüğü üzere bu çalışmada, yüzün merkezi hesaplanmış olup, kafa hareketleri izlenmekte ve bunun sonucunda Adaboost yöntemi kullanılarak, yorgunluk algılama sistemi geliştirilmiştir. Aynı zamanda "Active Appearance Model" yöntemi ile gözlerin açık ve kapalı olma durumu tespit edilmiştir. Zhang vd. yaptıkları

alıřma ile karřılařtırıldıęında, iki ayrı algoritmanın birlikte kullanılması, Jian-Feng Xie'nin yntemini daha gvenilir hale getirmiřtir.



řekil 1.2 Literatr-gz tanıma resmi [3]

Teyeb vd. [4], grnt iřleme yntemini kullanarak, srcnn uyanıklık durumunu doęrulamak iin gzlerin kapanma sresini ve gz kırpma sıklıęını inceleyen bir algoritma geliřtirmiřtir. Bu sistemin sonucunda makineğrenmesi yntemini de kullanarak, srcnn uyku durum tahmini yapılmıřtır.

Abouelnaga vd. [5], grnt iřleme yntemini kullanarak elde edilen veriler ile gerek zamanlı src dikkatsizlięini tespit eden bir algoritma geliřtirmiřtir. Src dikkatsizliklerine dair senaryolar (telefon ile konuřmak, mesajlařmak, radyo ile uęrařmak, arataki dięer bulunan kiřiler ile konuřmak) řekil 1.3'de grldę gibi olup, bu senaryolar dikkate alınarak src davranıřlarına dair veriler elde edilmiřtir. Elde edilen bu verileri evriřimsel sinir aęları (Convolutional Neural Network) yntemi ile iřleyerek, src davranıřları ile ilgili tahminler yapılmıřtır.



řekil 1.3 Literatrde src dikkatsizlik senaryoları [5]

Liang vd. [6], grnt iřleme yntemini kullanarak, srcnn yorgunluęunu tahmin eden bir algoritma geliřtirmiřtir. Bu algoritma ile araca yerleřtirilen bir kamera

kullanılarak sürücünün göz hareketleri ve yol şeridini ne kadar doğru bir biçimde takip ettiği incelenmiştir. Elde edilen veriler makine öğrenmesi yöntemi ile işlenerek, sürücünün yorgunluk seviyesi tahmin edilmiştir. Sistem elde edilen tahmin sonuçlarını kullanarak sürücü dikkatinin dağıldığı durumlara yönelik uyarılar vermektedir.

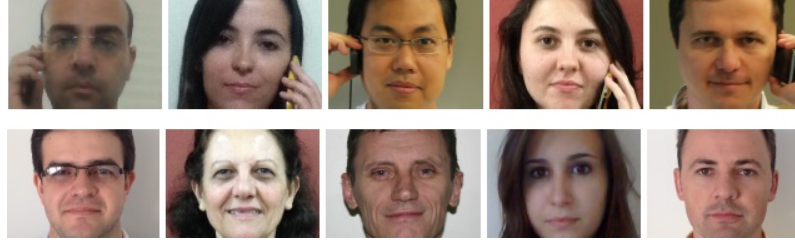
Kutilla vd. [7], çalışmalarında, sürücünün dikkat dağınıklığını, görsel ve bilinçsel olarak iki farklı açıdan incelemiştir. Görüntü işleme yöntemi ile sürücünün göz hareketleri ve şeridi takip etme doğruluğuna dair veriler elde edilmiş ve bu veriler makine öğrenmesi algoritmasında kullanılmıştır. Makine öğrenmesi algoritması oluşturulurken, en iyi sonuç destek vektör makinesi (SVM) yönteminde elde edilmiştir. Görsel dikkat dağınıklığının doğru tespit oranı %80 seviyelerinde gerçekleşmiş olup, bilinçsel dikkat dağınıklığının doğru tespit oranı %68-80 aralığında gerçekleşmiştir.

Wöllmer vd. [8], sürücünün dikkat dağınıklığını, uzun-kısa süreli bellek (LSTM) yöntemini kullanarak inceleyen gerçek zamanlı bir sistem geliştirmiştir. Bu sistemde, araca yerleştirilen bir kamera ile, sürücünün yedi farklı senaryoda sergilediği davranışlara ait veriler kaydedilmiştir. Senaryolar sırasıyla: radyonun sesinin değiştirilmesi, bir sonraki şarkıya geçilmesi, telefon rehberine kayıtlı bir kişinin bulunmaya çalışılması, telefon haritasından en yakın gaz istasyonunun aranması, telefon kullanımı, telefonda açık olan navigasyonda bir şehir aratılması, araçtaki televizyon modunun değiştirilmesi, telefonda açık olan navigasyon sesininin kısılmasıdır. Bu senaryolardan sürücü davranışlarına dair elde edilen veriler ile uzun-kısa süreli bellek yöntemi kullanılarak bir yapay sinir ağı modeli geliştirilmiştir.

Liu vd. [9], sürücü dikkat dağınıklığını yarı denetimli makine öğrenmesi yöntemi kullanarak inceleyen bir algoritma geliştirmiştir. Bu algoritmada, veriler bir kamera yardımı ile elde edilmiştir ve görüntü işleme yöntemi ile sürüş esnasında sürücünün sergilediği farklı senaryolardaki, yüz ve göz hareketlerini içermektedir. Bu veriler, yarı denetimli makine öğrenmesi yöntemi ile işlenerek, sürücünün dikkat dağınıklığına ilişkin sonuçlar ortaya konmuştur.

Berri vd. [10], sürüş esnasında sürücünün telefon kullanıp kullanmadığını tespit eden bir algoritma geliştirmiştir. Bu algoritmada, araca yerleştirilen bir kamera ile sürücü davranışları incelenmiş olup, Şekil 1.4'de görüldüğü üzere, sürücü davranışı sürücünün telefonu kullandığı durumlar ve sürücünün telefonu kullanmadığı durumlar olmak üzere ikiye ayrılmıştır. Deneme sürüşlerinde görüntü işleme yöntemi ile sürücüye ait veriler elde edilmiş ve bu veriler ile makine öğrenmesi gerçekleştirilerek bir tahmin algoritması geliştirmiştir. Makine öğrenmesi algoritmalarında en yüksek doğruluk destek vektör makinesi ile elde edilmiştir.

Baheti vd. [11], evrimsel sinir ağları yöntemini kullanarak sürücü dikkat



Şekil 1.4 Literatürde telefon kullanımı [10]

dağınıklığını tespit eden bir algoritma geliştirmiştir.



Şekil 1.5 Literatürde sürücü dikkatsizlik durumları [11]

Bu çalışmada, araca yerleştirilen bir kamera yardımı ile sürücünün dikkatini bozan Şekil 1.5'deki senaryolar incelenmiştir. Bu senaryolar doğrultusunda, sürücüye ait, görüntü işleme yöntemi ile elde edilen veriler, evrimsel sinir ağları yönteminde işlenerek, dikkat dağınıklığının tespiti amacıyla kullanılmıştır.

Das vd. [12], sürüş esnasında, görüntü işleme yöntemi ile ellerin direksiyon üzerinde bulunduğu konumu tespit eden bir algoritma geliştirmiştir (Şekil 1.6). Bu çalışmada, ellerin bulunduğu konumun doğru olup olmadığını tespit etmeye yönelik olarak "Aggregate Channel Features" yöntemi kullanılmıştır. Sistem sürücünün direksiyonu yanlış tutma durumunda sürücüye uyarı vermektedir.



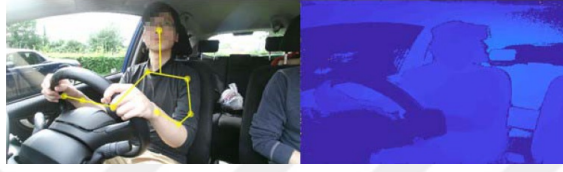
Şekil 1.6 Literatürde el konum tespiti [12]

Craye vd. [13] Kinect sensörü kullanarak sürücü dikkat dağınıklığını tespit eden bir

algoritma geliřtirmiřtir. Kinect sensörünün verdiđi harita görünümündeki renk ve derinlik verilerine dayanarak, sürücünün göz hareketi, elinin konumu, kafa hareketi ve yüz ifadeleri incelenmiřtir. Bu veriler, Adaboost ve "Hidden Markov Model" yöntemleri ile işlenerek, sürücüye ait dikkat dađınıklılıđı sınıflandırması yapılmıřtır.

Ohn-Bar vd. [14], görüntü işleme yönetimi ile sürücünün baş, göz ve el konumu ile ilgili verileri temin eden ve bu verileri destek vektör makinesi yöntemi ile ele alarak, sürücünün dikkat dađınıklılıđını tespit eden bir sistem geliřtirmiřtir.

Xing vd. [15], Kinect sensörünü (Şekil 1.7) kullanarak sürücünün dikkat dađınıklılıđını tespit eden bir sistem geliřtirmiřtir.



Şekil 1.7 Literatürde Kinect sensörü [15]

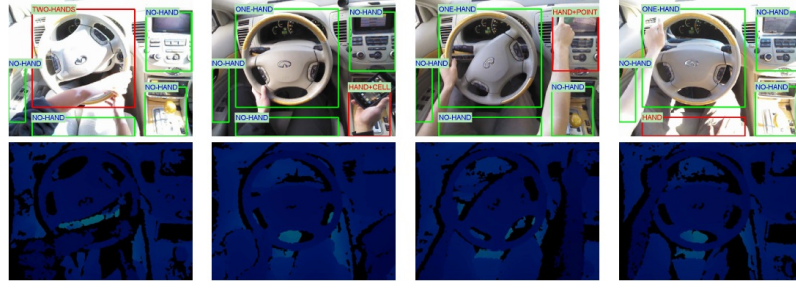
Bu çalışmada, sađ řerit kontrolü, sol řerit kontrolü, dikiz ayna kontrolü, telefon ile konuşmak, mesajlaşmak ve radyo ile uğrařmak gibi yedi farklı senaryo ele alınmıřtır. İlk üç senaryo güvenli sürüş olarak kabul edilmiřtir. Son dört senaryo ise dikkat dađınıklılıđı kategorisine giren senaryolar olarak ele alınmıřtır. Bu çalışmada Kinect sensöründen alınan veriler, ileri beslemeli sinir ađları (feedforward neural network - FFNN) yöntemi ile işlenerek, sürücü dikkat dađınıklılıđını tespit eden sistem geliřtirilmiřtir.

Amin vd. [16], sürücü duruşlarının ve hareketlerinin sürekli izlenmesi ve tanımlanması için düşük çözünürlüklü bir kızılötesi (IR) görüntüleme sistemi geliřtirmiřtir. Bu sistemde, kızılötesi sensörü ile alınan veriler dođrultusunda, yapay sinir ađları yöntemini kullanarak sürücüyü izleyen ve duruş tespiti yapan bir algoritma geliřtirilmiřtir.

Le vd. [17], sürücünün cep telefonu kullanıp kullanmadıđını otomatik olarak belirleyen ve ellerinin direksiyonda olup olmadıđını tespit eden, bölge teklif ađı (Region Proposal Network - RPN) tabanlı bir sistem geliřtirmiřtir.

Ohn-Bar vd. [18], Kinect sensörü kullanarak sürücünün el hareketlerini inceleyen bir sistem geliřtirmiřtir. Buna ek olarak, bu sistemden elde edilen verileri, destek vektör makinesi yöntemi ile işleyerek, sürücünün direksiyonu dođru tutup tutmadıđını analiz eden bir algoritma geliřtirilmiřtir.

Gupta vd. [19], yapay sinir ađları ile sürücünün dikkat dađınıklılıđını tespit eden



Şekil 1.8 Literaturde kinect sensörü 2 [18]

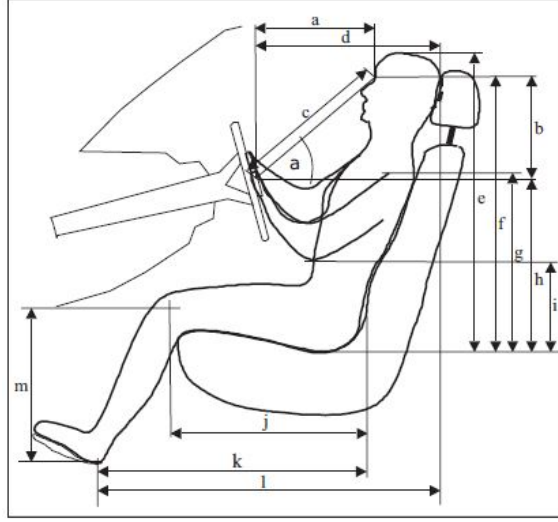
bir sistem geliştirmiştir. Bu çalışmada aracın içerisine yerleştirilen bir kamera ile sürücünün davranışları incelenmiştir. Bu davranışlar altı farklı senaryoda ele alınmış ve bu senaryolar içerisinden üçü güvenli sürüş olarak nitelendirilmiş olup, diğer üçü dikkat dağınıklığına sebebiyet veren davranışlar olarak ele alınmıştır.

Riener vd. [20], biyolojik işaretler tabanlı sürücü postür analizi yapan bir sistem geliştirmiştir. Geliştirilen bu sistemde sürücü koltuğuna göze çarpmayacak şekilde yerleştirilmiş, yüksek çözünürlüklü bir basınç sensörü kullanılmıştır. Bu sensör ile pelvik kemiğinin direksiyona olan mesafesi analiz edilmiş olup, elde edilen bu biyometrik veriler ile bir duruş tanıma tekniği geliştirilmiştir. Sürücünün kendini öne doğru eğdiği durum örnek bir senaryo olarak ele alınmıştır. Bu durumda, koltuğun arka kısmındaki basınç sensöründen giderek daha düşük basınç değeri alınacağından, sürücünün o anki duruşu ile ilgili tespit yapılmıştır. Bu gibi ortaya çıkabilecek olan bütün duruş senaryolarını, sürücünün biyolojik ölçümleri ile beraber analiz ederek, sürücünün sürüş esnasındaki davranışları analiz edilmeye çalışılmıştır.

Fu vd. [21] EEG (Elektroansefalografi), EMG (Elektromiyogram) ve solunuma ilişkin biyolojik işaretleri aynı anda işleyebilen, giyilebilir sensörler tarafından kayıt edilen ve Bluetooth ile bu verileri bilgisayara gönderen bir sistem geliştirmiştir. Alınan sinyaller "Hidden Markov Model" yönteminde işlenerek, sürücünün yorgunluk seviyesi tespit edilmiştir.

Kovaçevic vd. [22], sürücüye ait antropometrik ölçümlerin, ergonomik sürücü duruşu ve güvenliği üzerindeki etkisini analiz eden bir sistem geliştirmiştir (Şekil 1.9). Bu sistemin amacı, elde edilen antropometrik ölçümler ile sürüş esnasında sürücünün optimum fiziki postürünü belirlemektir.

Tanaka vd. [23] sürücünün direksiyona uyguladığı kuvvet ve direksiyonu tutma şeklini dikkate alarak, sürüş esnasında meydana gelen farklı engellere sürücünün reaksiyonlarını tespit eden bir sistem geliştirmiştir. Sürücünün engellere karşı gösterdiği bu reaksiyonlar sürücünün el empedansı, direksiyon açısı ve torka bağlı olarak belirlenmiştir. Elde edilen değerler ile bir kontrol algoritması oluşturularak,



Şekil 1.9 Literatürde biyolojik ölçüm [22]

sürücünün daha rahat bir sürüş sağlaması amaçlanmıştır.

1.2 Tezin Amacı

Akıllı sürücü sistemlerinin temel hedefi sürüş esnasında sürücü davranışını tespit etmek ve muhtemel kazaları önlemek için geri bildirim oluşturmaktır. Bu amaçla araştırmacılar tarafından farklı yöntemler geliştirilmiştir. Bu tez çalışmasının temel amacı, akıllı sürüş sistemlerinde kullanılmak üzere görüntü işleme teknikleri ile birleştirilerek sürücü davranışlarının daha hassas biçimde tahmini için yeni bir sürücü postür algoritması geliştirmektir. Geliştirilen algoritma, direksiyona yerleştirilen bir sensör yardımı ile sürücünün direksiyona uyguladığı kuvvet/tork değerleri kullanarak sürücü - direksiyon etkileşimini analiz eder. Bu analiz sonucunda, sürücünün elinin direksiyon üzerindeki konumu hesaplanır ve sensörlerden alınan kuvvet/tork ve direksiyon açısı değerleri makine öğrenmesi algoritmasında kullanılarak sürücünün direksiyonu hangi el ile sürdüğü tahmin edilir.

1.3 Orjinal Katkı

Bölüm 1.1’de, literatürde bu güne kadar sürücü postür tanımlama ve buna bağlı olarak sürücü durum tespiti ile ilgili yapılan çalışmalar açıklanmıştır. Tablo 1.1’de bu çalışmalarda kullanılan sensör ve yöntem bilgilerine yer verilmiştir. Literatürde kullanılan sistemlerin amaçları ve bu tez çalışmasından farklılıkları aşağıda kısaca ifade edilmiştir:

1. Önceki çalışmalarda görüntü işlemenin, sürücü durum tespitinde kullanılan en

yaygın yöntem olduđu göze çarpmaktadır. Tablo 1.1’de, görüntü işleme yöntemi kullanılan [2], [3], [4], [6] ve [9] referanslı bu çalışmalarda sürücünün yorgun, uykulu gibi durumları tespit edilmiştir. Görüntü işleme yöntemi kullanılan diğer çalışmalarda sürücünün fiziksel dikkat dağınıklığı tespit edilmiştir (telefon ile konuşmak, ellerin direksiyonda olmaması, mesajlaşmak, vb.). Günümüzde akıllı sürüş sistemleri ile donatılmış yeni nesil araçlarda, bu çalışmalardan elde edilen veriler aracın kontrolüne kadar müdahale edebilmektedir. Bu açıdan bakıldığında, elde edilen veriler ile karar alma süreçlerinin doğruluđu büyük önem arz etmektedir. Buna istinaden, görüntü işleme yöntemi ile karar alan sistemlerin ilave yöntemlerle desteklenerek güçlendirilmesi, akıllı sürüş sistemlerini daha güvenilir hale getirmeye yardımcı olur. Bu tez çalışmasında, tork/kuvvet sensörü kullanılarak sürücünün davranışı ile ilişkili analitik veriler toplanmaktadır. Bu verilerin, görüntü işleme yöntemlerinden elde edilen veriler ile birlikte kullanılması sürücü davranışlarına ilişkin daha doğru tahmin yapılabileceğı sonucunu doğuracağı ön görülmektedir.

2. Tablo 1.1’de yer alan ve Tanaka ve diğ. [23] tarafından gerçekleştirilen bir diğer çalışmasında da kuvvet/tork sensörü kullanılmıştır. Ancak bu tez çalışmasından farklı olarak, kuvvet/tork sensörü, engebeli yolda sürüş rahatlığı sağlama amacıyla kullanılmıştır ve gerçek zamanlı tespit yapan bir yöntem değildir.
3. Yukarıda bahsedilen gerçek zamanlı tespit yapan yöntemlere ek olarak, literatürde biyolojik parametre ölçüm yöntemleri ile sürücü postür analizi yapan, gerçek zamanlı olmayan sistemler de görülmektedir. Bu sistemlerin amacı ise, sürüş esnasında sürücünün optimum fiziki postürünü belirlemektedir. Aynı zamanda bu sistemler sürücünün ruhî ve fiziki davranışları ile ilgili gerçek zamanlı bir tespit yapmamaktadır.

Literatürdeki çalışmalar ele alındığında, bu tez çalışmasında Tablo 1.1’de görüldüğü üzere diğer çalışmalardan farklı olarak, görüntü işleme yöntemi kullanmadan, sadece bir kuvvet/tork sensörü kullanarak sürücünün ellerinin direksiyondaki konumu, aracı hangi el ile sürdüğünü, direksiyonu tuttuğı noktada ne kadar kuvvet ve tork uyguladığı sonuçları elde edilmiştir. Diğer yöntemlerde direksiyona uygulanan kuvvet/tork değerleri ölçülmemektedir. Kuvvet ve tork sensöründen elde edilen tüm verilerin doğru bir şekilde işlenerek görüntü işleme ile birlikte kullanılması durumunda, akıllı sürüş sistemlerinin daha etkin ve doğru sonuç üreteceğı beklenmektedir.

Tablo 1.1 Literatürdeki sürücü postür analiz yöntemleri

Referans No.	Sürücü Postür Analiz Yöntemi	Kullanılan Sensör	Tespit Yöntemi
[2]	Görüntü İşleme	Kamera	Fisher
[3]	Görüntü İşleme	Kamera	AdaBoost Active Appearance Model
[4]	Görüntü İşleme	Kamera	Karmaşık Dalgacık Dönüşümü Tabanlı YSA
[5]	Görüntü İşleme	Kamera	Evrişimsel Sinir Ağları
[6]	Görüntü İşleme	Kamera	Destek Vektör Makineleri
[7]	Görüntü İşleme	Kamera	Destek Vektör Makineleri
[8]	Görüntü İşleme	Kamera	Uzun-Kısa Süreli Bellek YSA
[9]	Görüntü İşleme	Kamera	Yarı Denetimli Makine Öğrenmesi
[10]	Görüntü İşleme	Kamera	Destek Vektör Makineleri
[11]	Görüntü İşleme	Kamera	Evrişimsel Sinir Ağları
[12]	Görüntü İşleme	Kamera	Aggregate Channel Features (MÖ)
[13]	Görüntü İşleme	Kinect	AdaBoost ve Hidden Markov Model
[14]	Görüntü İşleme	Kamera	Destek Vektör Makineleri
[15]	Görüntü İşleme	Kinect	İleri Beslemeli Sinir Ağları (FFNN)
[16]	Görüntü İşleme	Kamera	Yapay Sinir Ağları
[17]	Görüntü İşleme	Kinect	Raster RCNN, Region Proposal Network
[18]	Görüntü İşleme	Kamera	Bag-of-Words
[19]	Görüntü İşleme	Kamera	Yapay Sinir Ağları
[20]	Biyolojik Ölçüm	Basınç Sensörü	Matematiksel Algoritma
[21]	Biyolojik Ölçüm	EEG ve EMG	Hidden Markov Model
[22]	Biyolojik Ölçüm	Ağırlık Sensörü	Antropometrik Ölçüm
[23]	Sürücü Postür Algoritmaları	Kuvvet Sensörü	Matematiksel Algoritma
Tez Çalışması	Sürücü Postür Algoritması	Kuvvet/Tork Sensörü, Enkoder	Matematiksel Algoritma, Destek Vektör Makineleri

1.4 Yöntem

Bu tez çalışmasında sürücü postür analizi iki yöntem ile gerçekleştirilmiştir. Birinci yöntem sürücünün ellerinin direksiyon üzerinde bulunduğu konumu hesaplamaya yöneliktir. Bu yöntemde **tensör matrisinin kartezyen bileşenlerinin dönüşüm yasası** temel alınmıştır. İkinci yöntemde ise **makine öğrenmesi algoritması** kullanılmıştır. Bu yöntemde, sensör tarafından elde edilen sürücüye ait veriler ile sürücünün direksiyonu hangi el ile sürdüğü tahmin edilmiştir.

