POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering

Master's Degree Thesis

Driving Style Estimation with Driver Monitoring Systems using Nominal Driver Models

Supervisors

Prof. Massimo VIOLANTE

Dr. Nicolò CHIAPELLO

Dr. Marco LECCE

Candidate

Giuliano LORUSSO

April 2024

Glossary

AI Artificial Intelligence

DMS Driver Monitoring Systems

DSE Driver Style Estimation

NHTSA National Highway Traffic Safety Administration

ADAS Advance Driver Assistamnce System

ADS Automated Driving System

DDAW Driver Drowsiness and Attention Warning

ADDW Advanced Driver Distraction Warning

IEC International Electrotechnical Commission

ISO International Organization for Standardization

FFT Fast Fourier Transform

PCA Principal Component Analysis

IoT Internet of Things

ML Machine Learning

ECU Engine Control Unit

OEM Original Equipment Manufacturer

CAN Controller Area Network

ABS Antilock Braking System

ESC Electronic Stability Control

ECG electrocardiogram

UKF Unscented Kalman Filter

EKF Extended Kalman Filter

AWS Amazon Web Services

Summary

This research delves into the creation of an algorithm designed to identify driver distraction without the reliance on traditional biometric sensors. Instead, the software harnesses driving data directly obtained from the vehicle and its onboard sensors. The research is organized into two major domains, each addressing key aspects of distraction detection.

Identification of the Driver: The initial focus of the study revolves around the development of techniques for accurately identifying the driver. Utilizing vehicle data, the algorithm aims to distinguish between different individuals behind the wheel, laying the foundation for personalized distraction detection.

Distraction Detection Based on Correlation: The second part centers on the correlation between the actual and expected behavior of the driver. By analyzing real-time driving patterns and comparing them to a nominal behavior model, the algorithm can identify deviations indicative of distraction. This approach is designed to enhance the algorithm's adaptability and effectiveness across diverse driving scenarios. Methodology: The research employs a data-driven methodology, extracting and analyzing relevant information from the vehicle's sensors, including but not limited to speed, acceleration, and steering patterns. Machine learning and statistical modeling techniques are integrated to develop a robust algorithm capable of discerning distraction events with high accuracy.

Results and Contribution: Preliminary results showcase promising outcomes in both driver identification and distraction detection. The proposed algorithm not only contributes to the realm of driver safety but also introduces a novel approach that minimizes reliance on intrusive biometric sensors or gives support to the existing algorithm limited in detection of other types of distraction (cognitive distraction).

Future Implications: As the algorithm continues to evolve, its potential applications extend beyond distraction detection to include adaptive safety systems and personalized driving experiences. Future developments may also explore integration with emerging technologies in the automotive industry.

This thesis provides a comprehensive exploration of an algorithmic approach to distraction detection, offering a promising avenue for improving road safety without

the need for additional biometric hardware.

Acknowledgements

Table of Contents

Chapter 1

Introduction

1.1 Thesis genesis

This thesis is the result of a collaboration between Politecnico di Torino and Sensor Reply, a company within the Reply Group that focuses on IoT applications powered by Artificial Intelligence. Sensor Reply aims to provide software solutions using knowledge of model based and data driven approaches for decision support. A key achievement of this partnership is the creation of an Advanced Driver Distraction Warning (ADDW) system, which improves the efficiency and performance of the existing systems outlined in the baseline work [1]. The project introduces additional methods for this type of application, expanding the scope of investigation beyond the data-driven approaches previously discussed in further studies such as those documented by Nicolò Chiapello [2] and Enzo Yacacometti [3], focused on analyzing driving behaviors to develop impaired driving detection algorithms.

1.2 Objectives

The objectives established for achievement are as follows:

- Develop and refine a model-based approach for the extraction of supplementary features concerning driver behavior under scenario evolution. The principal aim is to enrich Machine Learning algorithms with data that extend beyond vehicle dynamics, including insights into the driver's intentions with respect to the surrounding environment, thus boosting both the system's performance and its learning capacity.