1.4.1 Tensör Matrisi

Tensör, Latince "Stres" veya "Gerilim" anlamına gelen *tensus* kelimesinden türetilmiştir ve vektörel veya skaler büyüklükler arasındaki doğrusal ilişkileri tanımlayan matematiksel nesneler olarak açıklanır. Bir tensörün türünü belirtmek için "rank" adı verilen dereceler kullanılır. "Rank 0" tensörleri tek bir boyutlu dizi kullanılarak tarif edilebilen fiziksel sistemler için kullanılır ve bu tensörlere skalar adı verilir. "Rank 1" tensörleri bir sayı dizisi ile ifade edilen sistemlerin tanımlanması için kullanılır ve bu tensörler vektör olarak isimlendirilir. "Rank 2" tensörleri vektör dizisi ile ifade edilen sistemler için kullanılır ve tensör matrisi olarak adlandırılır. Bu tez çalışmasında kuvvet/tork sensörü ve enkoderden gelen değerlerin tanımı için "Rank 2" tensör matrisi kullanılmıştır. "Rank 2" derecesine sahip olan bu tensör matrisleri aynı zamanda sadece **tensör** olarak da adlandırılır. Bu tez çalışmasında tensör matrisinin kartezyen bileşenleri yöntemi kullanılarak sürücü postür analizi yapılmıştır. Bu yöntemin anlaşılabilir hale gelmesi için ilk olarak tensör matrisinin bazı temel kuralları bilinmelidir.

Bir vektörün bileşenleri, bileşenleri tanımlamak için kullanılan baz vektörlere bağlıdır. Bu kural aynı şekilde tensörler için de geçerlidir. Buna istinaden, tensör matrisinin bileşenlerini tanımlamak için x_1 , x_2 ve x_3 yönlerinde olan n_1 , n_2 ve n_3 birim vektörleri ele alınır. n_1 , n_2 ve n_3 birim vektörleri dönüşüm (transformation - T) sonrasında Tn_1 , Tn_2 ve Tn_3 şeklinde ifade edilir (Denklem 1.1, Denklem 1.2 ve Denklem 1.3).

$$Tn_1 = T_{11}n_1 + T_{21}n_2 + T_{31}n_3 \quad (1.1)$$

$$Tn_2 = T_{12}n_1 + T_{22}n_2 + T_{32}n_3 \quad (1.2)$$

$$Tn_3 = T_{13}n_1 + T_{23}n_2 + T_{33}n_3 \quad (1.3)$$

Denklem 1.1'de T_{11} değeri hesaplandığında Denklem 1.4'deki ifade ortaya çıkmaktadır.

$$T_{11} = n_1 \cdot Tn_1 \quad (1.4)$$

Denklem 1.4'de i ve j sembollerini kullanarak genel bir şekilde yazıldığında Denklem 1.5 elde edilir.

$$T_{ij} = n_i \cdot T n_j \quad (1.5)$$

Bu durumda, Denklem 1.5 dikkate alınarak oluşturulan T tensörünün bileşenleri Denklem 1.6'da ifade edilen matris dizisinde gösterilmektedir.

$$[T] = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \quad (1.6)$$

n_1 , n_2 ve n_3 baz vektörleri i sembolü ile genel bir şekilde n_i ile ifade edilir. n_i ve n'_i ifadelerin, iki dikdörtgen kartezyen koordinat sistemine karşılık gelen birim vektörler olduğu durum elde alındığında, bu vektörler arasındaki bağlantı ortogonal tensör Q ile Denklem 1.7'de ifade edilir.

$$n'_i = Q n_i = Q_{mi} n_m \quad (1.7)$$

T tensör matrisinin n'_i baz vektörlerine göre bileşenleri Denklem 1.8'de ifade edilir.

$$T'_{ij} = n'_i T n'_j \quad (1.8)$$

Denklem 1.7'de yapılan tanım, Denklem 1.8'de yerine konulduğunda, Denklem 1.9 elde edilir.

$$T'_{ij} = Q_{mi} Q_{nj} T_{mn} \quad (1.9)$$

Denklem 1.9 matris şeklinde ifade edildiğinde, Denklem 1.10 elde edilir. Elde edilen bu denklem *kartezyen bileşenlerin dönüşüm yöntemi* olarak bilinmektedir. Bu tez çalışmasında bu yöntem, kuvvet/tork sensörü ve enkoder tarafından ölçülen değerler ile sürücünün ellerinin direksiyon üzerinde bulunduğu konumda oluşan değerler arasındaki analizi için kullanılmıştır.

$$[T] = [Q]^T [T'] [Q] \quad (1.10)$$

1.4.2 Makine Öğrenmesi

Makine öğrenmesi istatistiksel ve matematiksel yöntemleri kullanarak, daha önceden veya o an elde edilen verilerden çıkarımlar yapar ve yapılan çıkarımlar sonucunda bilinmeyen verilere dair tahminlerde bulunur. Makine öğrenmesi denetimli, denetimsiz ve pekiştirmeli olmak üzere üçe ayrılır. *Denetimli makine öğrenmesi* sınıflandırma ve regresyon olmak üzere kendi içerisinde ikiye ayrılır. Sınıflandırma yöntemi, makineye öğretilen sınıflar arasında bir sonuç vermektedir (siyah veya beyaz, doğru veya yanlış, sağ veya sol vb.). Regresyon yönteminde ise sonuç gerçek bir sayı olarak ortaya çıkmaktadır [24]. *Denetimsiz makine öğrenme yöntemi*, bilinmeyen bir yapı üzerine tahmin yapabilmek için kümeleme yöntemini kullanır. *Kümeleme yöntemi* tanımlanmış kriterlere göre benzerlik gösteren verilerin gruplandırılması olarak tanımlanır. Denetimsiz makine öğrenmesi yönteminde ise herhangi bir insan operatörü veya cevap anahtarı bulunmamaktadır. Bu yöntem, genellikle çok büyük veri setleri olduğu durumlarda kullanılır ve bu verilerle ilgili kümeleme yöntemi ile karar verme becerisine sahiptir. Denetimli makine öğrenmesi ise belirlenmiş veri setleri ile öğretilir ve sonrasında yeni gelen verilerle ilgili tahmin yapmaya çalışır. Denetimli ve denetimsiz makine öğrenmesi yöntemlerinden farklı olarak, *pekiştirmeli makine öğrenmesi yöntemi*, eğitim verisi olmadan bulunan çevreden alınan tepkiler ile öğretilir. Bu tez çalışmasında, sürücünün direksiyonu hangi el ile sürdüğü denetimli makine öğrenmesinin sınıflandırma yöntemi ile tahmin edilmiştir.

Sınıflandırma yöntemi dört aşamadan oluşur:

- Verilere erişilmesi veya keşfedilmesi
- Verilerin ön işlenmesi
- Öngörülü modelin geliştirilmesi
- Analitik çözümün sistemlere entegre edilmesi

Birinci aşamada, analizi yapılacak olan veriler elde edilir. Veriler yapılan çalışmaya göre farklı yöntemler ile elde edilebilir (örnek olarak bu çalışmada sistemin verileri kuvvet/tork sensörü ile elde edilmiş olup, bu elde edilen veriler uygun format biçimde dosyalara kayıt edilmiştir). Sınıflandırma yöntemini kullanan makine öğrenmesi algoritmalarında tahmin yapılabilmesi için iki adet veri türü gerekir. Birincisi, eğitim verisi olarak isimlendirilen, algoritmanın öğrenmesi için kullanılan veri setidir. İkincisi, test verisi olarak isimlendirilen, algoritmayı test etmek için kullanılan, daha önce algoritmaya verilmeyen veri setidir.

İkinci aşamada, elde edilen eğitim verileri ile özellik çıkarım işlemi yapılır. Özellik çıkarım işlemi tecrübesiz bir insana bir işi öğretmek olarak anlatılabilir. Örneği, kalbi ile ilgili rahatsızlığı olan bir hastanın, sorununu tespit edebilmek için hastanın, "nerede doğduğu" bilgisi (özelligi) bir anlam ifade etmeyecektir. Onun yerine, hastanın "sigara kullanıyor mu, daha önce benzer bir hastalık geçirdi mi" gibi bilgilerinin hesaba katılması daha faydalı olacaktır. Örnekte görüldüğü üzere, ikinci aşamada, yapılan çalışmaya en uygun özellikler belirlenir ve özellik çıkarma işlemi tamamlanır.

Üçüncü aşamada, özellik çıkarım işlemi tamamlandıktan sonra bir matris dizisi oluşturulur. Makine öğrenme algoritması bu matris üzerinden çalışır. Elde edilen matris sonucunda, daha önce görülmeyen verilere dair tahmin amaçlı bir model oluşturulur.

Dördüncü aşama, üçüncü aşamada elde edilen modelin sisteme entegrasyonunu içerir. Makine öğrenmesinin programlanabilmesi için farklı programlama dilleri kullanılır (MATLAB, Python, C, C++). Bu tez çalışmasında makine öğrenmesi algoritması MATLAB tabanlı geliştirilmiştir.

1.5 İçerik

Bölüm 2.1’de sürücü postürü ile ilgili temel kavramlar açıklanmıştır. Bölüm 2.2’de geliştirilen direksiyon sisteminin analizi yapılmıştır. Yapılan bu analiz ile sistemin matematiksel modeli elde edilmiş olup, model ve ölçülen parametreler ile sürücünün ellerinin direksiyonda bulunduğu konum hesaplanmıştır. Bölüm 2.3’de deney düzeneği açıklanmıştır. Bölüm 2.4’de mobil, web ve konsollar için kolay bir şekilde 3D oyunlar ve uygulamalar oluşturmaya yarayan Unity programı üzerinde geliştirilen arayüz ve Unity programının MATLAB ile haberleşmesi açıklanmıştır. Bölüm 3.1’de geliştirilen makine öğrenmesi algoritması açıklanmıştır. Bölüm 3.2’de Simulink modelin gerçek zamanlı çalışması açıklanmıştır. Bölüm 3.3 ve Bölüm 3.4’de deney düzeneğinde yapılan sürüş testleri sonucunda sürücü postür ve makine öğrenmesi algoritmasına ait deneysel sonuçlar verilmiştir. Bölüm 4’de bu tez çalışmasına ait sonuç ve tartışmaya yer verilmiştir.

2.1 Sürücü Postürü

Türkiye’de yapılan araştırmalara göre, en çok kullanılan ulaşım yöntemi karayoludur [25]. Trafiğin yoğun olduğu yerlerde, sürüş süreleri uzun olmaktadır. Trafikte yoğun zaman geçiren sürücülerin, sürüş esnasında sergiledikleri davranışlar, kaza riskini beraberinde getirmektedir.

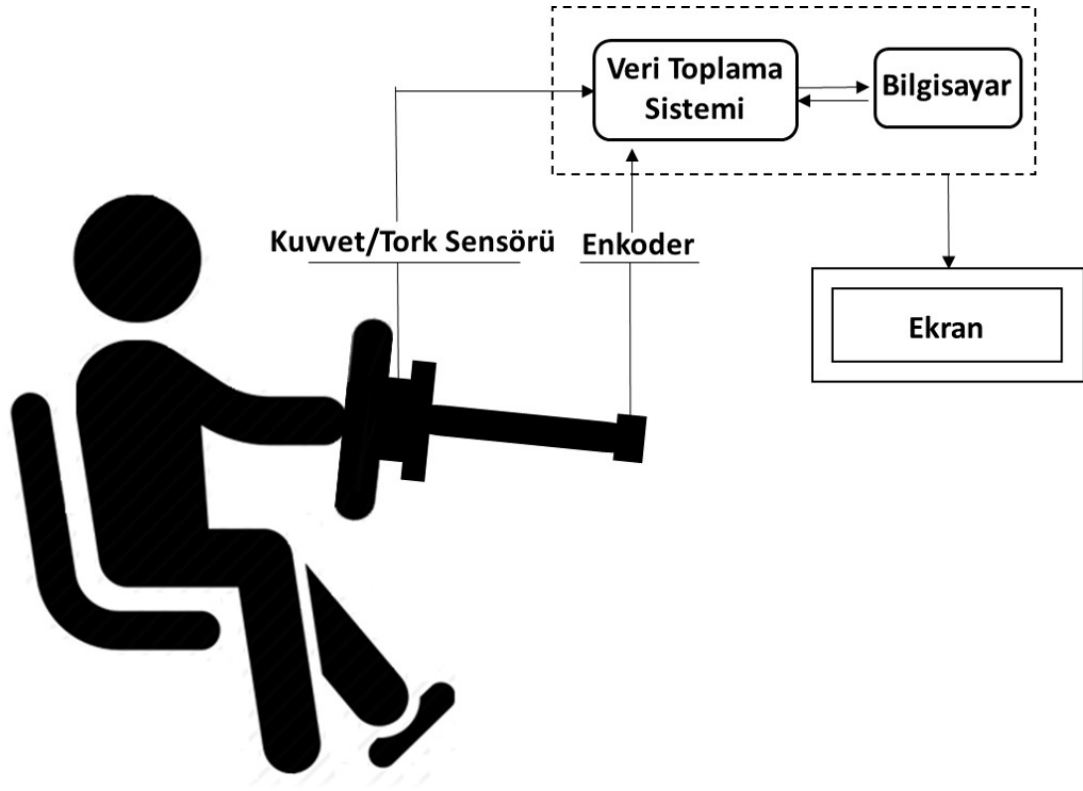
Sürüş esnasında sürücünün sergilediği bütün davranışlar, sürücü postürü olarak tanımlanmaktadır [26]. Sürücünün aracı kullanırken direksiyonu tutma şekli, direksiyona olan uzaklığı, göz hizası ve ayna ayarı sürücü postürünü belirleyen unsurlardan bazılarıdır. Aracı kullanırken doğru bir sürücü postürüne sahip olduğunda, uzun bir süre araç kullanmaktan dolayı meydana gelebilecek olan kas-iskelet problemleri azalır. Bu aynı zamanda, doğru postüre sahip bir sürücünün, kaza durumunda daha iyi korunacağı anlamına da gelir. Örneğin direksiyona çok yakın oturulması durumunda, emniyet kemerinin, çarpışma esnasında tepki göstermesi için yeterli zamanı olmayabilir. Benzer şekilde, hava yastıklarının açılması için yeterli alan olmayabilir. Bu durumda hava yastığı yüz yaralanmalarına da sebebiyet verebilir. Buna ek olarak, direksiyondan uzak oturmakta postür açısından yanlış bir araç kullanım tarzı olarak nitelendirilir. Sürüş esnasında ellerin direksiyondan çok uzak olması durumunda, direksiyon kontrolü istenildiği gibi yapılamamaktadır. Yukarıda bahsedilen sürücü postürüne ait olası problemler direksiyonun tutulma şeklinin önemine işaret etmektedir. Bu çalışmada, sürücünün direksiyonu tutma şekli analiz edilmiştir.

2.2 Sistemin Modellenmesi

2.2.1 Direksiyon Sistemi

Şekil 2.1’de bu çalışmada geliştirilen direksiyon sisteminin şematik görünümü verilmiştir. Sistem donanımsal olarak kuvvet/tork sensörü, enkoder, veri toplama

sistemi ve bilgisayar içermektedir. Sürücü tarafından direksiyona uygulanan kuvvet ve tork değerleri, kuvvet/tork sensörü ile ölçülmekte ve veri toplama sistemi ile bilgisayar ortamına aktarılmaktadır. Sürücünün direksiyon simidini döndürmesi ile oluşan direksiyon açısı veri toplama sistemi ile bilgisayar ortamına aktarılmaktadır. Elde edilen değerler sonucunda, geliştirilen algoritma ile sürücü ekranında sürücü postür analizi ile ilgili sonuçlar görüntülenir.



Şekil 2.1 Direksiyon sistemi

Bu çalışmada, Şekil 2.2’de gösterilen kuvvet/tork sensörü, direksiyon simidinin merkezinde yerleştirilmiş olup, sürücü tarafından direksiyona uygulanan kuvvet ve tork değerlerini ölçmek için kullanılmıştır.

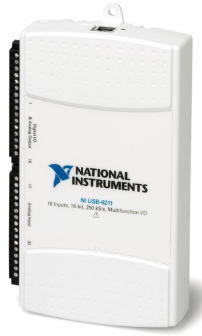


Şekil 2.2 ATI-Delta S-330-30, kuvvet/tork sensörü ve kontrolörü [27]

ATI-Delta kuvvet/tork sensörü, gerçek zamanlı kuvvet ve tork değerlerini yüksek hassasiyet ile ölçme yeteneğine sahiptir [27]. X ve Y eksenlerinde 330 N, Z ekseninde

990 N değerine kadar kuvvet ölçebilmektedir. Sensör verileri kendi kontrolör sistemi (F/T controller, Şekil 2.2) yolu ile veri toplama sistemine (DAQ) analog olarak iletilir. Kontrolör dijital çıkış verme özelliğine de sahiptir.

Kuvvet/tork sensöründen gelen değerlerin bilgisayar ortamına akatarılması Şekil 2.3’de gösterilen veri toplama sistemi ile sağlanmıştır. Deney düzeneğinde NI-USB 6211 veri toplama sistemi kullanılmıştır.



Şekil 2.3 NI-USB 6211, veri toplama sistemi [Ni]

NI USB - 6211 çok işlevli bir veri toplama sistemidir. Analog giriş, analog çıkış, dijital giriş, dijital çıkış ve sayıcı modülü bulunmaktadır [Ni]. MATLAB/Simulink ile uyumlu çalışan bu cihazın en önemli avantajlarından biri gerçek zamanlı veri aktarım kabiliyetinin olmasıdır. Aynı zamanda, Şekil 2.4’de görülen enkoder tarafından alınan direksiyon açısı bilgileri, bilgisayar ortamına veri toplama sistemi sayesinde aktarılmaktadır. Deney düzeneğinde Kübler artımsal enkoder kullanılmıştır.

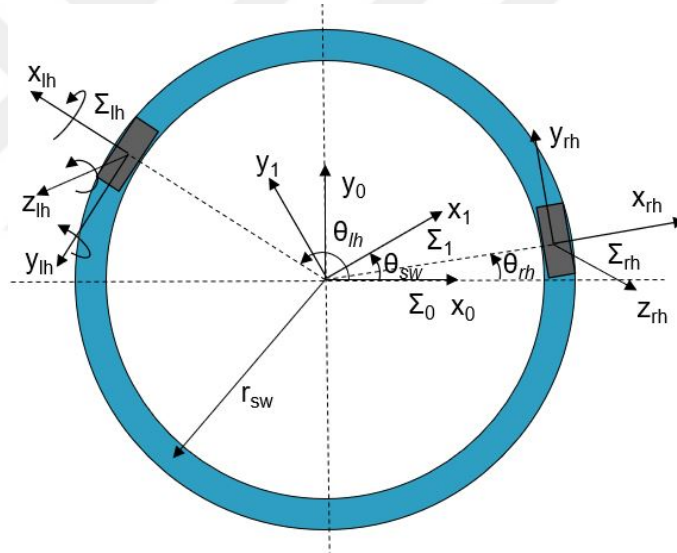


Şekil 2.4 Kübler, artımsal enkoder [28]

Artımsal enkoder, bir milin açısal hareketini veya konumunu tanımlamak için, hareketi analog veya dijital veriye dönüştüren bir tür kodlayıcı cihazdır. Bir hareketin hızı, periyodu ölçerek veya zaman birimi başına darbeyi (pulse) sayarak belirlenir. Bu tez çalışmasında kullanılan Kübler enkoder tam tur döndüğünde 2500 pulse üretmektedir ve 5÷30V DC besleme gerilimi ile çalışmaktadır [28].

2.2.2 Sürücü Sistemi Analiz Algoritmaları

Sistemde sürücü tarafından direksiyona uygulanan kuvvetlerin analizi için oluşturulan direksiyon sisteminin serbest cisim diyagramı Şekil 2.5’de gösterilmektedir. Serbest cisim diyagramında, iki adet siyah dikdörtgen kutu ile gösterilen bölgeler sürücünün sağ ve sol elini temsil etmektedir. Sistemde dört adet Kartezyen eksen takımı (koordinat sistemi) bulunmaktadır. İlk eksen takımı sistemin merkezinde olup orjin eksen takımı olarak tanımlanmaktadır. Orjin eksen takımı X_0, Y_0 ve Z_0 eksenlerinden oluşmaktadır ve e_0 olarak isimlendirilmektedir. Sürücü tarafından direksiyonun θ_{sw} kadar döndürülmesi ile ikinci eksen takımı X_1, Y_1 ve Z_1 eksenleri ile tanımlanmaktadır ve bu eksen takımı e_1 olarak isimlendirilmektedir. Sürücünün sol el ile temas ettiği nokta üçüncü eksen takımı X_{lh}, Y_{lh} ve Z_{lh} , sağ el ile temas ettiği nokta dördüncü eksen takımı X_{rh}, Y_{rh} ve Z_{rh} olarak tanımlanmaktadır. Üçüncü eksen takımı e_{lh} olarak isimlendirilirken, dördüncü eksen takımı e_{rh} olarak isimlendirilmektedir. Eksen takımlarının yönleri e_0 eksen takımına göre sağ el kuralı ile hesaplanmaktadır. Serbest cisim diyagramında belirtilen r_{sw} direksiyon simidinin yarıçapını göstermektedir.



Şekil 2.5 Direksiyon sisteminin serbest cisim diyagramı

Bölüm 2.2.1’de bahsedildiği gibi, kuvvet/tork sensörü direksiyon simidinin merkezinde bulunmakta ve sürücü tarafından direksiyona uygulanan kuvvetler e_0 eksen takımında ölçülmektedir. Sürüş esnasında, sürücünün sergilediği davranışlar sonucunda direksiyon açısı değişmektedir. Direksiyon açısının değişimi θ_{sw} ile tanımlanmaktadır ve ölçülen kuvvet/tork değerleri e_1 eksen takımına aktarılır. Ölçülen kuvvet/tork değerleri, e_1 eksen takımında çarpık-simetrik (skew-symmetric) matris biçiminde Denklem 2.1’de ifade edilmektedir.

$$Phi_1 = \begin{bmatrix} 0 & -m_z & m_y & f_x \\ m_z & 0 & -m_x & f_y \\ -m_y & m_x & 0 & f_z \\ -f_x & -f_y & -f_z & 0 \end{bmatrix} \quad (2.1)$$

Phi_1 , kuvvet/tork sensörü tarafından ölçülen değerlerin çarpık-simetrik matrisini ifade etmektedir. Bu matriste, f_x ve m_x e_1 eksen takımının X eksenindeki ölçülen kuvvet ve tork değerlerini temsil eder. Aynı şekilde, f_y ve m_y , Y eksenindeki ölçülen kuvvet ve tork değerlerini temsil eder. Z ekseninde ölçülen kuvvet ve tork değerleri de f_z ve m_z ile gösterilir.

Denklem 2.2’de ifade edilen Phi_{1lh} ve Denklem 2.3’de ifade edilen Phi_{1rh} bu sistemde hesaplanması istenilen değerlerdir. Phi_{1lh} ve Phi_{1rh} sol (left) ve sağ (right) elin temas ettiği noktada oluşan kuvvet ve tork değerlerinin çarpık-simetrik matrisini gösterir.

$$Phi_{1lh} = \begin{bmatrix} 0 & -m_{zlh} & m_{ylh} & m_{xlh} \\ m_{zlh} & 0 & -m_{xlh} & f_{ylh} \\ -m_{ylh} & m_{xlh} & 0 & f_{zlh} \\ -f_{xlh} & -f_{ylh} & -f_{zlh} & 0 \end{bmatrix} \quad (2.2)$$

$$Phi_{1rh} = \begin{bmatrix} 0 & -m_{zrh} & m_{yrh} & m_{xrh} \\ m_{zrh} & 0 & -m_{xrh} & f_{yrh} \\ -m_{yrh} & m_{xrh} & 0 & f_{zrh} \\ -f_{xrh} & -f_{yrh} & -f_{zrh} & 0 \end{bmatrix} \quad (2.3)$$

Denklem 2.2’de ifade edilen matriste, f_{xlh} , f_{ylh} , f_{zlh} , m_{xlh} , m_{ylh} ve m_{zlh} sol elin e_{lh} eksen takımında oluşturduğu X, Y ve Z eksenlerindeki kuvvet ve tork değerlerini gösterir. Aynı şekilde, Denklem 2.3’deki matriste yer alan, f_{xrh} , f_{yrh} , f_{zrh} , m_{xrh} , m_{yrh} ve m_{zrh} sağ elin e_{rh} eksen takımında oluşturduğu X, Y ve Z eksenlerindeki kuvvet ve tork değerlerini gösterir.

Phi_{1lh} ve Phi_{1rh} çözümlerine yönelik algoritmaların analitik analizi bazı varsayımlar gerektirir. İlk olarak, bu çalışmada tek el ile sürüş durumu ele alınmıştır. Bu doğrultuda, analitik analiz bu çalışmada sürücünün aracı sadece sol el ile kullanıldığı varsayılarak yapılmıştır ($Phi_{1rh} = 0$). Bir diğer varsayım, bu çalışmada geliştirilen algoritma sadece uzun yolculuk senaryosu dikkate alarak geliştirilmiştir. Şehir içinde

manevraların çok fazla olduğu ve aracın çok fazla durup hareket ettiği durumlar için bu çalışma göz önüne alınmamıştır. Bu ana varsayımlara göre, başlangıçta e_0 eksen takımı ile e_{lh} eksen takımı değerleri arasındaki ilişkinin tanımlanması gerekmektedir. e_0 eksen takımının, e_{lh} eksen takımına göre dönme matrisi Denklem 2.4’de ifade edilmektedir.

$$R_{1lh} = \begin{bmatrix} \cos(\Delta\theta_{lh}) & -\sin(\Delta\theta_{lh}) & 0 & 0 \\ \sin(\Delta\theta_{lh}) & \cos(\Delta\theta_{lh}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

θ_{lh} değeri sürücünün sol el konum açısını temsil eder ve e_0 eksen takımı ile e_{lh} eksen takımı arasındaki açı olarak tanımlanmıştır (Denklem 2.5).

$$\Delta\theta_{lh} = \theta_{lh} - \theta_{sw} \quad (2.5)$$

Denklem 2.6 e_0 eksen takımının, e_{lh} eksen takımına göre öteleme matrisini ifade etmektedir ve T_{1lh} ile ifade edilir.

$$T_{1lh} = \begin{bmatrix} 1 & 0 & 0 & \cos(\Delta\theta_{lh})r_{sw} \\ 0 & 1 & 0 & \sin(\Delta\theta_{lh})r_{sw} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

Denklem 2.6 ve Denklem 2.5 sonucunda, e_0 eksen takımı ile e_{lh} eksen takımı arasındaki dönüşüm matrisi Denklem 2.7’de gösterilmektedir ve H_{1lh} ile ifade edilir.

$$H_{1lh} = \begin{bmatrix} \cos(\Delta\theta_{lh}) & -\sin(\Delta\theta_{lh}) & 0 & \cos(\Delta\theta_{lh})r_{sw} \\ \sin(\Delta\theta_{lh}) & \cos(\Delta\theta_{lh}) & 0 & \sin(\Delta\theta_{lh})r_{sw} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

Bir tensör matrisinin kartezyen bileşenlerinin dönüşüm yasasından Φ_1 ölçülen değer matrisi ve Φ_{lh} hesaplanması gereken sol el değerler matrisi arasındaki ilişki Denklem 2.8 ile ifade edilmektedir [29].

$$Phi_1 = H_{1lh} \cdot Phi_{lh} \cdot H'_{1lh} \quad (2.8)$$

Denklem 2.8 incelendiğinde, eşitliğin sol tarafında 7 adet ölçülen değer ($f_x, f_y, f_z, m_x, m_y, m_z, \theta_{sw}$) sağ tarafında ise 7 adet bilinmeyen değer bulunmaktadır ($f_{xlh}, f_{ylh}, f_{zlh}, m_{xlh}, m_{ylh}, m_{zlh}, \theta_{lh}$). Denklem 2.8 hesaplandığında 7 bilinmeyen değeri olan 6 denklem ortaya çıkmaktadır. Bu denklemin çözüme ulaşabilmesi için bir varsayıma daha ihtiyaç duyulmaktadır. Sürücü, normal sürüş standartlarında, sürüş esnasında direksiyon simidine hiçbir zaman z ekseninde tork uygulamadığı varsayılmaktadır ve bu varsayım Denklem 2.9 'de ifade edilir.

$$m_{zlh} = 0 \quad (2.9)$$

Yapılan bu varsayım sonucunda 6 bilinmeyenli 6 denklem ortaya çıkmaktadır. Buna göre, Denklem 2.8 yapılan bu varsayım ile çözüldüğünde, Denklem 2.10, 2.11, 2.12 ve Denklem 2.13 ile verilen çözüm matrisi sonuçları elde edilir. Bu çözüm matrisinin sonuçları P ile temsil edilir.

$$P(1,:) = \begin{bmatrix} 0 \\ f_{ylh} \cdot r_{sw} \\ f_{zlh} \cdot r_{sw} \cdot \cos(\Delta\theta_{lh}) - m_{ylh} \cdot \cos(\Delta\theta_{lh}) - m_{xlh} \cdot \sin(\Delta\theta_{lh}) \\ f_{ylh} \cdot \sin(\Delta\theta_{lh}) - f_{xlh} \cdot \cos(\Delta\theta_{lh}) \end{bmatrix} \quad (2.10)$$

$$P(2,:) = \begin{bmatrix} f_{ylh} \cdot r_{sw} \\ 0 \\ f_{zlh} \cdot r_{sw} \cdot \sin(\Delta\theta_{lh}) - m_{ylh} \cdot \sin(\Delta\theta_{lh}) - m_{xlh} \cdot \cos(\Delta\theta_{lh}) \\ -f_{ylh} \cdot \cos(\Delta\theta_{lh}) - f_{xlh} \cdot \sin(\Delta\theta_{lh}) \end{bmatrix} \quad (2.11)$$

$$P(3,:) = \begin{bmatrix} -f_{zlh} \cdot r_{sw} \cdot \cos(\Delta\theta_{lh}) + m_{ylh} \cdot \cos(\Delta\theta_{lh}) + m_{xlh} \cdot \sin(\Delta\theta_{lh}) \\ -f_{zlh} \cdot r_{sw} \cdot \sin(\Delta\theta_{lh}) + m_{ylh} \cdot \sin(\Delta\theta_{lh}) - m_{xlh} \cdot \cos(\Delta\theta_{lh}) \\ 0 \\ -f_{zlh} \end{bmatrix} \quad (2.12)$$

$$P(4,:) = \begin{bmatrix} f_{xlh} \cdot \cos(\Delta\theta_{lh}) - f_{ylh} \cdot \sin(\Delta\theta_{lh}) + m_{xlh} \\ f_{ylh} \cdot \cos(\Delta\theta_{lh}) + f_{xlh} \cdot \sin(\Delta\theta_{lh}) - m_{xlh} \\ f_{zlh} \\ 0 \end{bmatrix} \quad (2.13)$$

Yukarıda bahsedilen çözüm matrislerinin MATLAB ortamında hesaplanması ile Denklem 2.14, 2.15, 2.16, 2.17, 2.18, 2.19, 2.20, 2.21, 2.22 ve Denklem 2.23’de bilinmeyen değerlere ait çözüm denklemleri gösterilmektedir. Bu çözüm denklemlerinde, d , A , b ve x değerleri ara değerlerdir. Bu ara değerler, çözüm denkleminin anlaşılabilir olması için kullanılmıştır.

$$f_{ylh} = \frac{m_z}{r_{sw}} \quad (2.14)$$

$$f_{xlh} = \sqrt{f_x^2 + f_y^2 - f_{ylh}^2} \quad (2.15)$$

$$d = \frac{\begin{bmatrix} f_{xlh} & -f_{ylh} \\ f_{xlh} & f_{ylh} \end{bmatrix}}{\begin{bmatrix} f_x \\ f_y \end{bmatrix}} \quad (2.16)$$

$$\Delta\theta_{lh} = \text{atan2}(d(2), d(1)) \quad (2.17)$$

$$f_{zlh} = f_z \quad (2.18)$$

$$A = \begin{bmatrix} \sin(\Delta\theta_{lh}) & \cos(\Delta\theta_{lh}) \\ \cos(\Delta\theta_{lh}) & \sin(\Delta\theta_{lh}) \end{bmatrix} \quad (2.19)$$

$$b = \begin{bmatrix} m_y + r_{sw} \cdot \cos(\Delta\theta_{lh}) \\ -m_x + r_{sw} \cdot \sin(\Delta\theta_{lh}) \end{bmatrix} \quad (2.20)$$

$$x = \left(\frac{A}{b}\right)' \quad (2.21)$$

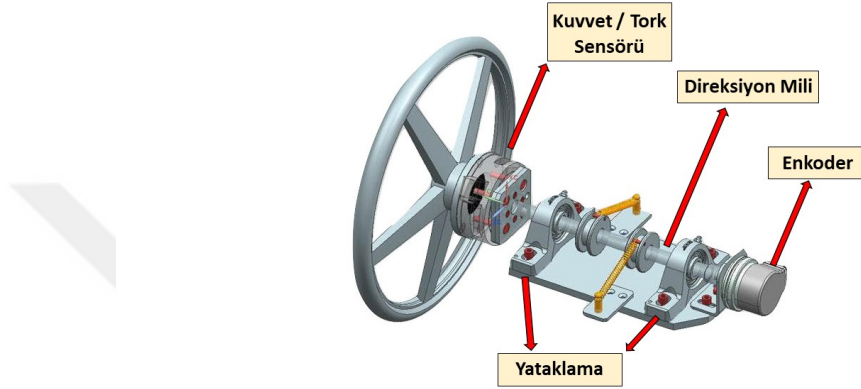
$$m_{xlh} = x(1) \quad (2.22)$$

$$m_{ylh} = x(2) \quad (2.23)$$

Elde edilen sonuç denklemleri ile sürücünün kuvvet, tork ve direksiyon açısı değerleri ölçülerek, sürücünün sürüş esnasında direksiyonu nereden tuttuğu sonucuna varılır.

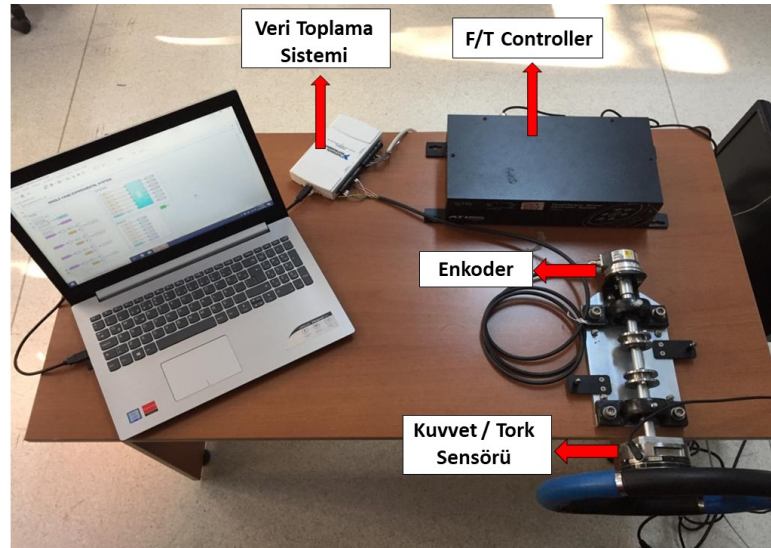
2.3 Deney Düzeneği

Geliştirilen direksiyon sisteminin katı modeli Şekil 2.6 'de deney düzeneği ise Şekil 2.7 görülmektedir.



Şekil 2.6 Direksiyon sistemi

Direksiyon sisteminin 3D çizimi incelendiğinde Şekil 2.1'de de görüldüğü üzere, direksiyon simidinin merkezinde bir adet kuvvet/tork sensörü ve direksiyon milinin sonunda bir adet enkoder bulunmaktadır.



Şekil 2.7 Deney düzeneği

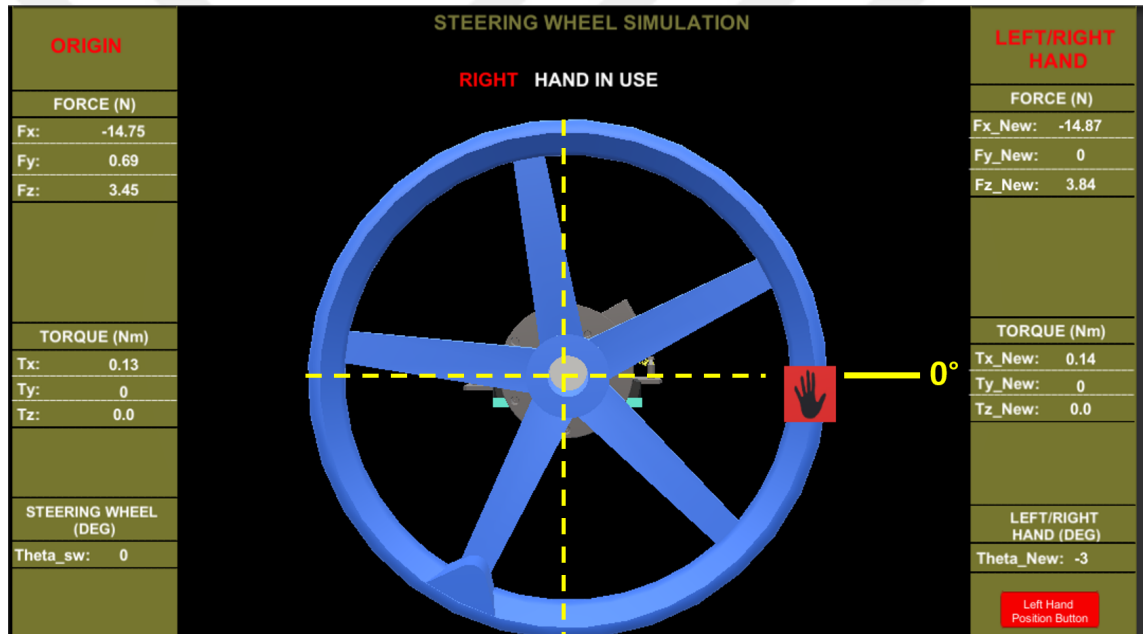
Aynı zamanda direksiyon sistemi üzerine deneylerin yapılabilmesi için direksiyon

sistemi sabitlenmiştir. Bunun için 3D çizimde görüldüğü üzere deney düzeneğinin masaya monte edilebilmesi için uygun yataklama yapılmıştır.

2.4 Sistem Arayüzü ve Haberleşme Protokolü

2.4.1 Sistem Arayüzü

Unity programı masaüstü, mobil, web ve konsollar için kolay bir şekilde 3D oyunlar ve uygulamalar oluşturmaya yarayan C# tabanlı çalışan Visual Studio'ya ait bir uygulamadır ve günümüzde oyun sektörü, insansız hava aracı ve otomotiv simülasyonları gibi yoğun olarak birçok alanda kullanılmaktadır. Solidworks ile oluşturulan 3D direksiyon deney sistemi katı modeli Unity ortamına kolayca aktarılabilir. Şekil 2.8'de direksiyon deney sisteminin arayüzü gösterilmiştir.

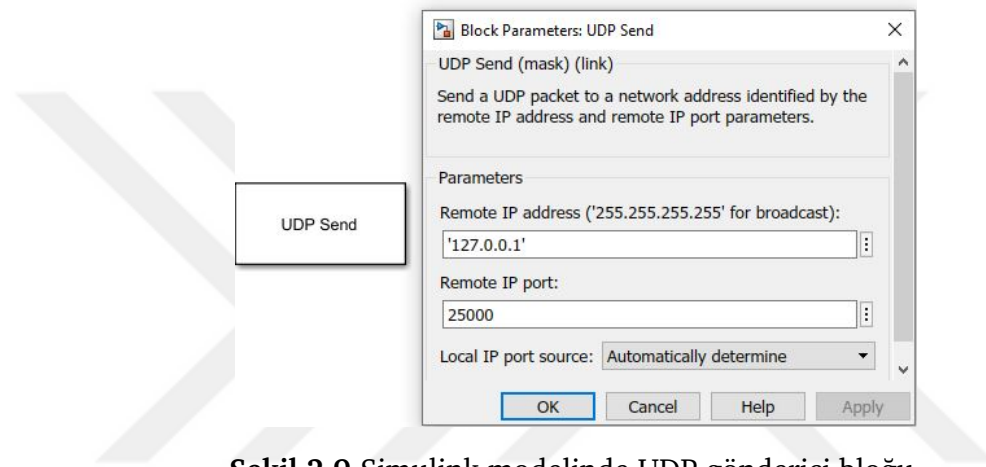


Şekil 2.8 Unity arayüzü

Arayüzün sol tarafında bulunan çerçevede, kuvvet/tork sensörü tarafından e_0 eksen takımında ölçülen kuvvet ve tork değerleri ile enkoder tarafından ölçülen direksiyon açısı verilmektedir. Arayüzün sağ tarafındaki çerçevede Simulink ortamında hesaplanan, sağ veya sol elin temas ettiği noktada ortaya çıkan kuvvet ve tork değerleri verilmektedir. Aynı zamanda, elin temas ettiği noktaya ait konum açısı arayüzün sağ tarafında gösterilmektedir ve direksiyon üzerinde kırmızı kutu içerisine alınmış el simgesi ile temsil edilmektedir. Son olarak, arayüzün üst kısmında bir mesaj görülmektedir. Bu yazı seçilen moda göre sürücüye farklı uyarılarda bulunmaktadır. Örnek olarak, Şekil 2.8'de görülen mesaj sürücünün o an direksiyonu hangi el ile kullandığına dair bilgi vermektedir. Geliştirilen arayüz ile sürüş esnasında sürücünün davranışları gerçek zamanlı ifade edilebilmektedir.

2.4.2 Haberleşme Protokolü

Unity simülasyonu MATLAB ile haberleşmesini UDP protokolü ile gerçekleştirir. UDP, ağ üzerinden bir cihazdan diğerine **datagram** olarak da adlandırılan bir veri birimini almak için internet protokolünü kullanan haberleşme yöntemidir. MATLAB ile birlikte kullanıldığı durumda ilk olarak gönderici ve alıcı tarafların belirlenmesi gerekir. Bu çalışmada, MATLAB/Simulink üzerinden hesaplanan değerler Unity simülasyonuna gönderilmektedir. Böylelikle, Simulink gönderici taraf olurken, Unity alıcı taraf olarak tanımlanmaktadır. Simulink gönderici olduğu durumlarda, Şekil 2.9'de görülen "UDP Send" bloğu kullanılmaktadır. Bu blok Simulink'ten Unity'ye veri aktarmak için kullanılır. Blok içinde girilen IP adresi ve IP portu alıcı tarafı ile eşleşmelidir.