- Acquire authentic driving data, including both nominal and distracted driving instances, to facilitate the effective operation of the AI algorithms developed.

- Elevate the Driving Style Estimation (DSE process using data acquisition via a driving simulator.
- Realize an operational Advanced Driver Distraction Warning (ADDW) system that adheres to the current drafts of the 2019/2144 EU regulations, with a focus on maintaining a low False Positive (FP) rate, for future integration with dashboard camera-based biometric data.
- Assess the full spectrum of potential solutions to determine the optimal configuration that meets the standards set forth by regulatory requirements.

1.3 Document structure

The present work is structured as follows:

1. **Introduction:** An overview of the problem context and a detailed explanation of the methodologies employed to address it.
2. **State of the Art:** A review of pertinent regulations, a definition of the project's objectives, and an exploration of alternative approaches.
3. **Background:** A comprehensive explanation of the techniques, algorithms, and physical models utilized in this study.
4. **Material and Methods:** A detailed account of the experimental setup and the methodologies used in system development.
5. **Experimental Results:** A presentation and analysis of the training and testing outcomes at each stage of the implementation.
6. **Discussion:** An examination of the adopted approaches and methodologies in relation to the results obtained.
7. **Conclusions:** Final thoughts and a recapitulation of the aims of this study, with a perspective on potential future developments and applications.

Sustainable Development Goals

Chapter 2

State of Art

2.1 Driver Behavior

Add something here

2.1.1 Driver State Monitoring

In the field of intelligent transport, ensuring driver alertness is critical to road safety. The advent of Driver State Monitoring (DSM) systems marks a significant leap forward in reducing accidents caused by driver fatigue and distraction. Advances in computing and artificial intelligence have played a crucial role in improving the effectiveness of DSM systems, making them an integral part of both conventional and automated vehicles. This chapter examines the evolution, methods, challenges and regulatory landscape of DSM, highlighting its critical role in modern vehicle safety.

The Evolution of Driver State Monitoring Systems

DMS systems have evolved significantly, transitioning from rudimentary alertness checks to sophisticated AI-driven solutions. Initially, these systems were focused on simple metrics such as steering wheel movement and eyelid tracking. An early example includes Volvo's introduction of the Driver Alert Control in 2007 [4], which monitored the car's movements to detect signs of driver inattention or fatigue. Over time, DMS systems have incorporated more complex algorithms capable of analyzing a myriad of physiological and behavioral cues to assess driver state accurately. Furthermore, the integration of wearable devices is opening up new avenues for monitoring physiological signals such as heart rate variability and brain activity, providing deeper insights into driver status. A notable example is the collaboration between Ford and the Massachusetts Institute of Technology

[5] to investigate the use of wearable ECG monitors to detect stress levels and adjust vehicle systems accordingly to improve driver safety. These developments highlight the interdisciplinary nature of DMS's research, combining insights from psychology, computer science and automotive engineering to create systems that are increasingly sensitive, accurate and able to operate in real time.

Methodologies in Driver State Monitoring

The methodologies employed in DSM are several, each with its unique approach to

- **Image-Based Measures**: These measures utilize cameras to monitor the driver's facial expressions, eye movements, and head posture. By employing computer vision techniques, they can detect signs of drowsiness or distraction effectively. For instance, the Driver Attention Monitoring System by Toyota [6], uses a camera-based approach to monitor the driver's gaze direction to ensure attention is maintained on the road.

- **Biological-Based Measures**: This approach involves sensors that track physiological signals, such as heart rate variability, brainwave patterns, and galvanic skin response. These measures offer insights into the driver's physical state, potentially indicating levels of stress, fatigue, or drowsiness.

- **Vehicle-Based Measures**: These measures analyze data from the vehicle's control systems, such as steering patterns, braking behavior, and acceleration. Variations in these parameters can infer the driver's level of engagement and alertness. For example, research by Dong et al. (2011) explored the use of steering wheel movements and lane-keeping behavior as indicators of driver fatigue [7].

- **Hybrid Measures**: Combining data from multiple sources, hybrid systems offer a more comprehensive assessment of the driver's state. These systems might correlate physiological signals with behavioral patterns to provide a holistic view of the driver's condition.

This thesis investigates the specific aspect of driver distraction in more detail in the following chapter, where it becomes clear that understanding and mitigating distraction requires a complex approach that considers both the external and internal factors that influence driver attention.

2.1.2 Driver Distraction

Distracted driving remains a major public health concern, contributing to a significant proportion of road deaths. In 2021, in America, distracted driving will be responsible for 3,522 deaths, highlighting the urgent need for effective interventions and policies.[8] Distracted driving limits the driver's attention to essential tasks such as controlling the vehicle's position and maintaining speed. Distracted driving can be broadly defined as the diversion of attention from driving tasks caused by focusing on non-driving objects, tasks or events, thereby reducing the driver's awareness, decision making and performance. This distraction significantly increases the risk of corrective action required, near misses or actual crashes. In 2021, the European Commission reported an estimated 19,800 road fatalities within the EU, marking a 5% increase from 2020 but still showing a 13% decrease compared to the pre-pandemic levels in 2019. This indicates a fluctuating trend in road safety, partly influenced by traffic patterns during the pandemic.

Further evidence-based reviews have highlighted the relationship between distracted driving, particularly due to cell phone use, and an increased risk of automobile accidents. These reviews recommend minimizing in-vehicle distractions and specifically advise against texting or using touch messaging systems while driving. The risks associated with distracted driving are particularly heightened among younger, inexperienced drivers, who are advised to refrain from using cell phones or engaging in texting while driving [9].

In Figure 2.1 are reported statistics on activities commonly associated with distracted driving behaviors reported from the EU commission [10].

In response to the dangers posed by distracted driving, various campaigns and guidelines have been introduced to raise awareness and promote road safety. For instance, the Federal Motor Carrier Safety Administration, along with other agencies, has banned all hand-held cell phone use by commercial drivers and those carrying hazardous materials. The National Highway Traffic Safety Administration (NHTSA) has also been proactive, launching campaigns like "U Drive. U Text. U Pay." to emphasize the consequences of distracted driving.

These initiatives reflect a growing recognition of the need to address distracted driving through both policy measures and public awareness campaigns. The ultimate goal is to reduce the incidence of distracted driving and enhance road safety.

Types of Driving Distractions The U.S. National Highway Traffic Safety Administration (NHTSA) characterizes driving distractions as any endeavors that shift a driver's focus away from the road [11]. They highlight that such distractions are not limited to merely sending text messages or having phone conversations. Distractions can also involve eating, interacting with passengers, or controlling the car's audio or navigational systems Driving distractions can be categorized into several types, each with unique implications for driver focus and road safety:

Activity	Odds ratio (95% CI)	% driving time
Grouped activities		
All activities*[1]	**2.0** (1.8-2.4)	51.9
All primarily cognitive activities[2]	**1.25** (1.01-1.54)	20.0
All hand-held phone activities*[3]	**3.6** (2.9-4.5)	6.40
Visual-manual tasks: hand-held texting, browsing, or dialling	**2.56** (1.68-3.88)	1.8
All activities related to in-vehicle devices*	**2.5** (1.8-3.4)	3.53
Primarily cognitive activities		
Hand-held phoning (talking/listening)	1.27 (0.79-2.04)	2.7
Hands-free phoning (talking/listening)	0.4 (0.10-1.63)	0.9
Talking/singing alone	1.44 (0.99-2.08)	4.2
Interacting with passengers	1.26 (0.98-1.62)	12.2
Activities with in-vehicle devices		
Adjusting radio	1.57 (0.85-2.91)	1.3
Controlling temperature/air conditioning*	**2.3** (1.1-5.0)	0.56
Interacting with vehicle device (other)*	**4.6** (2.9-7.4)	0.83
Mobile phone activities		
Purely holding a phone in the hand	**2.05** (1.13-3.73)	1.1
Reaching for phone*	**4.8** (2.7-8.4)	0.58
Dialling (handheld)*	**12.2** (5.6-26.4)	0.14
Reading/writing text messages (texting)*	**6.1** (4.5-8.2)	1.91
Browsing (e.g., read email, check internet)*	**2.7** (1.5-5.1)	0.73
Other activities		
Reading/writing (also tablet)*	**9.9** (3.6-26.9)	0.09
Reaching for an object (no phone)*	**9.1** (6.5-12.6)	1.08
Prolonged looking at external object*	**7.1** (4.8-10.4)	0.93
Eating*	**1.8** (1.1-2.9)	1.90
Drinking (no alcohol)*	1.8 (1.0-3.3)	1.22
Personal hygiene (e.g., make-up)*	1.4 (0.8-2.5)	1.69
Child in rear seat*	0.5 (0.1-1.9)	0.80