Şekil 2.9 Simulink modelinde UDP gönderici bloğu

Unity simülasyonuna yollanan kuvvet, moment, direksiyon açısı ve el konum bilgileri "UDP Send" bloğuna "Byte Packing" (bayt paketleme) bloğunu kullanarak verilir. Bayt şeklinde yollanılan veriler Unity simülasyonunda çözümlenerek kullanılır. MATLAB tarafından gönderici ayarları tamamlandıktan sonra, Unity simülasyonunda alıcı ayarları yapılır. Unity bahsedildiği üzere C# tabanlı çalışan bir programdır ve bu çalışmada alıcı oluşturmak için `UdpClient` komutu kullanılmıştır. Gönderici tarafından belirlenen IP adres ayarları (127.0.0.1) ve IP port ayarları (25000) aynı şekilde Unity programında tanımlanır ve simülasyonda haberleşme için gerekli yapılandırma tamamlanır. Gerekli ayarlar tamamlandıktan sonra UDP haberleşme ile Simulink modelinde hesaplanan ve ölçülen değerler gerçek zamanlı bir şekilde Unity programına gönderilir.

3.1 MATLAB ile Makine Öğrenmesi Algoritması

Bu tez çalışmasında, sürücünün direksiyonu hangi el ile sürdüğünü tahmin eden bir makine öğrenmesi algoritması geliştirilmiştir. Makine öğrenmesi yöntemleri sınıflandırma problemlerinde etkin kullanılmaktadır. MATLAB'in sınıflandırma yöntemi için oluşturmuş olduğu "Classification Learner" uygulaması bulunmaktadır. "Classification Learner" verileri sınıflandırmak için modelleri eğiten bir uygulamadır. Modelin eğitilebilmesi için, sisteme ait giriş ve çıkış verilerinin gösterilmesi gerekir. Makine öğrenmesi algoritmasının yüksek doğrulukta tahmin yapabilmesi için verilerin çok hassas bir şekilde elde edilmesi ve işlenmesi gerekmektedir. Bu çalışmada, kullanılan kuvvet/tork sensörü ve enkoder ile veriler deney sistemi üzerinden elde edilmiştir. Alınan veriler, eğitim veri setleri ve deneme veri setleri olarak ikiye ayrılmıştır. Eğitim veri setleri, algoritmanın öğrenmesi için kullanılmış olup, deneme veri setleri ise algoritmanın doğruluğunu ölçmek için kullanılmıştır. Bu durumda, sistemin çıkışları "Sağ" ve "Sol" olarak tanımlanmış olup, sistemin girişleri sensörler tarafından birçok farklı sürüş senaryoları ile elde edilmiştir.

3.1.1 Veri Toplama

Makine öğrenmesinde doğru veri toplama ve işleme, algoritmanın çıktıları açısından kritiktir. Doğru veri işleme, gerçekleştirilecek senaryoları doğru modelleme ile paralellik arz eder. Bu çalışmada verilerin en doğru şekilde elde edilebilmesi için sürüş esnasında ortaya çıkabilecek olan farklı senaryolar oluşturulmuş ve detayları aşağıda verilmiştir:

- i. Direksiyonun sağ el ile sürülmesi durumunda, direksiyon simidinin 0 derecede tutulması. Sağ el bu konumdayken, sağ viraja 5, 10, 15, 20, 25, 30, 35, 40 ve 45 derece ile ayrı ayrı dönüş sağlandığı durumlar.
- ii. Direksiyonun sağ el ile sürülmesi durumunda, direksiyon simidinin 45 derecede

- tutulması. Sağ el bu konumdayken, sağ viraja 5, 10, 15, 20, 25, 30, 35, 40 ve 45 derece ile ayrı ayrı dönüş sağlandığı durumlar.
- iii. Direksiyonun sağ el ile sürülmesi durumunda, direksiyon simidinin 90 derecede tutulması. Sağ el bu konumdayken, sağ viraja 5, 10, 15, 20, 25, 30, 35, 40 ve 45 derece ile ayrı ayrı dönüş sağlandığı durumlar.
- iv. Direksiyonun sağ el ile sürülmesi durumunda, direksiyon simidinin 135 derecede tutulması. Sağ el bu konumdayken, sağ viraja 5, 10, 15, 20, 25, 30, 35, 40 ve 45 derece ile ayrı ayrı dönüş sağlandığı durumlar.
- v. Direksiyonun sağ el ile sürülmesi durumunda, direksiyon simidinin 180 derecede tutulması. Sağ el bu konumdayken, sağ viraja 5, 10, 15, 20, 25, 30, 35, 40 ve 45 derece ile ayrı ayrı dönüş sağlandığı durumlar.
- vi. Direksiyonun sağ el ile sürülmesi durumunda, direksiyon simidinin 225 derecede tutulması. Sağ el bu konumdayken, sağ viraja 5, 10, 15, 20, 25, 30, 35, 40 ve 45 derece ile ayrı ayrı dönüş sağlandığı durumlar.
- vii. Direksiyonun sağ el ile sürülmesi durumunda, direksiyon simidinin 270 derecede tutulması. Sağ el bu konumdayken, sağ viraja 5, 10, 15, 20, 25, 30, 35, 40 ve 45 derece ile ayrı ayrı dönüş sağlandığı durumlar.

Bahsedilen senaryolar direksiyon simidini sağ el ile tutup sağ viraja dönüş yapıldığı durumları içermektedir. Aynı şekilde yukarıda bahsedilen senaryoların sağ el ile sol viraja dönüş sağlama durumları da oluşturulmuştur. Bu durumda, sağ el ile ortaya çıkabilecek olan tüm sürüş senaryoları tamamlanmış olup, sağ el verileri elde edilmiştir. Aynı şekilde bütün senaryolar sol el için de tekrar edilmiştir. Sonuç olarak, toplam 28 senaryo ile sürücü davranışlarının tahminine yönelik en olası senaryolar ele alınmıştır.

3.1.2 Verilerin Önişlenmesi

Makine öğrenmesi dört aşamadan oluşmakta olup, ikinci aşaması verilerin önişlenmesidir. "Classification Learner" uygulamasında yüksek tahmin doğruluğu elde edebilmek için verilerin önişleme sürecinden geçirilmesi gerekir. Bu çalışmada, sistemin çıktıları "Sağ" veya "Sol" olarak tanımlanmıştır, girdileri ise farklı senaryolardan elde edilen $f_x, f_y, f_z, m_x, m_y, \theta_{sw}$ değerler olarak ele alınmıştır. Önişlenen verilerin tamamı tek bir tabloda toplanmış olup, gelecek tahmini tablosu olarak ifade edilmiştir.

Tablo 3.1 Gelecek tahmin tablosu

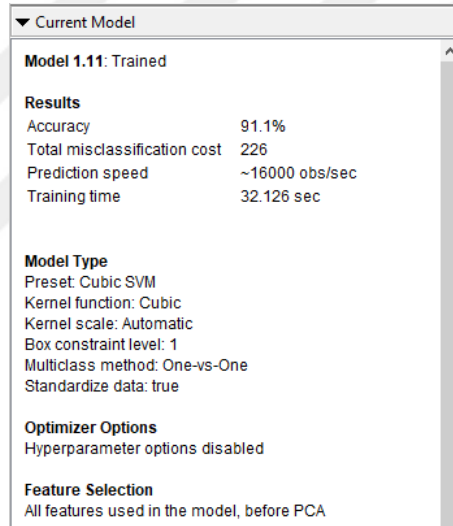
t	f_x	f_y	f_z	m_x	m_y	θ_{sw}	Sonuç
163.8	-30.2400	0.1859	0.1037	-30.24	-30.24	0	'Sağ'
164.0	-30.5280	2.3489	2.2234	-30.5280	-26.2080	4.3200	'Sağ'
164.2	-25.7760	8.6488	7.3286	-25.7760	-14.0040	11.7720	'Sağ'
164.4	-0.5760	0.5924	0.5166	-1.4400	-0.5400	0.9000	'Sağ'
164.6	-1.0080	0.3941	0.2788	-1.0080	-0.8640	0.1440	'Sağ'
164.8	-18.4320	13.6051	12.3914	-32.4720	-2.1240	30.3480	'Sağ'
165.0	-35.4240	0.4610	0.3940	-35.8560	-35.1000	0.7560	'Sağ'
165.2	-40.6080	2.4983	2.2994	-40.6080	-35.8200	4.7880	'Sağ'
165.4	-40.3200	0.3125	0.2627	-40.4640	-40.0320	0.4320	'Sağ'
165.6	-36.1440	1.8771	1.6004	-39.0240	-35.7120	3.3120	'Sağ'
165.8	-8.6400	15.0719	13.8516	-35.4600	-3.0600	32.4000	'Sağ'
166.0	-0.4320	5.1735	3.0966	-0.4320	-0.2880	0.1440	'Sağ'
166.2	-44.9280	5.9689	4.6112	-45.0720	-39.3120	5.7600	'Sağ'
166.4	-45.5040	0.2870	0.1829	-45.6480	-45.5040	0.1440	'Sağ'
166.6	-30.5280	9.1521	8.3267	-44.7840	-26.1360	18.6480	'Sağ'
166.8	-6.0480	6.5680	5.2255	-10.5480	-2.2680	8.2800	'Sağ'
167.0	-1.8720	0.9908	0.7294	-2.5920	-1.6920	0.9000	'Sol'
167.2	-1.1520	0.5028	0.4470	-1.2960	-0.1440	1.1520	'Sol'
167.4	-1.2960	0.5438	0.3935	-1.4400	-1.0080	0.4320	'Sol'
167.6	-1.5840	0.9759	0.8032	-2.5920	-1.3680	1.2240	'Sol'
167.8	0.1440	0.4685	0.3364	-0.1800	0.3240	0.5040	'Sol'
167.8	0.5760	0.8537	0.6820	0.1440	1.2600	1.1160	'Sol'
168.0	2.4480	0.7772	0.4884	2.4480	2.5920	0.1440	'Sol'
168.2	2.4480	0.5750	0.4226	1.9800	2.6280	0.6480	'Sol'

Tablo 3.1'de oluşturulan gelecek tahmini matris tablosunun bir kısmı görülmektedir. Gelecek tahmin matris tablosu tamamlandıktan sonra öngörülü model geliştirilmiştir. Bu çalışmada 43 sütunluk bir tablo oluşturulmuş olup, 43 nolu sütun, Tablo 3.1'de görüldüğü üzere "Sağ" veya "Sol" olmak üzere sonucu göstermektedir. Diğer 42 sütun, deney düzeneğinde yapılan sürüş senaryolarında $f_x, f_y, f_z, m_x, m_y, \theta_{sw}$ verilerinin her birinden elde edilen 7 farklı istatistiksel değeri göstermektedir (ortalama, medyan, standart sapma, ortalama mutlak sapma, birinci çeyreklik, üçüncü çeyreklik ve çeyrekler açıklığı). Bu doğrultuda, 7 girdi değeri yerine 42 adet girdi oluşturularak, daha fazla girdi ile tahmin doğruluğunun artırılması amaçlanmış ve verilerin önışlemesi tamamlanmıştır. (Mevcut çalışmada sürücü tarafından z ekseninde direksiyona moment uygulanmadığı varsayımı yapıldığından, m_z değeri makine öğrenmesi tahmin algoritmasında yer almamıştır.)

3.1.3 Öngörülü Modelin Geliştirilmesi

"Classification Learner" uygulaması makine öğrenmesi algoritmasını gerçekleştirebilmek için girdi olarak gelecek tahmin matris tablosuna ihtiyaç duyar. Verilen tablo sonucunda, uygulama "Train" komutunu kullanarak 25 farklı metod ile makine öğrenmesi tahmin algoritmasını geliştirir ve bütün metodların doğruluk

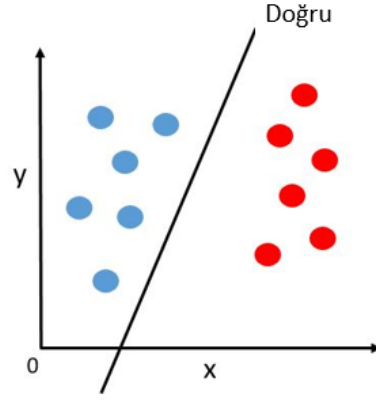
oranını gösterir. Yapılan çalışmada, Şekil 3.1 'de görüldüğü üzere, %91.1 doğruluk oranı ile en yüksek tahmin algoritması kübik destek vektör makinesi (Support Vector Machine — SVM) yöntemi kullanılarak elde edilmiştir. Tablo 3.2 'de diğer yöntemlerden elde edilen doğruluk oranları gösterilmektedir. "Classification Learner" uygulamasından elde edilen en yüksek doğruluk oranlı yöntemin projede kullanılması için aynı zamanda *kesinlik* (precision), *duyarlılık* (recall), ve F_1 ölçütü sonuçlarının da incelenmesi gerekir. *Kesinlik* değeri, tekrarlanan tahminler arasında tahminlerin birbirine ne kadar yakın olduğunu göstermektedir ve bu çalışmada 0.9598 olarak hesaplanmıştır. *Duyarlılık* değeri, pozitif olarak tahmin edilmesi gereken işlemlerin ne kadarının pozitif olarak tahmin edildiğini gösteren bir metriktir ve bu çalışmada 0.9783 olarak hesaplanmıştır. F_1 ölçüt değeri ise, kesinlik ve duyarlılık değerlerinin harmonik ortalamasını göstermektedir ve bu çalışmada 0.9697 olarak hesaplanmıştır. Elde edilen *kesinlik*, *duyarlılık*, F_1 ölçütü değerlerinin ve doğruluk oranının yüksek olması kübik SVM yönteminin bu çalışma için uygun olduğunu göstermektedir.



Şekil 3.1 Classification learner SVM doğruluk oranı

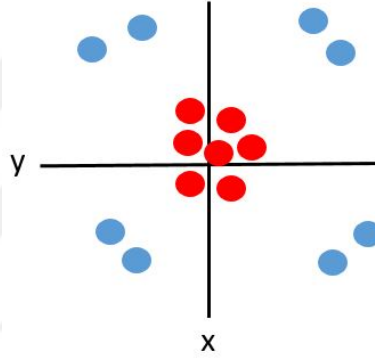
Bu tez çalışmasında kullanılan destek vektör makinesi yöntemi, iki gruplu sınıflandırma problemleri için sınıflandırma algoritmalarını kullanan denetimli bir makine öğrenme modelidir. Destek vektör makinesi, eğitim veri setini alarak oluşturduğu model ile daha önce karşılaşmadığı senaryolar hakkında tahmin yapabilir [30]. Destek vektör makinelerinin nasıl çalıştığı ve temel prensipleri ile ilgili örnek bir çalışma aşağıda gösterilmiştir.

Örnek olarak, bir koordinat sisteminde, kırmızı ve mavi olmak üzere iki adet sınıflandırma yapılması istenmektedir. İlk olarak mevcut örnek için toplanılan mavi ve kırmızı renklere ait eğitim verileri Şekil 3.2'de görüldüğü üzere bir düzleme yerleştirilir. Bir destek vektör makinesi bu veri noktalarını alır ve etiketleri en iyi ayıran doğruyu çıkarır. Bu doğru karar sınırıdır: çizginin bir tarafında düşen veriler



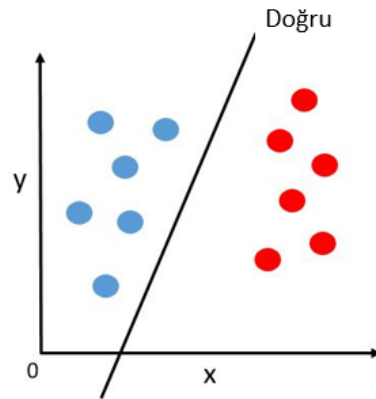
Şekil 3.2 Destek vektör makinesi 2D hiper düzlem örneği

mavi, diğer tarafında kalanlar ise kırmızı olarak sınıflandırılır. Bu yöntem doğrusal destek vektör makinesi yöntemi için geçerlidir [30].



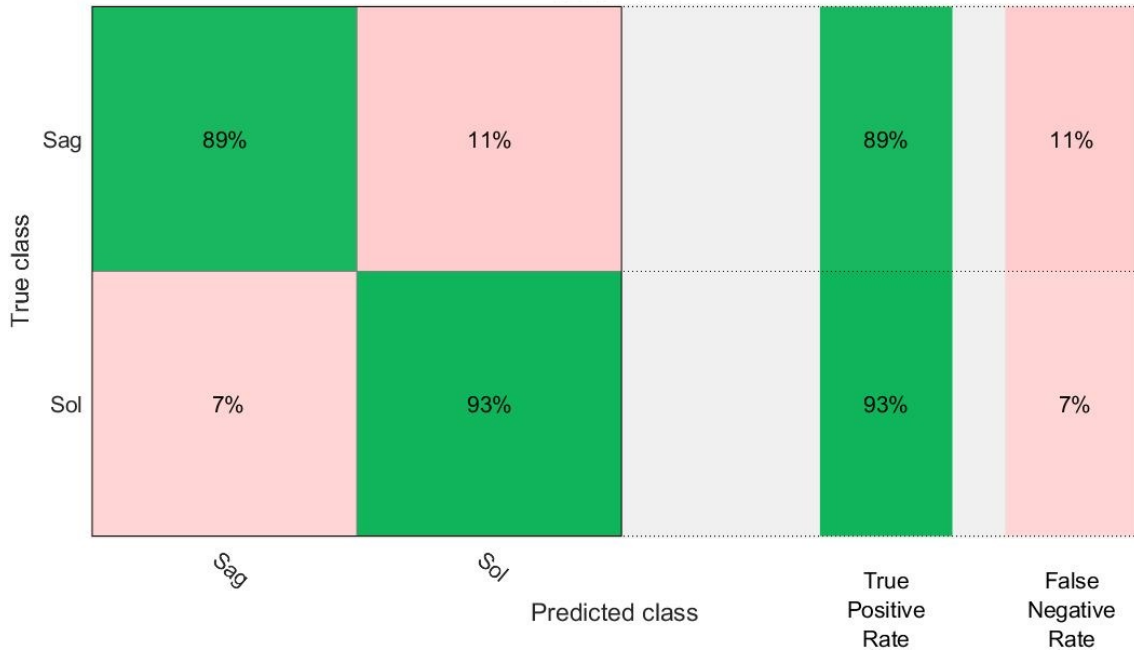
Şekil 3.3 Destek vektör makinesi 3D örneği

Şekil 3.3’de doğrusal olmayan eğitim verilerini içeren bir örnek görülmektedir. Doğrusal olmayan durumlarda üçüncü boyut (z eksen) eklemesi yapılır. Böylelikle üç boyutlu bir alan elde edilir ve Şekil 3.4’de bu alana ait görüntü görülmektedir. Bu doğrultuda, z ve x eksenleri arasında aynı şekilde kolay bir biçimde bir doğru belirlenir ve sınıflandırma yapılır [30].



Şekil 3.4 Destek vektör makinesi 3D hiper düzlem örneği

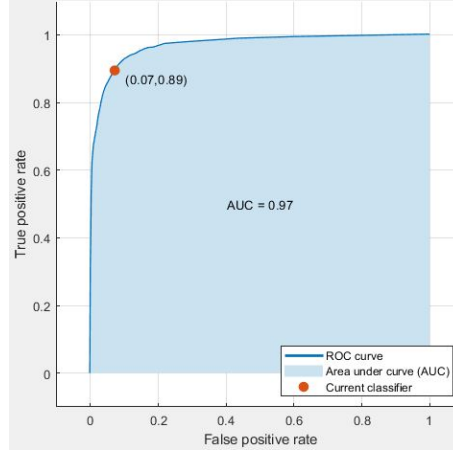
Yukarıda ifade edilen ve bu tez çalışmasında en yüksek doğruluk oranı veren destek vektör makinesi yöntemi ile geliştirilen tahmin algoritmasının hata matrisi Şekil 3.5’de görülmektedir. Aracın sağ el ile sürüldüğü durumlarda algoritmanın sağ el doğru tespit oranı %89 iken, sol el ile sürüldüğü durumlarda algoritmanın sol el doğru tespit oranı %93 olarak elde edilmiştir.



Şekil 3.5 Makine öğrenmesi algoritması hata matrisi

Makine öğrenmesi algoritmalarının doğruluk oranı eğitim verileri ile doğrudan ilişkilidir. Fakat bu oran sistemin tamamen doğru sonuçlar vereceği anlamına gelmemektedir. Kullanılan model az sayıda özelliği ele alarak sonuca varıyorsa model hızlı bir şekilde çalışacaktır ve doğruluk oranı yüksek gözükcektir. Fakat farklı veri kümeleri ile karşılaştığı durumda yanlış tahminlere yol açabilir. Bu tür olayların meydana gelmemesi için mevcut modele birkaç özellik daha eklenerek, model daha karmaşık hale getirilir. Karmaşık modellerde de dikkat edilmesi gereken ezberleme durumudur. Ezberleme durumunda doğruluk oranı %95’den fazla olarak görülür, ancak yeni gelen veri kümeleri ile karşılaştığında doğru tahmin yapmayabilir. Bu durumda, bir makine öğrenmesi algoritması oluşturulduğunda modelin ne çok karmaşık ne de çok basit olmaması gerekir.

Modelin doğruluk oranı istenilen rakamlara ulaşmadıysa, model üzerine iyileştirmeler yapılmalıdır. Sınıflandırma tahmini yapan makine öğrenmesi modelinin performans ölçümü için ROC (Receiver Operating Characteristic) eğrileri oldukça sık kullanılan bir yöntemdir. ROC bir olasılık eğrisidir ve AUC (Area Under The ROC Curve) ayrılabilirliğin derecesini veya ölçüsünü temsil eder. AUC değerinin yüksek olması, yüksek tahmin yapılabildiğini gösterir. Şekil 3.6, çalışmaya ait MATLAB’in "



Şekil 3.6 ROC eğrisi

ROC Curve" komutu ile elde edilen ROC eğrisi ve AUC oranını göstermektedir. AUC oranından görüldüğü üzere mevcut model yüksek derecede doğru tahmin yapmaktadır ve diğer makine öğrenmesi yöntemlerinden elde edilen AUC değerleri Tablo 3.2'de gösterilmektedir. Bu tabloda aynı zamanda yukarıda açıklandığı üzere "Classification Learner" uygulamasının "Train" komutu ile elde edilen 25 makine öğrenmesi yöntemine ait doğruluk oranları gösterilmektedir ve %91.1 doğruluk oranı ile en yüksek tahmin algoritması kübik destek vektör makinesi yöntemi kullanılarak elde edilmiştir.