Notes: Odds ratios significantly different from 1 are in ***bold*** *($p \leq 0.05$) (Dingus et al., 2019 and *2016)*

Figure 2.1: Distraction Activities and Crash Risk associated with them

- **Visual Distractions:** This involves taking one's eyes off the road. Examples include looking at a GPS device, reading billboards, or observing an event outside the vehicle.

- **Manual Distractions:** These occur when the driver takes one or both hands off the steering wheel. Examples include eating, drinking, adjusting the radio or climate controls, and smoking.
- **Cognitive Distractions:** This type of distraction happens when the driver's mind is not focused on driving. Cognitive distractions can be due to daydreaming, deep conversations with passengers, or being preoccupied with personal, family, or work-related issues.
- **Auditory Distractions:** These are caused by sounds that are not related to driving, such as conversations among passengers, phone calls, or loud music, which might lead the driver to lose focus on the road.

Given the complexity of distracted driving and its profound impact on road safety, section 2.2.4 will focus into the mechanisms of distraction, exploring the effectiveness of current interventions, and examine emerging technologies and policies aimed at mitigating this issue. Before of entering in deep details it is important to

2.2 Technical Legislation

The development and implementation of driver state monitoring technologies requires a solid regulatory foundation. This foundation not only provides a consistent framework for manufacturers, but also ensures compliance with a standardised set of rules, thereby enhancing the quality, safety and efficiency of automotive technologies. The international scope of the automotive industry requires a clear understanding of the various legislative documents governing the field of automated driving systems (ADS) and advanced driver assistance systems (ADAS).

It is important to distinguish between:

- **International Standards** such as ISO 26262, play a central role in setting global benchmarks for the functional safety of electrical and electronic systems in road vehicles. These standards guide the entire lifecycle of vehicle development, from design to production, and ensure that driver state monitoring technologies meet universally recognised safety and quality standards.
- **International Regulations**, particularly those from the United Nations Economic Commission for Europe (UNECE), provide detailed mandates on the legal requirements vehicles, systems, or components must satisfy to be considered roadworthy. These regulations are mandatory and detail the

compliance criteria for ADAS and ADS technologies, unlike the advisory nature of International Standards.

2.2.1 SAE

In the domain of vehicular safety and performance analysis, precisely defined metrics are indispensable. These metrics facilitate an objective assessment of driver behavior and vehicle operation, contributing to the development and evaluation of Advanced Driver Assistance Systems (ADAS). The SAE Driver Metrics, Performance, Behaviors and States Committee provides a set of operational definitions for such measures, which are crucial for standardizing research and development practices within the automotive industry. Table 2.1 delineates a selection of these driver performance measures as standardized in SAE J2944 [12].

2.2.2 ISO 26262

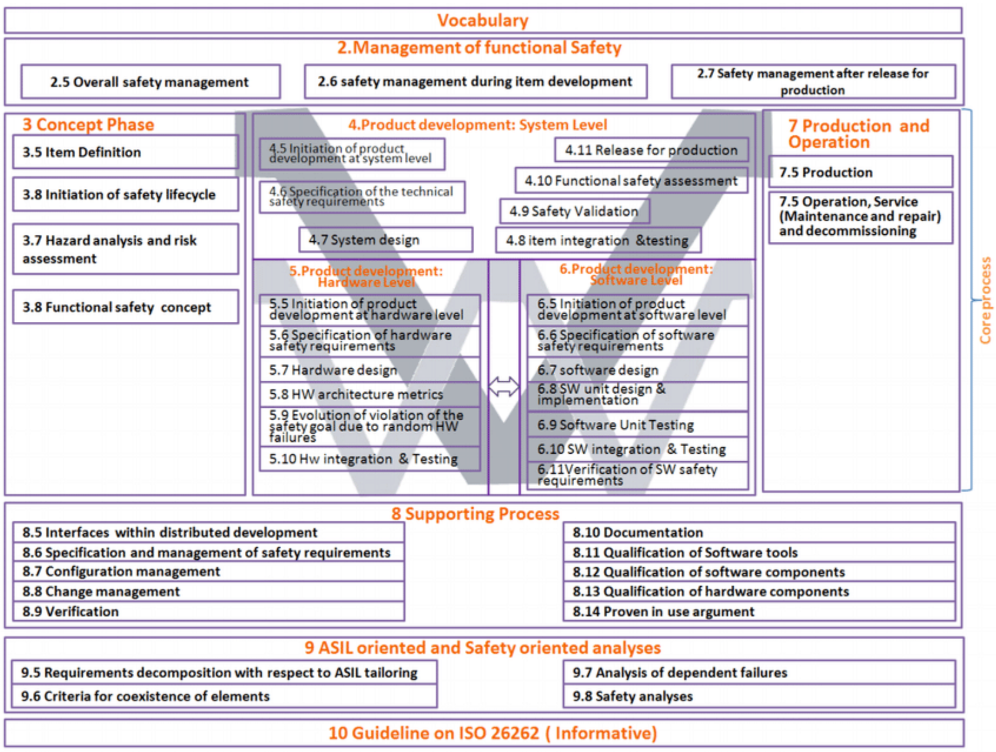

Figure 2.2: ISO 26262 - Functional Safety procedure

ISO 26262 is an international standard that ensures the functional safety of electrical/electronic (E/E) systems in road vehicles. It was jointly developed by

Table 2.1: Driver Performance Measures as per SAE J2944

Measure	Unit	Description
Steering Wheel Angle	[rad]	The angle of the vehicle's steering wheel relative to its neutral position.
Yaw Rate	[rad/s]	The rate of change in the vehicle's heading angle, expressed in radians per second.
Steering Movement Time	[s]	The duration from the onset of a steering response to the event until the steering action is completed for vehicle trajectory correction.
Steering Reversal Rate	[1/min]	The frequency of directional changes in the steering wheel per minute.
Steering Entropy	[J/K]	A measure of the unpredictability in steering behavior, calculated by comparing the frequency distribution of steering angle errors against a baseline under increased task load.
Lateral Position	[cm]	The horizontal distance from a point on the vehicle to the lane boundary, measured at a right angle to the path of travel.
SD of Lateral Position	[cm]	A statistical metric representing the variability of the vehicle's lateral position in relation to the center of the lane.
Lane Departure	[s]	The period during which any portion of the vehicle exits its travel lane until it either returns entirely to the lane or stops, not including intended lane shifts or turns.
Jerk	[m/s^3]	The temporal rate of change in acceleration, denoting the third derivative of position with respect to time.

the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). The V-model is a software development model that illustrates the relationship between the phases of the development process. In the context of developing software for automotive applications, the ISO 26262 V-model offers guidance on how to integrate functional safety techniques into the software development process. The V-model phases can be used to identify the functional safety activities that need to be carried out in each phase of the development process. The development of driving distraction recognition software