3.1.4 Analitik Çözümün Sistemlere Entegre Edilmesi

"Classification Learner" uygulamasında "Export Model" seçeneği ile uygulamada oluşturulan modelin MATLAB ortamında kullanılabilir hale getirilmesi sağlanır. Bu doğrultuda, elde edilen model ile MATLAB ortamında "predict" komutunu kullanarak tahmin gerçekleştirilir. Kuvvet, tork ve direksiyon açısı bilgileri giriş olarak verilen bu modelde direksiyonu hangi el ile sürüldüğüne dair sonuç üretilir. Model sağ el ile sürüldüğü durumlarda çıktı olarak 1 değerini verirken sol el ile sürüldüğü durumlarda çıktı olarak -1 değerini verir

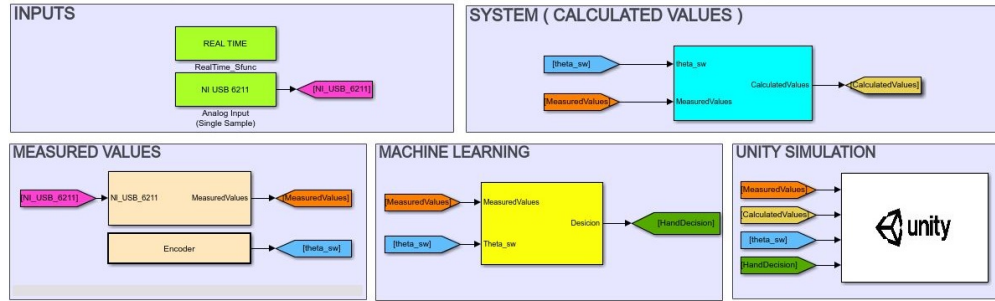
3.2 Sistemin Gerçek Zamanlı Çalışması

Tez çalışmasının en önemli ve yenilik katan noktalarından birisi de sürücü postür analizinin gerçek zamanlı olarak yapılmasıdır. Şekil 3.7 'de projenin Simulink'de oluşturulan modeli görülmektedir.

Bu tez çalışmasında kullanılan veri toplama sistemi MATLAB ile uyumlu olduğundan, Simulink ortamında "Data Acquisition Toolbox" kullanılarak veriler Simulink modeline

Tablo 3.2 Classification learner performans kıyaslaması

Kullanılan Yöntem	Doğruluk Oranı %	AUC
Fine Tree	88.3	0.90
Medium Tree	87.0	0.91
Medium Tree	76.5	0.72
Coarse Tree	75.2	0.78
Linear Discriminant	77.4	0.77
Quadratic Discriminant	73.6	0.78
Logistic Regression	61.0	0.77
Naive Bayes	75.7	0.86
Linear SVM	77.6	0.77
Quadratic SVM	86.3	0.94
Cubic SVM	91.1	0.97
Fine Gaussian SVM	87.1	0.95
Medium Gaussian SVM	83.4	0.93
Coarse Gaussian SVM	77.6	0.78
Fine KNN	88.7	0.88
Medium KNN	88.9	0.95
Coarse KNN	82.0	0.90
Cosine KNN	87.6	0.95
Cubic KNN	87.8	0.95
Weighted KNN	90.4	0.96
Boosted Trees	87.8	0.95
Bagged Trees	90.5	0.96
Subspace Discriminant	75.4	0.78
Subspace KNN	85.0	0.93
RUSBoosted Trees	87.2	0.94

**Şekil 3.7** Sistemin simulink modeli

aktarılmıştır. Kuvvet/tork sensörünün direksiyon ortasında bulunduğundan dolayı, sistem ilk çalıştırıldığında direksiyonun ağırlığından gelen kuvvet/tork değerlerinin kalibre edilmesi gereklidir. Kalibrasyon simülasyon başlatıldığında otomatik olarak direksiyon ağırlığından gelen değerleri sıfırlatacak şekilde yapılmıştır. Kalibre edilen değerler Simulink modelinde sürücü postür analiz algoritması ve makine öğrenmesi bloklarında kullanılmıştır. Simulink modelinin örnekleme zamanı 0.05 saniye olarak ayarlanmıştır. Modelden elde edilen sonuçlar "Unity" simülasyon programı üzerinden hazırlanmış olan arayüzde gösterilmiştir.

3.3 Sürücü Postür Algoritması Analiz Sonuçları

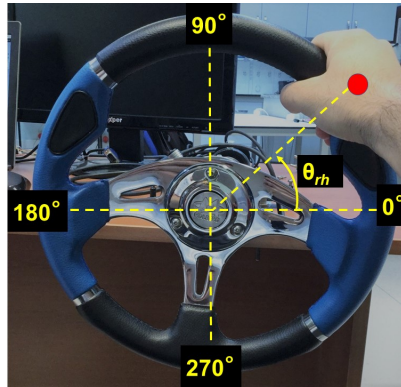
Bu tez çalışmasında, sürücü postür algoritmasının temel amacı, sensörler tarafından alınan kuvvet/tork ve direksiyon açısı değerleri ile sürücünün direksiyonu hangi konumdan tuttuğunu doğru bir şekilde hesaplanmasıdır. Bu algoritmanın sonuçlarını göstermek adına Tablo 3.3’de gösterilen senaryolar ele alınmıştır.

Tablo 3.3 Sürücü postür algoritması için sürüş senaryoları

Senaryo No.	Direksiyonu Tutan El	Elin Konum Aralığı (°)	Durum
1	Sağ	0-90	Şerit Değiştirme
2	Sağ	90-180	Şerit Değiştirme
3	Sağ	180-270	Şerit Değiştirme
4	Sağ	270-360	Şerit Değiştirme
5	Sol	0-90	Şerit Değiştirme
6	Sol	90-180	Şerit Değiştirme
7	Sol	180-270	Şerit Değiştirme
8	Sol	270-360	Şerit Değiştirme
9	Sağ	0-90	Ani Reaksiyon
10	Sol	90-180	Ani Reaksiyon

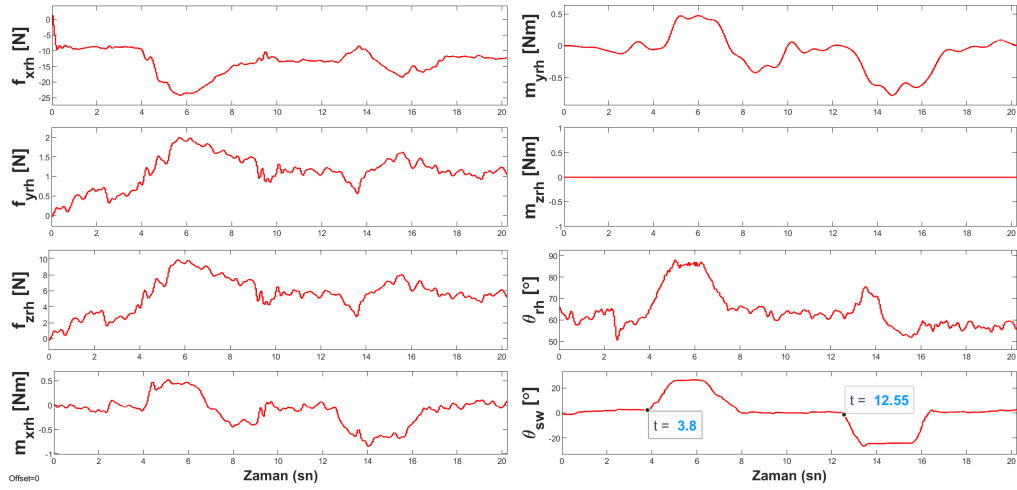
Bu senaryolar, sürüş esnasında sürücünün sağ veya sol el ile sürüş yaptığı durumlarda elin bulunabileceği tüm konum aralıklarını ele almaktadır:

- Şekil 3.8’de oluşturulan birinci sürüş senaryosuna dair durum görülmektedir. Şekil 3.8 incelendiğinde, sürücünün direksiyonu yaklaşık 60 derece konumundan tuttuğu ($\theta_{rh} = 60^\circ$) ve aracı sağ el ile kullandığı görülmektedir. Bu senaryoda, aracın yolun sağ şeridinde bulunduğu ve sürücünün şerit değişikliği yapıp sonrasında sağ şeride tekrardan geri döndüğü durum incelenmiştir.



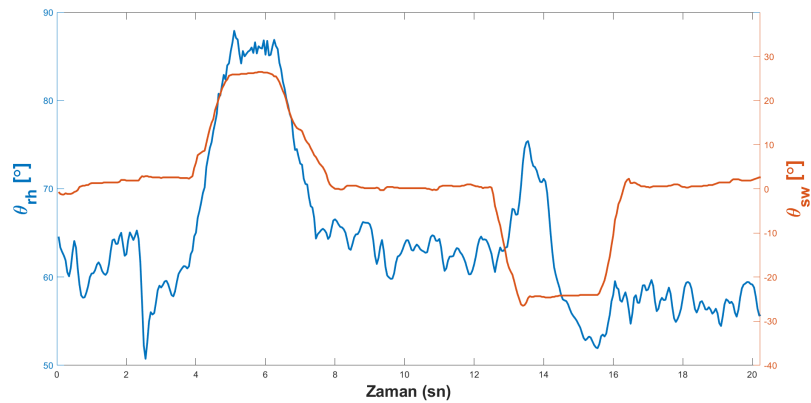
Şekil 3.8 Senaryo 1, sağ el ile sürüş

Şekil 3.9’da birinci senaryoda incelenen parametrelere dair sonuçlar grafiksel olarak gösterilmektedir. Bu parametreler arasında, f_{xrh} , f_{yrh} , f_{zrh} , m_{xrh} , m_{yrh} ve m_{zrh} sağ elin direksiyon üzerinde bulunduğu konumda oluşturduğu X, Y



Şekil 3.9 Senaryo 1, sağ el parametreleri grafikleri

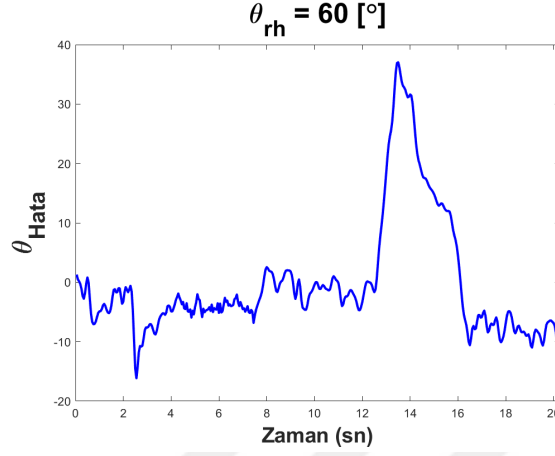
ve Z eksenlerindeki kuvvet ve tork değerlerini göstermektedir. θ_{rh} , geliştirilen sürücü postür algoritması sonucunda hesaplanan sağ el konum değerlerine ait grafiği ve θ_{sw} , enkoder tarafından ölçülen direksiyon açısı değerlerine ait grafiği göstermektedir. Şekil 3.9'da grafikler incelendiğinde, sürücünün ilk başta ($t = 0$ sn) düz bir şekilde sürüş gerçekleştirdiği gözlemlenmiştir ($\theta_{sw} \approx 0^\circ$). Sürüşün 3.8'inci saniyesinde ($t = 3.8$ sn), sürücünün sol şeride geçmek için direksiyonu yaklaşık 25 derecelik bir açı ile sola çevirdiği ($\theta_{sw} = 25^\circ$) ve sürüşün 12.5'inci saniyesinde ($t = 12.5$ sn), sürücünün tekrardan sağ şeride geçmek için direksiyonu sağa çevirdiği gözlemlenmiştir ($\theta_{sw} = -25^\circ$). Sürücünün şerit değişikliği yaparken direksiyonu çevirmesi ile birlikte sürücü postür algoritması ile hesaplanan θ_{rh} açısının aynı doğrultuda değişimi net bir şekilde gözlemlenmiştir ve bu değişim Şekil 3.10'de daha detaylı gösterilmektedir.



Şekil 3.10 Senaryo 1, direksiyon açısı ve sağ el konum grafikleri

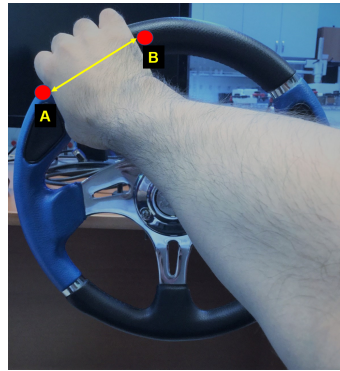
Şekil 3.11'de ise θ_{sw} ile θ_{rh} açısal konumların farkı görülmektedir (θ_{Hata}). Bu grafiği oluşturabilmek için θ_{rh} sinyali θ_{sw} başlangıç değerine ötelenmiştir. θ_{Hata} grafiği incelendiğinde, sürüşün 13.45'inci saniyesinde ($t = 13.45$ sn)

θ_{Hata} değerinin sürücünün direksiyonu sağa doğru çevirmesi ile yaklaşık 36° değerine çıktığı görülmektedir. Bu tez çalışmasında, sürücünün direksiyonu tek bir noktadan tuttuğu varsayılmaktadır.



Şekil 3.11 Senaryo 1, sağ el konum hata grafiği

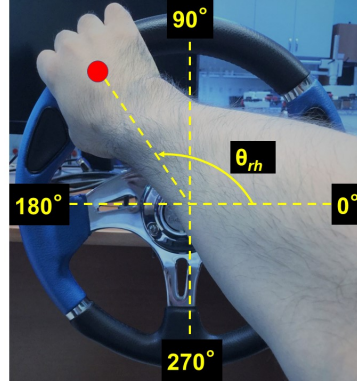
Diğer taraftan, Şekil 3.12'de görüldüğü üzere sürücü direksiyonu tek bir noktadan değil, geniş bir yüzeyden tutmaktadır. Bundan dolayı, sürücünün direksiyonu çevirdiği yöne ve harekete bağlı olarak elin direksiyona uyguladığı kuvvet ve torkun konumu Şekil 3.12'de görülen A ve B noktaları arasında değişebilmektedir (ortalama elin A ve B noktalar arasındaki uzaklığı 7.5cm olan bir kişinin yaklaşık olarak direksiyon üzerinde 30 derecelik farklı açıdan tutma ihtimali bulunur). Direksiyonun sağa doğru çevrilmesi ile sürücünün direksiyona uyguladığı kuvvet B noktasına doğru kaymaktadır ve bu doğrultuda θ_{Hata} değerinde artış gözlemlenmiştir.



Şekil 3.12 Direksiyonu tutma aralığı

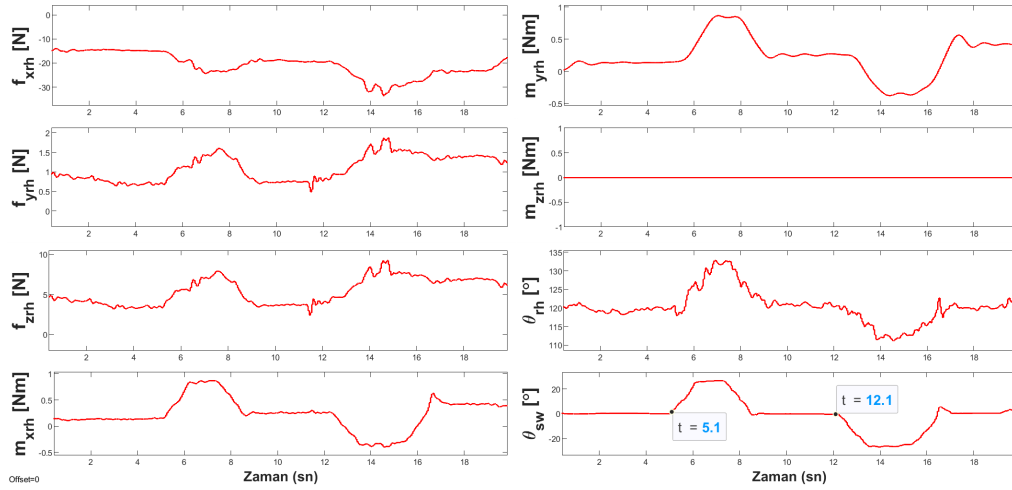
- ii. Şekil 3.13'de oluşturulan ikinci sürüş senaryosuna dair durum görülmektedir. Şekil 3.13 incelendiğinde, sürücünün direksiyonu yaklaşık 120 derece konumundan tuttuğu ($\theta_{rh} = 120^\circ$) ve aracı sağ el ile kullandığı görülmektedir. Bu senaryoda, aracın yolun sağ şeridinde bulunduğu ve sürücünün şerit

değişikliliği yapıp sonrasında sağ şeride tekrardan geri döndüğü durum incelenmiştir.



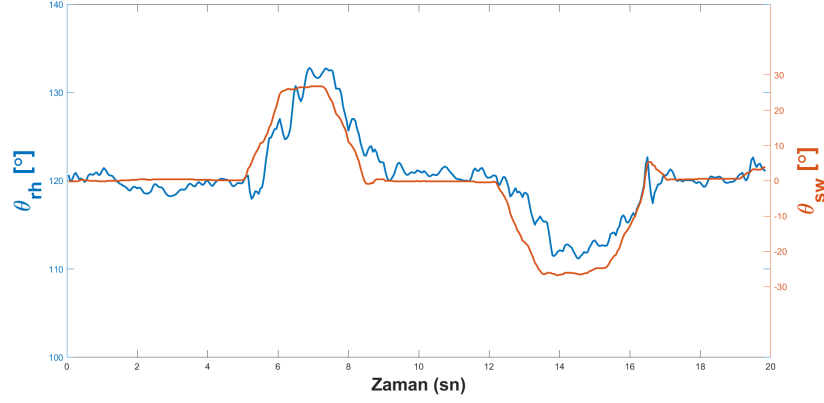
Şekil 3.13 Senaryo 2, sağ el ile sürüş

Şekil 3.14’de ikinci senaryoda incelenen parametrelere dair sonuçlar grafiksel olarak gösterilmektedir.



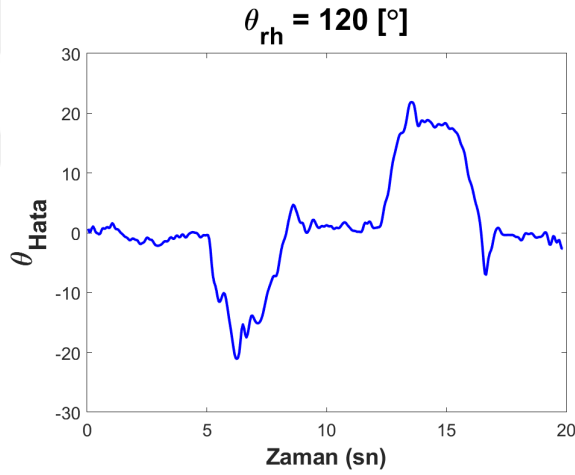
Şekil 3.14 Senaryo 2, sağ el parametreleri grafikleri

Şekil 3.14’de grafikler incelendiğinde, sürücünün ilk başta ($t = 0$ sn) düz bir şekilde sürüş gerçekleştirdiği gözlemlenmiştir ($\theta_{sw} \approx 0^\circ$). Sürüşün 5.1’inci saniyesinde ($t = 5.1$ sn), sürücünün sol şeride geçmek için direksiyonu yaklaşık 25 derecelik bir açı ile sola çevirdiği ($\theta_{sw} = 25^\circ$) ve sürüşün 12.1’inci saniyesinde ($t = 12.1$ sn), sürücünün tekrardan sağ şeride geçmek için direksiyonu sağa çevirdiği gözlemlenmiştir ($\theta_{sw} = -25^\circ$). Sürücünün şerit değişikliği yaparken direksiyonu çevirmesi ile birlikte sürücü postür algoritması ile hesaplanan θ_{rh} açısının aynı doğrultuda değişimi net bir şekilde gözlemlenmiştir ve bu değişim Şekil 3.15’de daha detaylı gösterilmektedir.



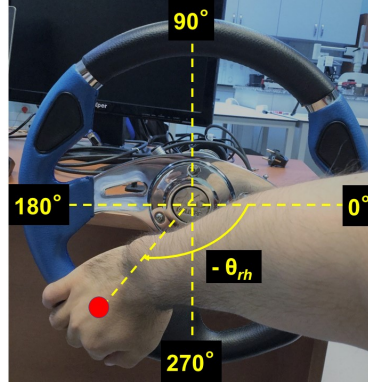
Şekil 3.15 Senaryo 2, direksiyon açısı ve sağ el konum grafikleri

Şekil 3.16'de ise θ_{sw} ile θ_{rh} açısal konumların farkı görülmektedir. Bu grafiği oluşturabilmek için θ_{rh} sinyali θ_{sw} başlangıç değerine ötelenmiştir. θ_{Hata} grafiği incelendiğinde, θ_{Hata} değerinin direksiyonun sola ve sağa çevrildiği durumlarda sürücünün direksiyona uyguladığı kuvvetin konumunun değişmesi nedeniyle arttığı gözlemlenmiştir.



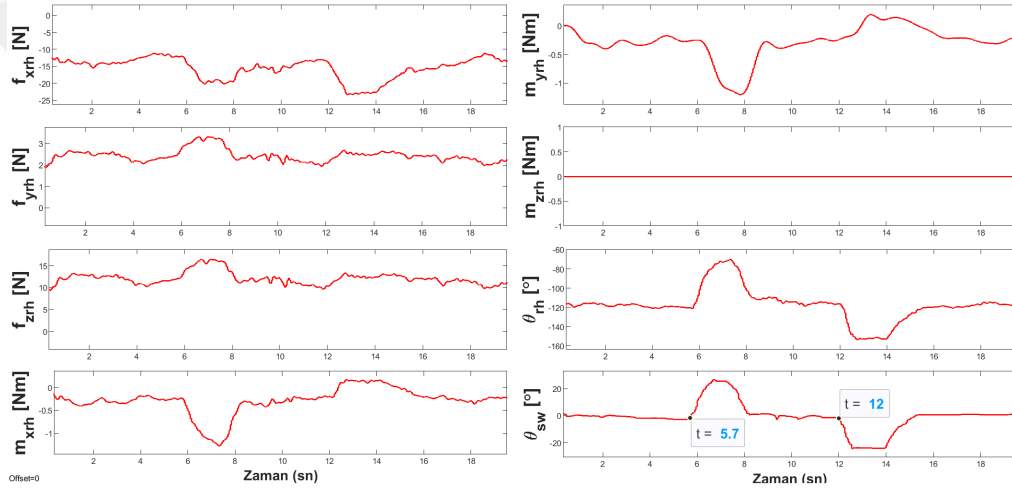
Şekil 3.16 Senaryo 2, sağ el konum hata grafiği

- iii. Şekil 3.17'da oluşturulan üçüncü sürüş senaryosuna dair durum görülmektedir. Şekil 3.17 incelendiğinde, sürücünün direksiyonu yaklaşık 230 derece konumundan tuttuğu ($\theta_{rh} = -130^\circ$) ve aracı sağ el ile kullandığı görülmektedir. Bu senaryoda, aracın yolun sağ şeridinde bulunduğu ve sürücünün şerit değişikliği yapıp sonrasında sağ şeride tekrardan geri döndüğü durum incelenmiştir.



Şekil 3.17 Senaryo 3, sağ el ile sürüş

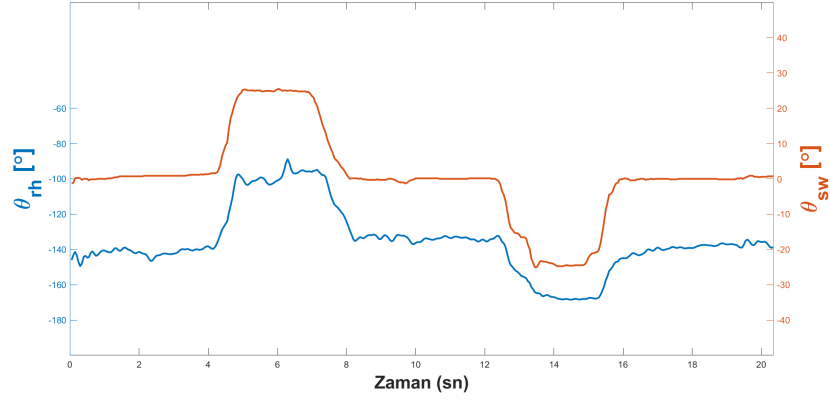
Şekil 3.18’de üçüncü senaryoda incelenen parametrelere dair sonuçlar grafiksel olarak gösterilmektedir.



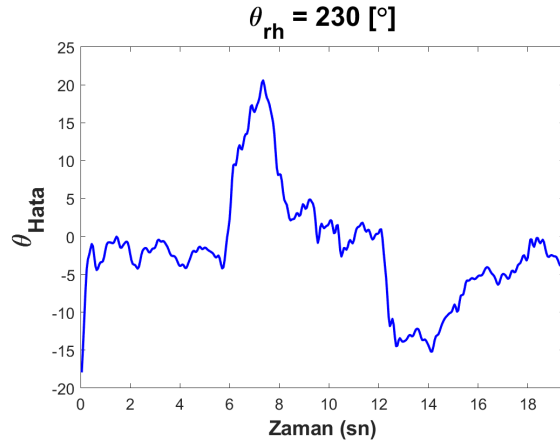
Şekil 3.18 Senaryo 3, sağ el parametreleri grafikleri

Şekil 3.18’de grafikler incelendiğinde, sürücünün ilk başta ($t = 0$ sn) düz bir şekilde sürüş gerçekleştirdiği gözlemlenmiştir ($\theta_{sw} \approx 0^\circ$). Sürüşün 5.7’inci saniyesinde ($t = 5.7$ sn), sürücünün sol şeride geçmek için direksiyonu yaklaşık 25 derecelik bir açı ile sola çevirdiği ($\theta_{sw} = 25^\circ$) ve sürüşün 12’inci saniyesinde ($t = 12$ sn), sürücünün tekrardan sağ şeride geçmek için direksiyonu sağa çevirdiği gözlemlenmiştir ($\theta_{sw} = -25^\circ$). Sürücünün şerit değişikliği yaparken direksiyonu çevirmesi ile birlikte sürücü postür algoritması ile hesaplanan θ_{rh} açısının aynı doğrultuda değişimi net bir şekilde gözlemlenmiştir ve bu değişim Şekil 3.19’de daha detaylı gösterilmektedir.

Şekil 3.20’de ise θ_{sw} ile θ_{rh} açısal konumların farkı görülmektedir. Bu grafiği oluşturabilmek için θ_{rh} sinyali θ_{sw} başlangıç değerine ötelenmiştir. θ_{Hata} grafiği incelendiğinde, θ_{Hata} değerinin direksiyonun sola ve sağa çevrildiği durumlarda sürücünün direksiyona uyguladığı kuvvetin konumunun değişmesi nedeniyle arttığı gözlemlenmiştir.



Şekil 3.19 Senaryo 3, direksiyon açısı ve sağ el konum grafikleri



Şekil 3.20 Senaryo 3, sağ el konum hata grafiği

iv. Şekil 3.21'de oluşturulan dördüncü sürüş senaryosuna dair durum görülmektedir.

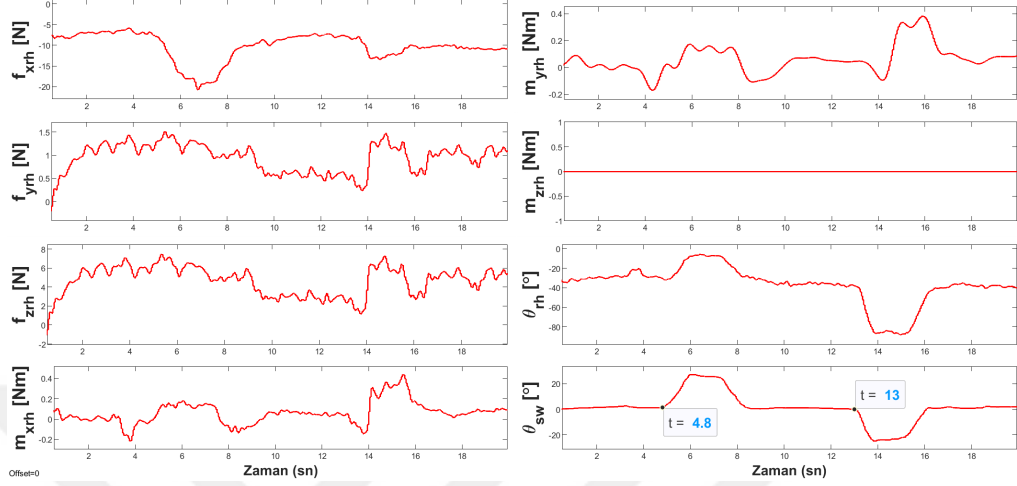


Şekil 3.21 Senaryo 4, sağ el ile sürüş

Şekil 3.21 incelendiğinde, sürücünün direksiyonu yaklaşık 330 derece konumundan tuttuğu ($\theta_{rh} = -30^\circ$) ve aracı sağ el ile kullandığı görülmektedir. Bu

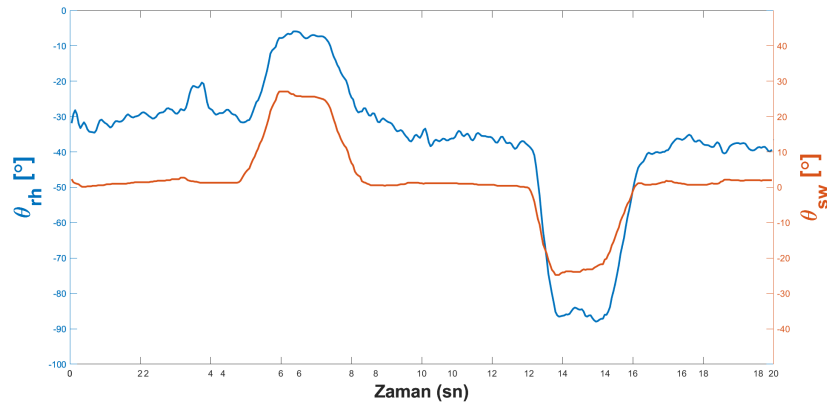
senaryoda, aracın yolun sağ şeridinde bulunduğu ve sürücünün şerit değişikliği yapıp sonrasında sağ şeride tekrardan geri döndüğü durum incelenmiştir.

Şekil 3.22’de dördüncü senaryoda incelenen parametrelere dair sonuçlar grafiksel olarak gösterilmektedir.



Şekil 3.22 Senaryo 4, sağ el parametreleri grafikleri

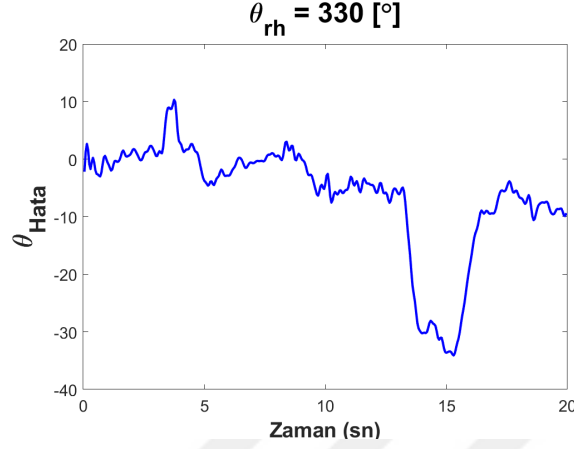
Şekil 3.22’de grafikler incelendiğinde, sürücünün ilk başta ($t = 0$ sn) düz bir şekilde sürüş gerçekleştirdiği gözlemlenmiştir ($\theta_{sw} \approx 0^\circ$). Sürüşün 4.8’inci saniyesinde ($t = 4.8$ sn), sürücünün sol şeride geçmek için direksiyonu yaklaşık 25 derecelik bir açı ile sola çevirdiği ($\theta_{sw} = 25^\circ$) ve sürüşün 13’üncü saniyesinde ($t = 13$ sn), sürücünün tekrardan sağ şeride geçmek için direksiyonu sağa çevirdiği gözlemlenmiştir ($\theta_{sw} = -25^\circ$). Sürücünün şerit değişikliği yaparken direksiyonu çevirmesi ile birlikte sürücü postür algoritması ile hesaplanan θ_{rh} açısının aynı doğrultuda değişimi net bir şekilde gözlemlenmiştir ve bu değişim Şekil 3.23’de daha detaylı gösterilmektedir.



Şekil 3.23 Senaryo 4, direksiyon açısı ve sağ el konum grafikleri

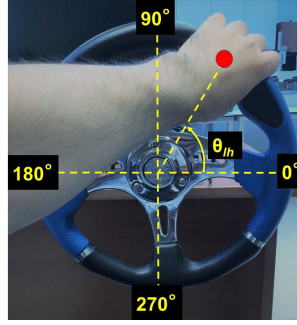
Şekil 3.24’de ise θ_{sw} ile θ_{rh} açısal konumların farkı görülmektedir. Bu grafiği oluşturabilmek için θ_{rh} sinyali θ_{sw} başlangıç değerine ötelenmiştir. θ_{Hata} grafiği

incelendiğinde, θ_{Hata} değerinin direksiyonun sola ve sağa çevrildiği durumlarda sürücünün direksiyona uyguladığı kuvvetin konumunun değişmesi nedeniyle arttığı gözlemlenmiştir.



Şekil 3.24 Senaryo 4, sağ el konum hata grafiği

v. Şekil 3.25’de oluşturulan beşinci sürüş senaryosuna dair durum görülmektedir.

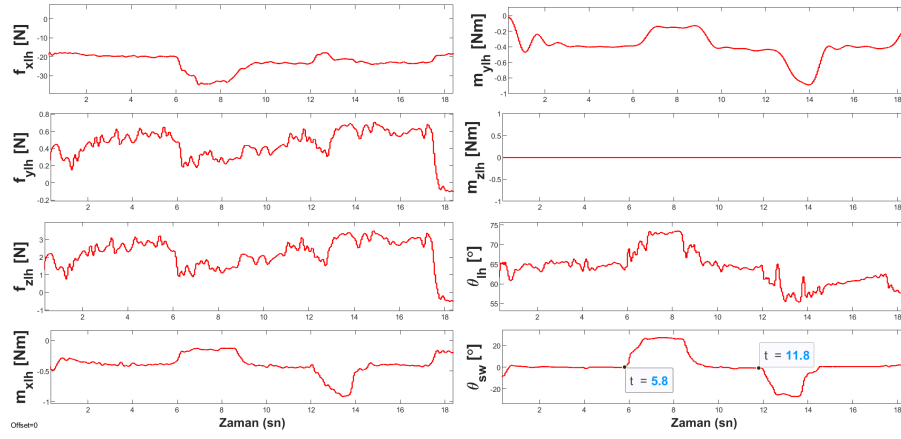


Şekil 3.25 Senaryo 5, sol el ile sürüş

Şekil 3.25 incelendiğinde, sürücünün direksiyonu yaklaşık 60 derece konumundan tuttuğu ($\theta_{lh} = 60^\circ$) ve aracı sol el ile kullandığı görülmektedir. Bu senaryoda, aracın yolun sağ şeridinde bulunduğu ve sürücünün şerit değişikliği yapıp sonrasında sağ şeride tekrardan geri döndüğü durum incelenmiştir.

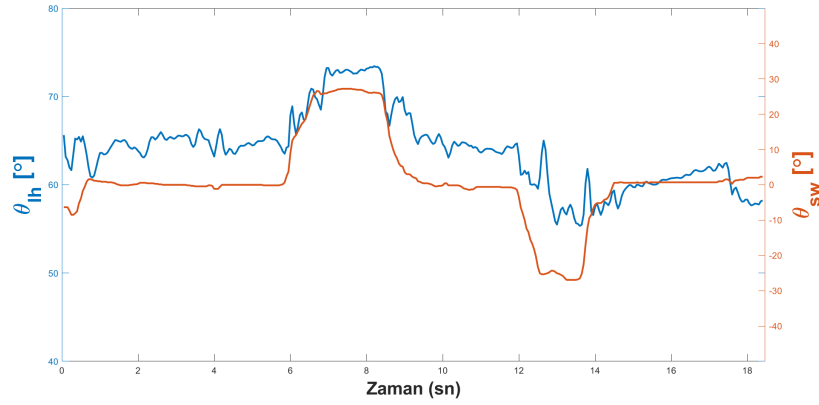
Şekil 3.26’de beşinci senaryoda incelenen parametrelere dair sonuçlar grafiksel olarak gösterilmektedir. Bu parametreler arasında, f_{xlh} , f_{ylh} , f_{zlh} , m_{xlh} , m_{ylh} ve m_{zlh} sol elin direksiyon üzerinde bulunduğu konumda oluşturduğu X, Y ve Z eksenlerindeki kuvvet ve tork değerlerini göstermektedir. θ_{lh} , geliştirilen sürücü postür algoritması sonucunda hesaplanan sol el konum değerlerine ait grafiği ve θ_{sw} , enkoder tarafından ölçülen direksiyon açısı değerlerine ait grafiği göstermektedir.

Şekil 3.26’de grafikler incelendiğinde, sürücünün ilk başta ($t = 0$ sn) düz bir şekilde sürüş gerçekleştirdiği gözlemlenmiştir ($\theta_{sw} \approx 0^\circ$). Sürüşün 5.8’inci

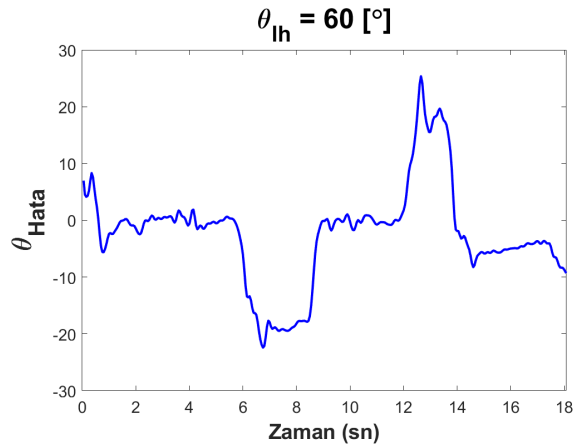


Şekil 3.26 Senaryo 5, sol el parametreleri grafikleri

saniyesinde ($t = 5.8$ sn), sürücünün sol şeride geçmek için direksiyonu yaklaşık 25 derecelik bir açı ile sola çevirdiği ($\theta_{sw} = 25^\circ$) ve sürüşün 11.8'inci saniyesinde ($t = 11.8$ sn), sürücünün tekrardan sağ şeride geçmek için direksiyonu sağa çevirdiği gözlemlenmiştir ($\theta_{sw} = -25^\circ$).



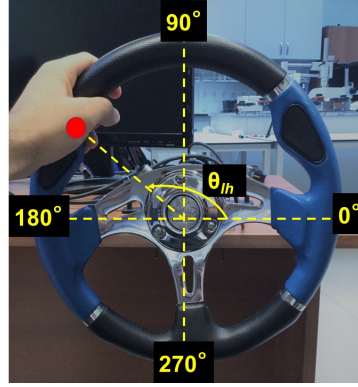
Şekil 3.27 Senaryo 5, direksiyon açısı ve sol el konum grafikleri



Şekil 3.28 Senaryo 5, sol el konum hata grafiği

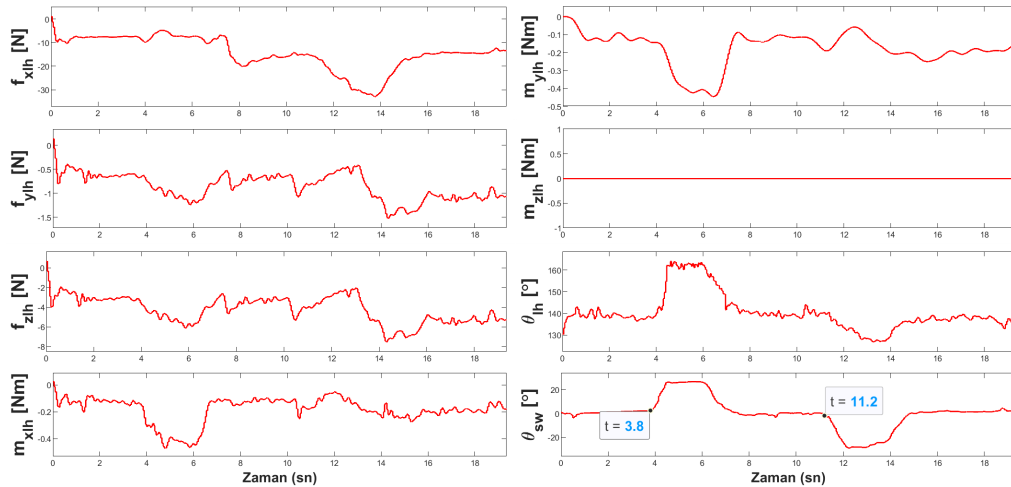
Sürücünün şerit değişikliği yaparken direksiyonu çevirmesi ile birlikte sürücü postür algoritması ile hesaplanan θ_{lh} açısının aynı doğrultuda değişimi net bir şekilde gözlemlenmiştir ve bu değişim Şekil 3.27’de daha detaylı gösterilmektedir. Şekil 3.28’de ise θ_{sw} ile θ_{lh} açısal konumların farkı görülmektedir. Bu grafiği oluşturabilmek için θ_{lh} sinyali θ_{sw} başlangıç değerine ötelenmiştir. θ_{Hata} grafiği incelendiğinde, θ_{Hata} değerinin direksiyonun sola ve sağa çevrildiği durumlarda sürücünün direksiyona uyguladığı kuvvetin konumunun değişmesi nedeniyle arttığı gözlemlenmiştir.

- vi. Şekil 3.29’de oluşturulan altıncı sürüş senaryosuna dair durum görülmektedir. Şekil 3.29 incelendiğinde, sürücünün direksiyonu yaklaşık 140 derece konumundan tuttuğu ($\theta_{lh} = 140^\circ$) ve aracı sol el ile kullandığı görülmektedir. Bu senaryoda, aracın yolun sağ şeridinde bulunduğu ve sürücünün şerit değişikliği yapıp sonrasında sağ şeride tekrardan geri döndüğü durum incelenmiştir.



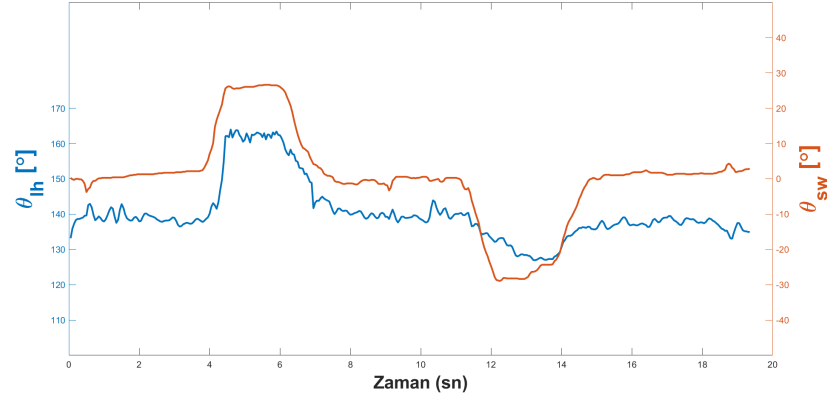
Şekil 3.29 Senaryo 6, sol el ile sürüş

Şekil 3.30’da altıncı senaryoda incelenen parametrelere dair sonuçlar grafiksel olarak gösterilmektedir.