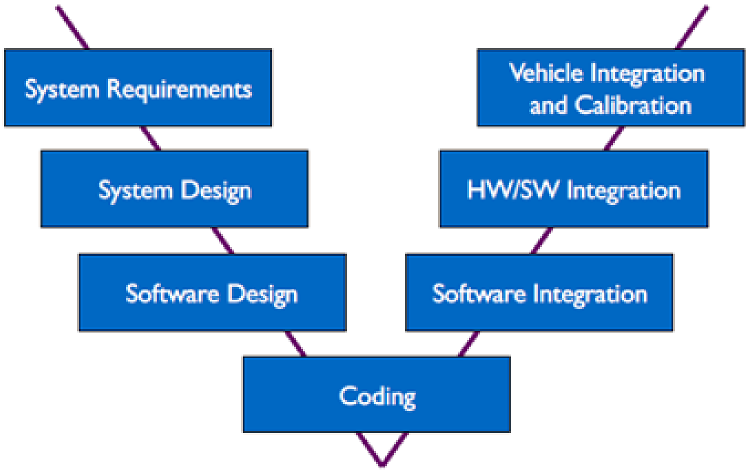

Figure 2.3: V-model development process

is an important initiative in road safety and the automotive industry. This software aims to identify signs of driver distraction, such as cell phone use, lack of attention to the road or unsafe behavior while driving, in order to prevent accidents and improve overall road safety. This research of thesis fits into the V-model during the early phase of "system design" and "software development", trying to meet the requirements imposed by the latest European regulations described in section 2.2.3.

2.2.3 ADAS

Regulatory oversight of Advanced Driver Assistance Systems (ADAS) is vital for ensuring these technologies enhance road safety without introducing new hazards. Key international standards include the UNECE framework and SAE's J3016 levels of driving automation. In the US, the NHTSA sets safety standards, while in the EU, directives are harmonized with input from entities like the ACEA. Crucial to this regulatory landscape are ISO norms such as ISO 26262 for functional safety and ISO/PAS 21448 (SOTIF) for non-malfunctioning behavior of ADAS.

Continuous updates in regulations keep pace with technological progress, requiring manufacturers to stay compliant and prioritize safety.

ADDW

The European Union has set forth regulations to ensure the integration of advanced safety systems in vehicles, including systems designed to mitigate driver distraction. One such system is the 'advanced driver distraction warning' (ADDW), which is mandated to aid drivers in maintaining focus on driving tasks and alert them in instances of detected distraction [13].

In accordance with Commission Delegated Regulation (EU) 2023/2590 [14], the project's distraction recognition system was designed to meet specific technical and operational requirements. These requirements are essential to ensure the system's effectiveness in enhancing driver safety by mitigating distraction-related risks. The key requirements addressed in the system design include:

- **Operational Efficiency** The system is engineered to detect instances when the driver's visual attention deviates from the driving task. Specifically, the system activates automatically when the vehicle's speed exceeds 20 km/h. This automatic activation is critical to ensuring that the system is operational in most driving conditions, thereby enhancing its utility and effectiveness in promoting driving safety.

- **Minimization of False Positives** A critical aspect of the system's design is its capability to distinguish between genuine instances of driver distraction and benign activities that do not compromise driving safety. This requirement is vital to minimize false positives, thereby reducing unnecessary distractions and potential desensitization to system alerts.

- **Privacy and Data Protection** The system adheres to stringent privacy and data protection standards by operating without relying on personal biometric data from vehicle occupants other than the driver. It is designed to function effectively using non-personal data, ensuring compliance with the EU's data protection regulations.

- **System Availability and Manual Override** The system is configured to activate automatically at speeds above 20 km/h, with provisions for manual deactivation by the driver to accommodate specific driving scenarios or preferences. This feature allows for flexibility while ensuring that the system is active during most driving conditions.

- **Environmental Adaptability** Recognizing the diverse conditions under which vehicles operate, the system is designed to function reliably both during the day and at night. This adaptability ensures consistent performance in various lighting conditions, enhancing its utility across different times of the day.