Şekil 3.30 Senaryo 6, sol el parametreleri grafikleri

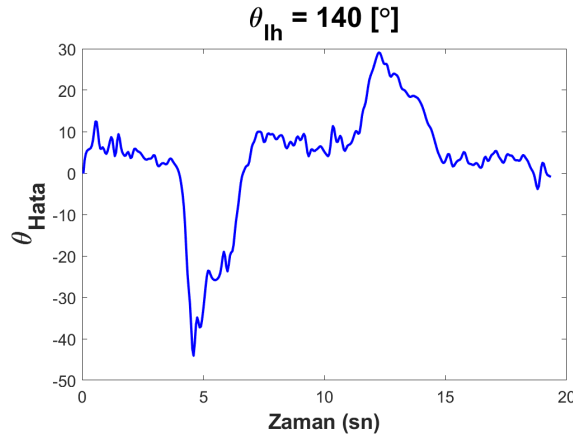
Şekil 3.30'da grafikler incelendiğinde, sürücünün ilk başta ($t = 0$ sn) düz bir şekilde sürüş gerçekleştirdiği gözlemlenmiştir ($\theta_{sw} \approx 0^\circ$). Sürüşün 3.8'inci saniyesinde ($t = 3.8$ sn), sürücünün sol şeride geçmek için direksiyonu yaklaşık 25 derecelik bir açı ile sola çevirdiği ($\theta_{sw} = 25^\circ$) ve sürüşün 11.2'inci saniyesinde ($t = 11.2$ sn), sürücünün tekrardan sağ şeride geçmek için direksiyonu sağa çevirdiği gözlemlenmiştir ($\theta_{sw} = -25^\circ$).



Şekil 3.31 Senaryo 6, direksiyon açısı ve sol el konum grafikleri

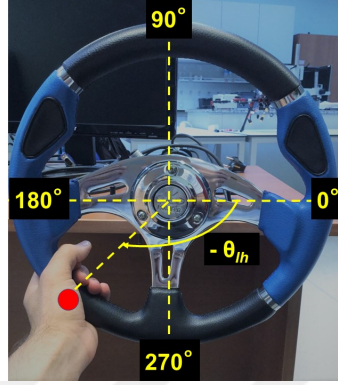
Sürücünün şerit değişikliği yaparken direksiyonu çevirmesi ile birlikte sürücü postür algoritması ile hesaplanan θ_{lh} açısının aynı doğrultuda değişimi net bir şekilde gözlemlenmiştir ve bu değişim Şekil 3.31'de daha detaylı gösterilmektedir.

Şekil 3.32'de ise θ_{sw} ile θ_{lh} açısal konumların farkı görülmektedir. Bu grafiği oluşturabilmek için θ_{lh} sinyali θ_{sw} başlangıç değerine ötelenmiştir. θ_{Hata} grafiği incelendiğinde, θ_{Hata} değerinin direksiyonun sola ve sağa çevrildiği durumlarda sürücünün direksiyona uyguladığı kuvvetin konumunun değişmesi nedeniyle arttığı gözlemlenmiştir.



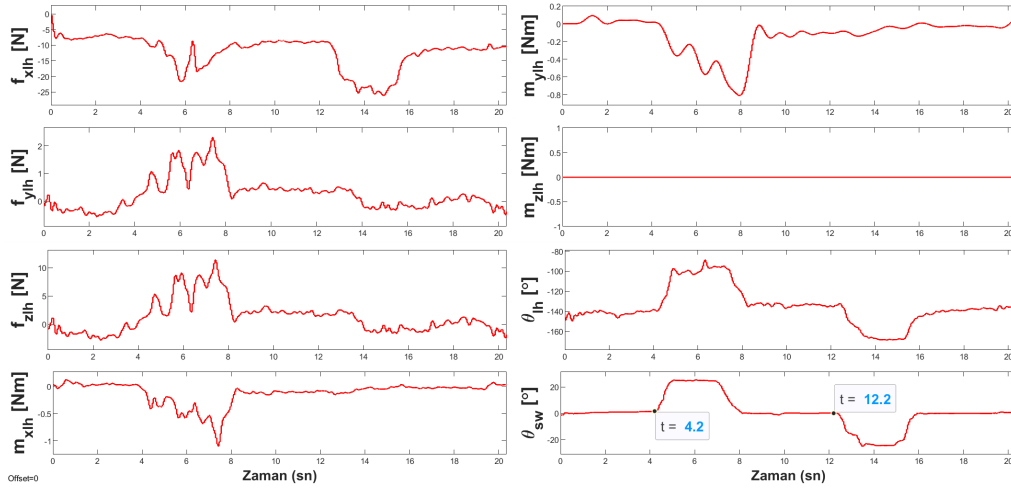
Şekil 3.32 Senaryo 6, sol el konum hata grafiği

- vii. Şekil 3.33’de oluşturulan yedinci sürüş senaryosuna dair durum görülmektedir. Şekil 3.33 incelendiğinde, sürücünün direksiyonu yaklaşık 220 derece konumundan tuttuğu ($\theta_{lh} = -140^\circ$) ve aracı sol el ile kullandığı görülmektedir. Bu senaryoda, aracın yolun sağ şeridinde bulunduğu ve sürücünün şerit değişikliği yapıp sonrasında sağ şeride tekrardan geri döndüğü durum incelenmiştir.



Şekil 3.33 Senaryo 7, sol el ile sürüş

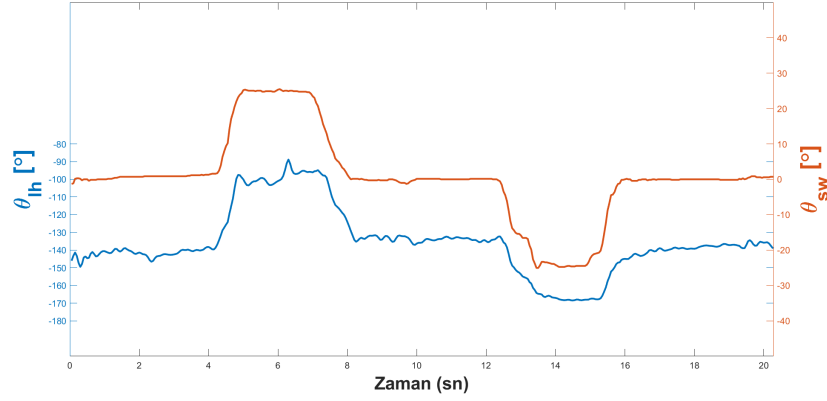
Şekil 3.34’de yedinci senaryoda incelenen parametrelere dair sonuçlar grafiksel olarak gösterilmektedir.



Şekil 3.34 Senaryo 7, sol el parametreleri grafikleri

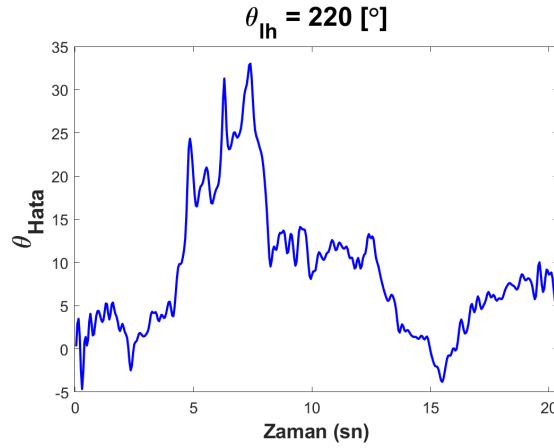
Şekil 3.34’de grafikler incelendiğinde, sürücünün ilk başta ($t = 0$ sn) düz bir şekilde sürüş gerçekleştirdiği gözlemlenmiştir ($\theta_{sw} \approx 0^\circ$). Sürüşün 4.2’inci saniyesinde ($t = 4.2$ sn), sürücünün sol şeride geçmek için direksiyonu yaklaşık 25 derecelik bir açı ile sola çevirdiği ($\theta_{sw} = 25^\circ$) ve sürüşün 12.2’inci saniyesinde ($t = 12.2$ sn), sürücünün tekrardan sağ şeride geçmek için direksiyonu sağa çevirdiği gözlemlenmiştir ($\theta_{sw} = -25^\circ$). Sürücünün şerit değişikliği yaparken direksiyonu çevirmesi ile birlikte sürücü postür algoritması ile hesaplanan θ_{lh}

açısının aynı doğrultuda değişimi net bir şekilde gözlemlenmiştir ve bu değişim Şekil 3.35’de daha detaylı gösterilmektedir.



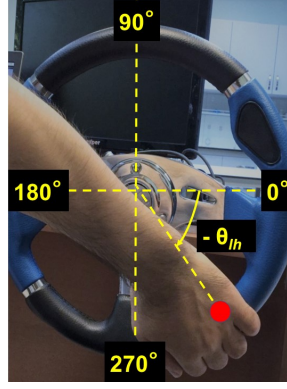
Şekil 3.35 Senaryo 7, direksiyon açısı ve sol el konum grafikleri

Şekil 3.36’de ise θ_{sw} ile θ_{lh} açısal konumların farkı görülmektedir. Bu grafiği oluşturabilmek için θ_{lh} sinyali θ_{sw} başlangıç değerine ötelenmiştir. θ_{Hata} grafiği incelendiğinde, θ_{Hata} değerinin direksiyonun sola ve sağa çevrildiği durumlarda sürücünün direksiyona uyguladığı kuvvetin konumunun değişmesi nedeniyle arttığı gözlemlenmiştir.



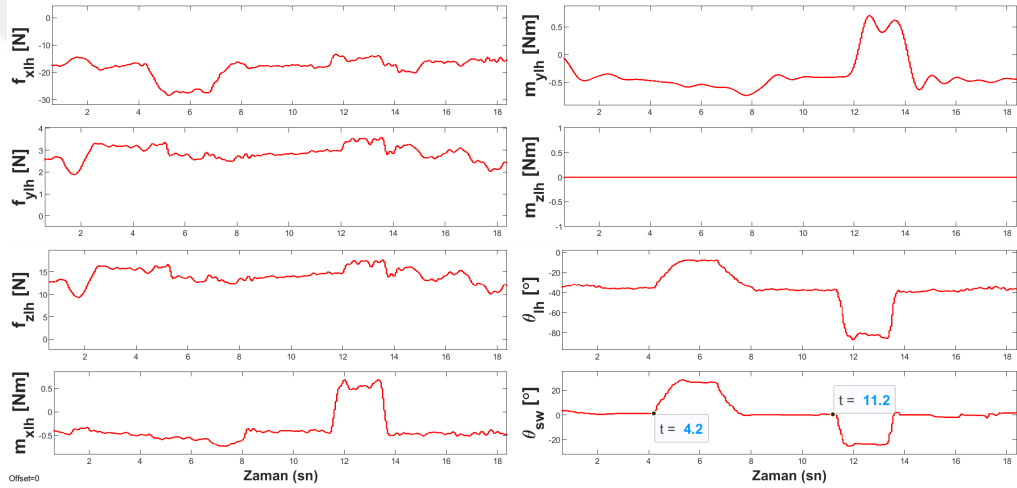
Şekil 3.36 Senaryo 7, sol el konum hata grafiği

- viii. Şekil 3.37’de oluşturulan sekizinci sürüş senaryosuna dair durum görülmektedir. Şekil 3.37 incelendiğinde, sürücünün direksiyonu yaklaşık 330 derece konumundan tuttuğu ($\theta_{lh} = -30^\circ$) ve aracı sol el ile kullandığı görülmektedir. Bu senaryoda, aracın yolun sağ şeridinde bulunduğu ve sürücünün şerit değişikliği yapıp sonrasında sağ şeride tekrardan geri döndüğü durum incelenmiştir.



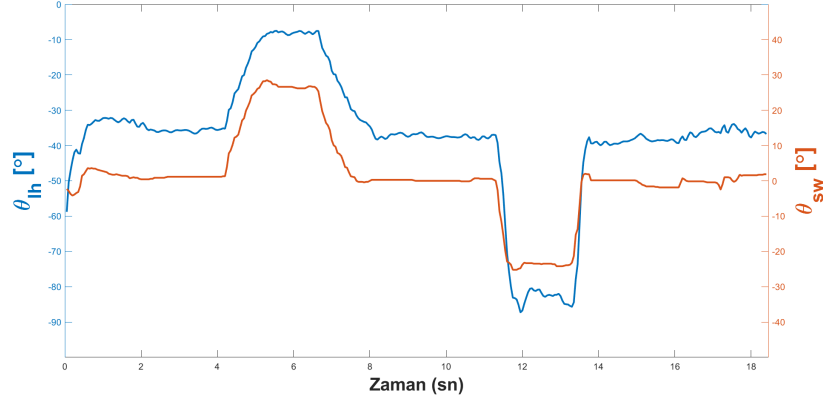
Şekil 3.37 Senaryo 8, sol el ile sürüş

Şekil 3.38’de sekizinci senaryoda incelenen parametrelere dair sonuçlar grafiksel olarak gösterilmektedir.



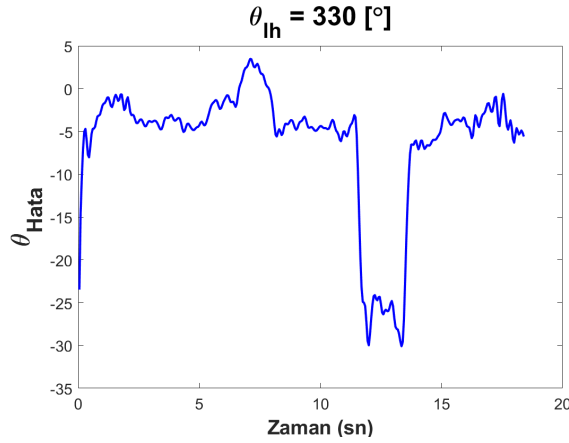
Şekil 3.38 Senaryo 8, sol el parametreleri grafikleri

Şekil 3.38’de grafikler incelendiğinde, sürücünün ilk başta ($t = 0$ sn) düz bir şekilde sürüş gerçekleştirdiği gözlemlenmiştir ($\theta_{sw} \approx 0^\circ$). Sürüşün 4.2’inci saniyesinde ($t = 4.2$ sn), sürücünün sol şeride geçmek için direksiyonu yaklaşık 25 derecelik bir açı ile sola çevirdiği ($\theta_{sw} = 25^\circ$) ve sürüşün 11.2’inci saniyesinde ($t = 11.2$ sn), sürücünün tekrardan sağ şeride geçmek için direksiyonu sağa çevirdiği gözlemlenmiştir ($\theta_{sw} = -25^\circ$). Sürücünün şerit değişikliği yaparken direksiyonu çevirmesi ile birlikte sürücü postür algoritması ile hesaplanan θ_{lh} açısının aynı doğrultuda değişimi net bir şekilde gözlemlenmiştir ve bu değişim Şekil 3.39’de daha detaylı gösterilmektedir.



Şekil 3.39 Senaryo 8, direksiyon açısı ve sol el konum grafikleri

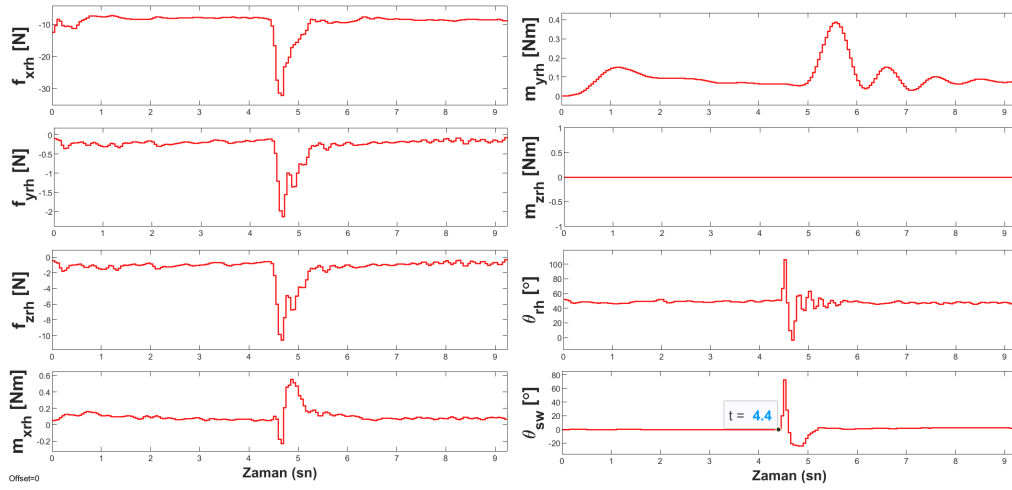
Şekil 3.40'da ise θ_{sw} ile θ_{lh} açısal konumların farkı görülmektedir. Bu grafiği oluşturabilmek için θ_{lh} sinyali θ_{sw} başlangıç değerine ötelenmiştir. θ_{Hata} grafiği incelendiğinde, θ_{Hata} değerinin direksiyonun sola ve sağa çevrildiği durumlarda sürücünün direksiyona uyguladığı kuvvetin konumunun değişmesi nedeniyle arttığı gözlemlenmiştir.



Şekil 3.40 Senaryo 8, sol el konum hata grafiği

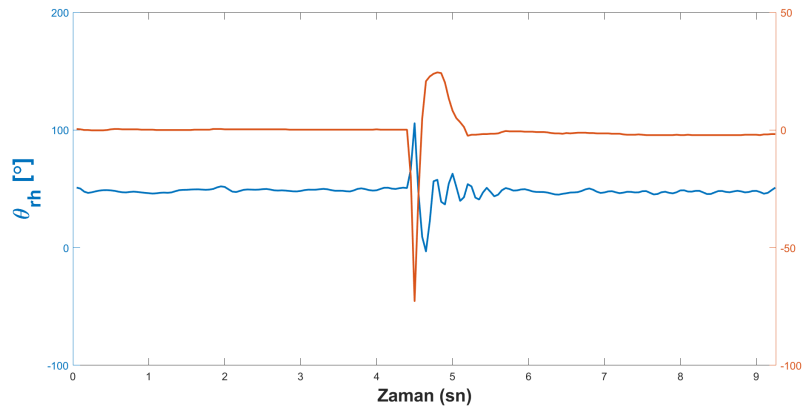
- ix. Dokuzuncu senaryoda, sürücünün düz bir şekilde yolda giderken önüne çıkan ani bir engelle gösterdiği reaksiyon incelenmiştir. Bu senaryoda sürücü aracı sağ el ile kullanmaktadır ve bu senaryonun diğer sekiz senaryodan farkı, ani bir şekilde direksiyonun yüksek derecelerde çevrilmesi durumunda sürücü postür algoritması ile hesaplanan el konum değerlerinin incelenmesidir.

Şekil 3.41'de bu senaryoda incelenen parametrelere dair sonuçlar grafiksel olarak gösterilmektedir. Şekil 3.41'de grafikler incelendiğinde, sürüşün 4.4'üncü saniyesinde ($t=4.4$ sn) sürücünün ani bir şekilde direksiyonu 75 derecelik bir açı ile sola çevirdiği ($\theta_{sw} = 75^\circ$) ve ardından 25 derecelik bir açı ile sağa çevirdiği ($\theta_{sw} = -75^\circ$) gözlemlenmiştir. Sürücünün ani bir şekilde direksiyonu sola ve



Şekil 3.41 Senaryo 9, sağ el parametreleri grafikleri

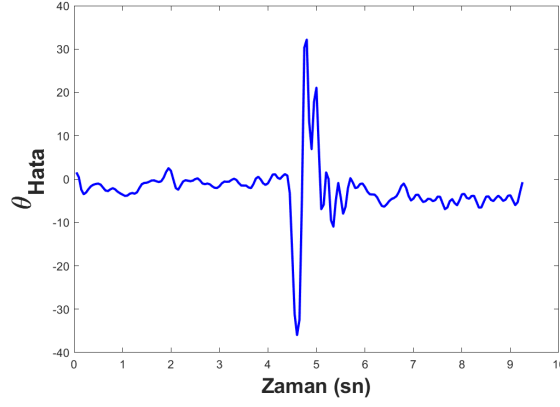
sağa çevirmesi ile birlikte sürücü postür postür algoritması ile hesaplanan θ_{rh} açısının aynı doğrultuda değişimi net bir şekilde gözlemlenmiştir ve bu değişim Şekil 3.42’de daha detaylı gösterilmektedir.



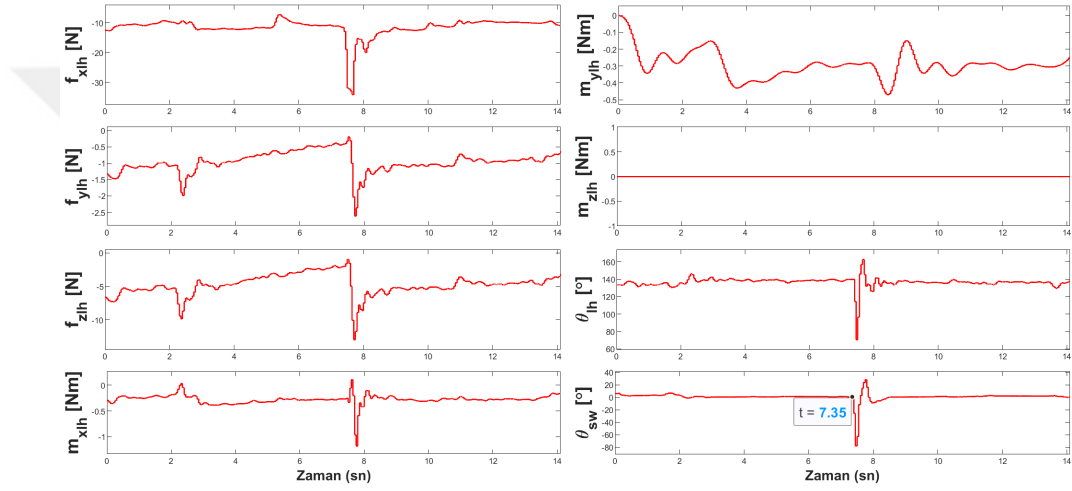
Şekil 3.42 Senaryo 9, direksiyon açısı ve sağ el konum grafikleri

Şekil 3.43’da ise θ_{sw} ile θ_{rh} açısal konumların farkı görülmektedir. Bu grafiği oluşturabilmek için θ_{rh} sinyali θ_{sw} başlangıç değerine ötelenmiştir. θ_{Hata} grafiği incelendiğinde, bu senaryoda ele alınan ani bir şekilde direksiyonun yüksek derecelere çıkması ile oluşan θ_{Hata} değerinin direksiyonun sola ve sağa çevrildiği durumlarda sürücünün direksiyona uyguladığı kuvvetin konumunun değişmesi nedeniyle arttığı ve diğer sekiz senaryoya benzer sonuç ürettiği gözlemlenmiştir.

- x. Onuncu senaryoda, sürücünün düz bir şekilde yolda giderken önüne çıkan ani bir engelle gösterdiği reaksiyon incelenmiştir. Bu senaryoda sürücü aracı sol el ile kullanmaktadır ve Şekil 3.44’de bu senaryoda incelenen parametrelere dair sonuçlar grafiksel olarak gösterilmektedir.



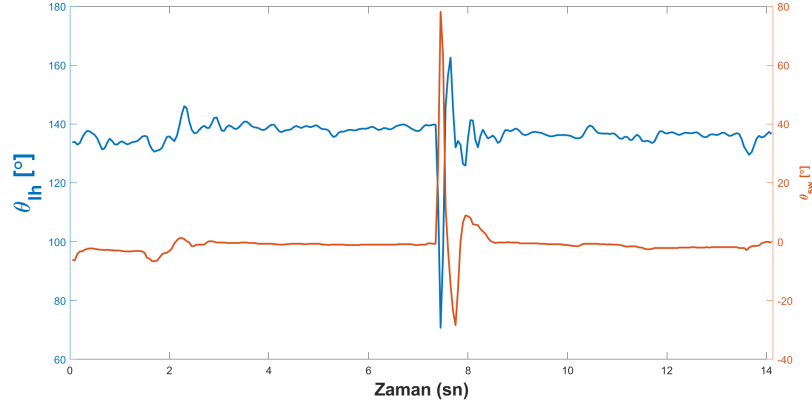
Şekil 3.43 Senaryo 9, sağ el konum hata grafiği



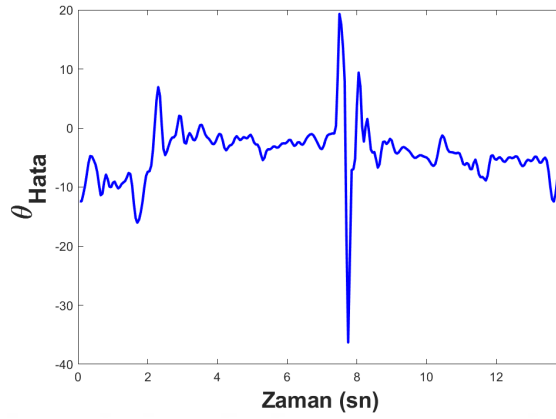
Şekil 3.44 Senaryo 10, sol el parametreleri grafikleri

Şekil 3.44'de grafikler incelendiğinde, sürüşün 7.35'inci saniyesinde ($t=7.35$ sn) sürücünün ani bir şekilde direksiyonu 80 derecelik bir açı ile sağa çevirdiği ($\theta_{sw} = -80^\circ$) ve ardından 30 derecelik bir açı ile sola çevirdiği ($\theta_{sw} = 30^\circ$) gözlemlenmiştir. Sürücünün ani bir şekilde direksiyonu sağa ve sola çevirmesi ile birlikte sürücü postür algoritması ile hesaplanan θ_{rh} açısının aynı doğrultuda değişimi net bir şekilde gözlemlenmiştir ve bu değişim Şekil 3.45'de daha detaylı gösterilmektedir.

Şekil 3.46'da ise θ_{sw} ile θ_{lh} açısal konumların farkı görülmektedir. Bu grafiği oluşturabilmek için θ_{lh} sinyali θ_{sw} başlangıç değerine ötelenmiştir. θ_{Hata} grafiği incelendiğinde, θ_{Hata} değerinin direksiyonun sağa ve sola çevrildiği durumlarda sürücünün direksiyona uyguladığı kuvvetin konumunun değişmesi nedeniyle arttığı gözlemlenmiştir.



Şekil 3.45 Senaryo 10, direksiyon açısı ve sol el konum grafikleri



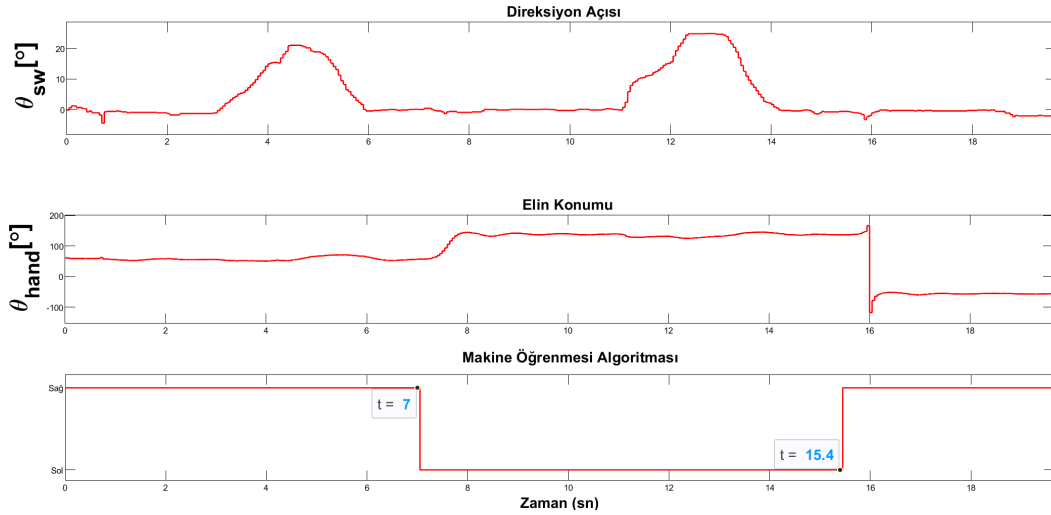
Şekil 3.46 Senaryo 10, sol el konum hata grafiği

3.4 Makine Öğrenmesi Algoritması Analiz Sonuçları

Projede geliştirilen makine öğrenmesi algoritma sonuçlarının gösterilmesi için yapılan birçok sürüş deneyleri arasından, modelin daha önce karşılaşmadığı 2 senaryo ele alınmıştır. Ele alınan senaryolar aşağıda verilmiştir:

- Şekil 3.47’de birinci senaryoda incelenen direksiyon açısı, elin direksiyonda bulunduğu konum açısı ve direksiyonu hangi el ile kullanıldığına dair grafikler gösterilmektedir.

Bu senaryoda sürücü ilk olarak düz bir yolda sağ el ile sürüş yapmaktadır ve bu esnada direksiyon simidini yaklaşık olarak 60 derecelik bir açı ile tutmaktadır ($\theta_{hand} = 60^\circ$). Ardından sürücü 20 derece ile sağ viraja dönüş sağlamaktadır ($\theta_{sw} = 20^\circ$) ve viraj bitiminde tekrardan düz yola devam etmektedir. Sürüşün 7’inci saniyesinde ($t = 7$ sn) sürücü sağ elini dinlendirmek adına, direksiyonda bulunan sağ elini bırakıp, sol el ile sürmeye devam etmektedir ve bu sırada

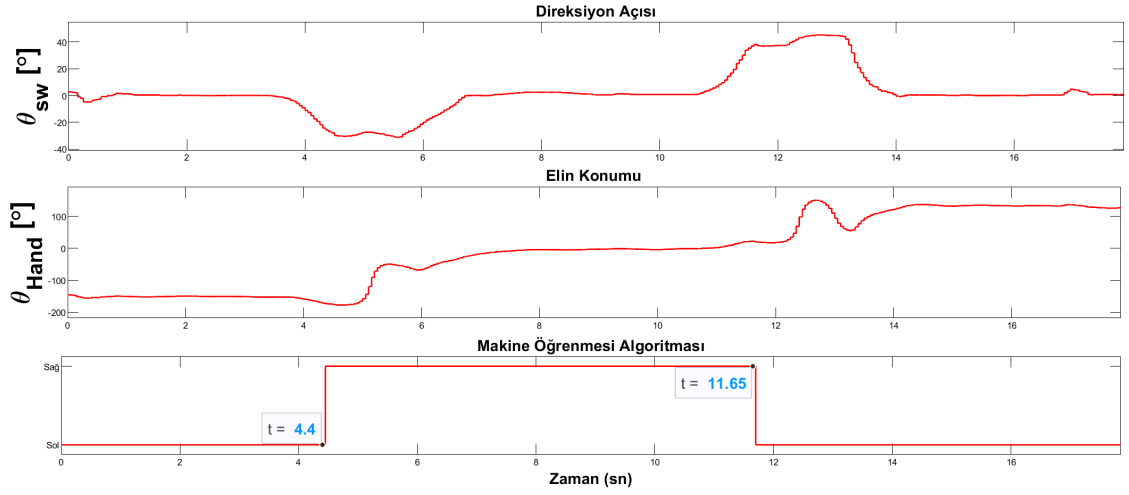


Şekil 3.47 Makine öğrenmesi birinci senaryo grafikleri

Şekil 3.47’de görüldüğü üzere makine öğrenmesi algoritması "Sol" sonucunu üretmiştir. Bu esnada, sürücü direksiyonu sol el ile yaklaşık 140 derecelik bir açı ile tutmaktadır ($\theta_{hand} = 140^\circ$). Sol el ile bir süre daha düz bir şekilde yoluna devam ettikten sonra, sürücü 25 derece ile tekrardan sağ viraja dönüş sağlamaktadır ve viraj bitiminde tekrardan düz yola devam etmektedir. Sürüşün 15.4’üncü saniyesinde ($t=15.4$ sn) sürücü tekrardan el değişikliği yaparak, sol elini direksiyon simidinden kaldırıp , sürüşe direksiyonu sağ el ile -55 derecelik bir açı ile tutarak devam eder ($\theta_{hand} = -55^\circ$). Bu sırada makine öğrenmesi algoritması grafiği incelendiğinde, algoritmanın "Sağ" sonucu ürettiği net bir şekilde gözlemlenmiştir.

- ii. Şekil 3.48’de ikinci senaryoda incelenen direksiyon açısı, elin direksiyonda bulunduğu konum açısı ve direksiyonu hangi el ile kullanıldığına dair grafikleri göstermektedir. Bu senaryoda sürücü ilk olarak düz bir yolda sol el ile sürüş yapmaktadır ve bu esnada direksiyon simidini yaklaşık olarak -130 derecelik bir açı ile tutmaktadır ($\theta_{hand} = -130^\circ$).

Ardından sürücü 30 derece ile sağ viraja dönüş sağlamaktadır ($\theta_{sw} = 30^\circ$). Sürüşün 4.4’üncü saniyesine ($t = 4.4$ sn), sürücü viraj dönüşünü yaparken daha konforlu bir şekilde dönüş sağlamak adına el değişikliği yapıp, direksiyonu sağ el ile sürmeye devam etmektedir ve bu esnada sürücü direksiyonu sağ el ile yaklaşık olarak -60 derecelik bir açı ile tutmaktadır ($\theta_{rh} = -60^\circ$). Bu esnada, Şekil 3.48’de görüldüğü üzere makine öğrenmesi algoritması "Sağ" sonucunu üretmiştir. Viraj bitiminde sürücü tekrardan aynı el ile bir süre daha düz yola devam etmektedir. Sağ el ile bir süre daha düz bir şekilde yoluna devam ettikten sonra, sürücü 45 derece ile sol viraja doğru dönüş sağlamaktadır ($\theta_{sw} = -45^\circ$). Sürüşün 11.65’inci saniyesinde ($t = 11.65$ sn),



Şekil 3.48 Makine öğrenmesi ikinci senaryo grafikleri

sürücü viraj dönüşünü yaparken daha konforlu bir şekilde dönüş sağlamak adına tekrardan el değişikliği yapıp, direksiyonu sol el ile sürmeye devam etmektedir ve bu sırada makine öğrenmesi grafiği incelendiğinde, algoritmanın "Sol" sonucu ürettiği net bir şekilde gözlemlenmiştir.

İki senaryo örneği sonucunda ortaya çıkan grafiklerden (Şekil 3.47 ve Şekil 3.48), makine öğrenmesi algoritmasının direksiyonun hangi el ile kullanıldığına dair doğru sonuçlar ürettiği görülmektedir. Makine öğrenmesi algoritmasında en önemli noktalardan biri sonucun değişkenlik göstermemesi, tutarlı olmasıdır. Bu çalışmada tutarlı sonuçların elde edilmesi verilerin doğru bir şekilde işlenmesi ve başarılı bir biçimde model oluşturulması anlamına gelmektedir. Yukarıda bahsi geçen makine öğrenmesi senaryolarının daha kolay anlaşılması için sürüşün kısa bir kısmı ele alınmıştır. Ayrıca, MATLAB üzerinde daha uzun süreli (1 saat) senaryolarda sistemin aynı şekilde doğru sonuçlar ürettiği gözlemlenmiştir. Buna istinaden, geliştirilen modelin yüksek doğrulukta sonuç üretmesi bu tür sistemlerin akıllı araçlarda gelişen teknoloji ile gittikçe daha fazla kullanılabilir olabileceğini göstermektedir.

4 SONUÇ VE ÖNERİLER

Bu çalışmada sürücünün direksiyonu tuttuğu konum ve kullandığı eli tespit edilerek sürüş postür analizi yapılmıştır. İlk olarak, direksiyona uygulanan kuvvet ve moment değerleri sensör ile ölçülüp incelenmiştir. Akabinde, bu ölçülen değerler Bölüm 2.2.2’de açıklanan sürücü postür algoritması ile çözümlenmiş olup, sürücünün direksiyonu tuttuğu konum hesaplanmıştır. Sürücü postür algoritmasının doğruluğunu ölçmek için bir deney düzeneği tasarlanmış ve üretilmiştir. Deney düzeneği ile yapılan testler sonucunda, direksiyonun 0-360° aralığında sürücü tarafından tutulması ihtimal dahilinde olan bütün tutuş aralıklarında doğru bir şekilde sonuç ürettiği görülmüştür. Aynı zamanda, elde edilen kuvvet/moment ve direksiyon açısı değerleri ile Bölüm 3.1’de bahsedilen makine öğrenmesi algoritması kullanılarak sürücünün direksiyonu hangi el ile sürdüğü başarı ile tespit edilmiştir. Makine öğrenmesinde destek vektör makineleri yöntemi kullanılmış olup, sürücünün aracı hangi el ile sürdüğü %91.1 doğruluk oranı ile tahminlenmiştir.

Bu tez çalışmasında geliştirilen algoritma, önceki bölümlerde açıklandığı üzere, literatürde yapılan çalışmalara istinaden gerçek zamanlı çalışmakta ve sürücünün fiziksel olarak direksiyona uyguladığı kuvvet ve moment değerlerini kullanarak çıkarım yapmaktadır. Algoritmanın ulaştığı en başarılı sonuçlardan biri, görüntü işleme yöntemi kullanılmadan sürücünün elinin direksiyon üzerindeki konumunu hesaplaması ve direksiyonu hangi el ile sürdüğünü yüksek doğruluk oranı ile tahmin edilmesidir. Projenin bir sonraki aşaması elde edilen sonuçlar ile tekrardan makine öğrenmesi algoritması kullanarak sürücünün mod tahminini yapmasıdır (uykulu, yorgun, sinirli). Aynı zamanda, elde edilen sonuçlar ile sürücüye ait birçok farklı çıkarım da yapılabilir. Örneğin, sürücünün direksiyonu tutma şeklinde bağlı olarak, sürücünün ani bir reaksiyon sergilediği davranışlar tespit edilebilir. Ani reaksiyonların tespiti, sürücünün kaza yapmaya yakın olduğu durumları tespit edebilir ve buna istinaden sürüş sistemine müdahale edilerek önlemler alınabilir. Bu tez çalışmasının görüntü işleme yöntemi ile birlikte kullanılması durumunda, sürücü davranışlarının daha hassas bir şekilde tahmin edileceği gözlemlenmiştir. Aynı zamanda, bu

tez alışmasından elde edilen veriler Stanford Üniversitesi tarafından geliştirilen "OpenSim" yazılım programı ile birlikte kullanılabilir. "OpenSim" programında, kas iskelet sisteminin modellenmesi sonucunda ters kinematik ve dinamik denklemler hesaplanabilmektedir. Bu tez alışmasında elde edilen el konumu ve kuvvet/tork deęerleri ile "OpenSim" programında kol ve gövde hareketleri hesaplanabilir. Böylelikle bu tez alışmasının bir sonraki aşamasında sürücü postürü ile ilgili kol ve gövde hareketlerine baęlı olarak farklı tespitler de yapılabilir.



- [1] *Türkiye istatistik kurumu*. [Online]. Available: <http://www.tuik.gov.tr/>.
- [2] W. Zhang, B. Cheng, and Y. Lin, "Driver drowsiness recognition based on computer vision technology," *Tsinghua Science and Technology*, vol. 17, no. 3, pp. 354–362, 2012.
- [3] Jian-Feng Xie, Mei Xie, and Wei Zhu, "Driver fatigue detection based on head gesture and perclos," in *2012 International Conference on Wavelet Active Media Technology and Information Processing (ICWAMTIP)*, 2012, pp. 128–131.
- [4] I. Teyeb, O. Jemai, M. Zaied, and C. Ben Amar, "A novel approach for drowsy driver detection using head posture estimation and eyes recognition system based on wavelet network," in *IISA 2014, The 5th International Conference on Information, Intelligence, Systems and Applications*, 2014, pp. 379–384.
- [5] Y. Abouelnaga, H. M. Eraqi, and M. N. Moustafa, *Real-time distracted driver posture classification*, 2017. arXiv: 1706.09498 [cs.CV].
- [6] Y. Liang, M. L. Reyes, and J. D. Lee, "Real-time detection of driver cognitive distraction using support vector machines," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 340–350, 2007.
- [7] M. Kutila, M. Jokela, G. Markkula, and M. R. Rue, "Driver distraction detection with a camera vision system," in *2007 IEEE International Conference on Image Processing*, vol. 6, 2007, pp. VI –201–VI –204.
- [8] M. Wollmer, C. Blaschke, T. Schindl, B. Schuller, B. Farber, S. Mayer, and B. Trefflich, "Online driver distraction detection using long short-term memory," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 574–582, 2011.
- [9] T. Liu, Y. Yang, G. Huang, Y. K. Yeo, and Z. Lin, "Driver distraction detection using semi-supervised machine learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1108–1120, 2016.
- [10] R. A. Berri, A. G. Silva, R. S. Parpinelli, E. Girardi, and R. Arthur, "A pattern recognition system for detecting use of mobile phones while driving," in *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 2, 2014, pp. 411–418.
- [11] B. Baheti, S. Gajre, and S. Talbar, "Detection of distracted driver using convolutional neural network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018.

- [12] N. Das, E. Ohn-Bar, and M. M. Trivedi, "On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 2953–2958.
- [13] C. Craye and F. Karray, *Driver distraction detection and recognition using rgb-d sensor*, 2015. arXiv: 1502.00250 [cs.CV].
- [14] E. Ohn-Bar, S. Martin, A. Tawari, and M. M. Trivedi, "Head, eye, and hand patterns for driver activity recognition," in *2014 22nd International Conference on Pattern Recognition*, 2014, pp. 660–665.
- [15] Y. Xing, C. Lv, Z. Zhang, H. Wang, X. Na, D. Cao, E. Velenis, and F. Wang, "Identification and analysis of driver postures for in-vehicle driving activities and secondary tasks recognition," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 1, pp. 95–108, 2018.
- [16] I. Amin, A. Taylor, and R. Parkin, "Driver tracking and posture detection using low-resolution infrared sensing," *Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering*, vol. 221, Sep. 2007. DOI: 10.1243/09544070JAUT0517.
- [17] C. Z. K. L. T. Hoang Ngan Le Yutong Zheng and M. Savvides, "Multiple scale faster-rcnn approach to driver's cell-phone usage and hands on steering wheel detection," 2015.
- [18] E. Ohn-Bar and M. M. Trivedi, "The power is in your hands: 3d analysis of hand gestures in naturalistic video," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2013.
- [19] R. Gupta, Mangalraj P, A. Agrawal, and A. Kumar, "Posture recognition for safe driving," in *2015 Third International Conference on Image Information Processing (ICIIP)*, 2015, pp. 141–146.
- [20] A. Riener and R. Alois Ferscha, "Supporting implicit human-to-vehicle interaction: Driver identification from sitting postures," Jul. 2008.
- [21] "Dynamic driver fatigue detection using hidden markov model in real driving condition," *Expert Systems with Applications*, vol. 63, pp. 397–411, 2016.
- [22] S. K. Staba Kovacevic Jovan Vucinic and N. Pejnovic, "Impact of anthropometric measurements on ergonomic driver posture and safety," Oct. 2009.
- [23] Y. Tanaka, Y. Kashiba, N. Yamada, T. Suetomi, K. Nishikawa, T. Nouzawa, and T. Tsuji, "Active-steering control system based on human hand impedance properties," in *2010 IEEE International Conference on Systems, Man and Cybernetics*, 2010, pp. 1697–1702.
- [24] B. R. S. T. Sarkar Dipanjan, *Practical Machine Learning with Python*. Apress, Jun. 2018.
- [25] D. Taşkın, "Türkiye’de ulaşım sektöründe yaşanan değişimler ve mevcut durum," *Doğu Coğrafya Dergisi*, 2016.
- [26] C. Tran and M. Trivedi, "Towards a vision-based system exploring 3d driver posture dynamics for driver assistance: Issues and possibilities," Jul. 2010, pp. 179–184. DOI: 10.1109/IVS.2010.5547957.

- [27] “Engineered products for robotic productivity.” [Online]. Available: <https://www.ati-ia.com/>.
- [28] “Kübler encoders.” [Online]. Available: <https://www.kuebler.com/en/products/measurement/encoders>.
- [29] D. R. W. Michael Lai and E. Krempl, *Introduction to Continuum Mechanics*, 4th ed. Butterworth-Heinemann, Aug. 2009.
- [30] “Support vector machines in machine learning.” [Online]. Available: <https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/>.



TEZDEN ÜRETİLMİŞ YAYINLAR

İletişim Bilgileri: ceyhunibolar@gmail.com

Konferans Bildirisi

1. Ceyhun Ibolar, A. Emre Çetin, Erhan Akdoğan;"Analysis and identification of driver posture and stat of mind", 5th International Conference on Advanves in Mechanical, Istanbul (2020).