- **Comprehensive Monitoring** The system's monitoring capability extends to various areas of interest within the vehicle's cabin and the driver's line of sight. It is capable of detecting prolonged gaze fixation on non-driving related areas (such as the vehicle's infotainment system or mobile devices) and issuing timely alerts to redirect the driver's attention to the road.
- **Interface Design** The human-machine interface of the system employs a combination of visual, auditory, and tactile alerts to ensure that warnings are effectively communicated to the driver. The design of these alerts is in line with the regulation's specifications to maximize perceptibility and prompt an appropriate response. In this project only visual and auditory alerts has been considered.

By incorporating these requirements into the design and development of the distraction recognition system, the project not only aligns with the regulatory framework established by the EU but also sets a benchmark for safety in automotive technology.

Acceptance Criteria The acceptance criteria for the distraction detection system are not explicitly defined in the referenced regulations. In order to establish a comprehensive and effective evaluation framework, the following criteria, inspired by the DDAW acceptance standards, have been considered:

- **Sensitivity and specificity thresholds:** The system is considered effective if it achieves sensitivity and specificity values equal to or greater than the predefined thresholds for Advanced Driver Distraction Warning ADDW systems in at least 95% of the sample size tested [13]. This criterion ensures that the majority of instances of potential driver distraction are accurately identified and appropriately addressed by the system.

  The average sensitivity S is over 40% defined as:

  $$\bar{S} = \frac{\sum_{i=1}^{n} S_n}{n} \geq 40\% \tag{2.1}$$

  Where n is the number of participants, the sensitivity S for each participant is calculated as follows:

  $$S = \frac{n(TP)}{n(TP) + n(FN)} \cdot 100\% \tag{2.2}$$

  where:

  - $n(TP)$ true positive: both the system and the driver correctly identify that the driver is distracted;
  - $n(FN)$ false negative: the system fails to identify that the driver is distracted;
  - $n(FP)$ false positive: the system incorrectly identifies the driver as distracted;
  - $n(TN)$ true negative: the system correctly identify that the driver is not distracted.

- **Average performance and control of variation:** The system is also considered effective if the average sensitivity and specificity across all subjects meet or exceed the ADDW thresholds, with minimal variance in sensitivity between subjects. This criterion emphasises the importance of consistent system performance across different drivers, reducing the likelihood of discrepancies in system alerts and ensuring reliable driver support.

In simulated testing conditions, the regulation mandates a 5% reduction in the acceptance threshold for average sensitivity and a 2.5% decrease for its 90% confidence interval.[2] By following these criteria, the project aims to deliver a distraction detection system that meets regulatory expectations and industry best practices for driver support systems. This approach facilitates a balanced assessment of the system's ability to accurately detect distracting events (sensitivity) while minimising false alerts (specificity), thus improving overall driving safety and user confidence in the system's capability.

Testing procedure

The designated areas for the validation of the software defined by regulations [13], particularly for evaluating driver distraction, are defined as follows:

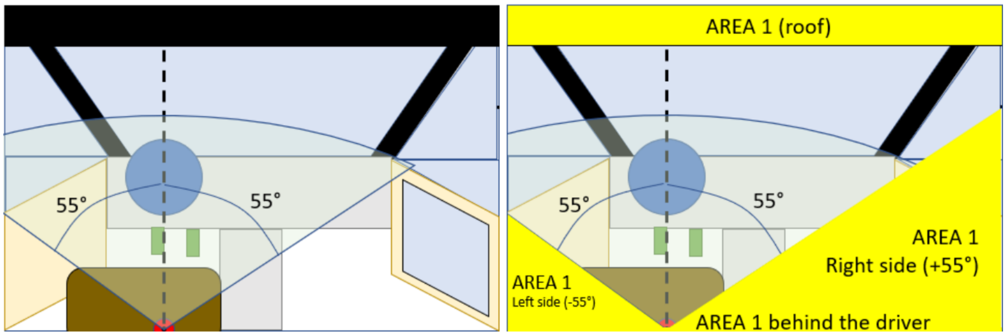

Figure 2.4: Distraction Areas defined by regulations [14]

Area 1: This area encompasses two primary zones:

- The vehicle's roof, acknowledging overhead distractions.
- Regions outside the driver's direct forward line of sight (considered at 0° orientation), delineated by two vertical planes rotated +55° to the right and −55° to the left, relative to the vehicle's longitudinal axis. These planes intersect at the driver's ocular reference point, emphasizing peripheral distractions.

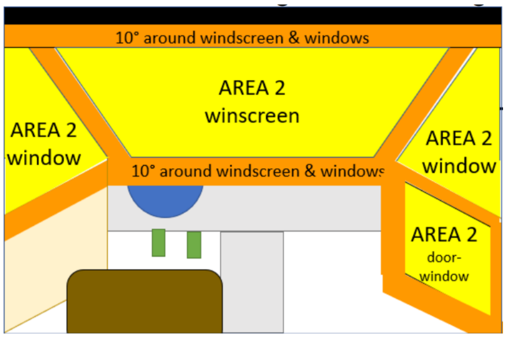

Figure 2.5: Distraction Areas defined by regulations [14]

Area 2: Comprises the following components:

- The windscreen and window areas, focusing on potential distractions through direct and peripheral vision.
- An extension of 10° surrounding the windscreen and windows, as viewed from the ocular reference point, to cover broader visual distractions.

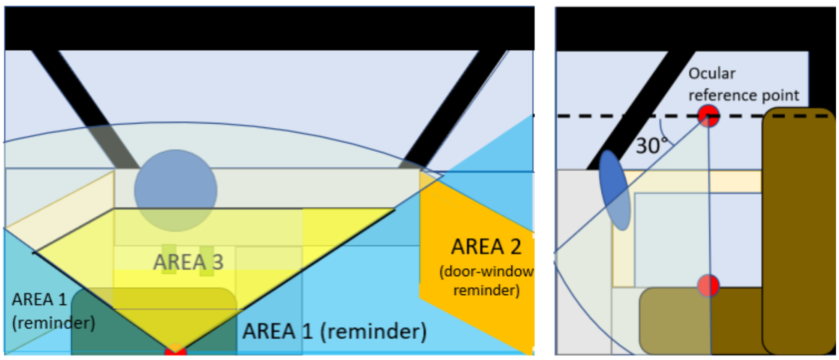

Figure 2.6: Distraction Areas defined by regulations [14]

Area 3: Defined by the space below a 30° downward angle from the ocular reference point, Area 3 is further refined through its relationship with Areas 1 and 2:

- Excludes any section falling within Area 1 by default to maintain a clear boundary for overhead distractions.

- Vehicle manufacturers have the discretion to incorporate parts of Area 1 into Area 3, allowing customization based on specific vehicle design and ADDW system considerations.

- Excludes all zones encompassed by Area 2, ensuring that forward-view distractions are distinct from those below the ocular line.

It is crucial to note that the demarcation and interaction between these areas are established from the perspective of the driver's ocular reference point. The initial mapping of one area onto another utilizes angular coordinates to accurately reflect the driver's field of view. Following this projection, spatial coordinates may be employed to concisely describe the defined areas, facilitating a more straightforward understanding and analysis of potential distraction sources within the vehicle environment.

Advancements in vehicular safety are increasingly propelled by the integration of distraction detection systems (DDS). These systems, underpinned by sophisticated algorithms, are designed to identify and mitigate risks associated with distracted driving. This section introduces the various types of ADDW solutions and their implementations, including camera-based systems, sensor-less technologies, and biometric monitoring.

2.2.4 Types of ADDW Solutions

ADDW solutions can be broadly classified into the following categories:

Camera-Based Systems: These systems utilize in-cabin cameras to monitor the driver's eye movement, head position, and overall alertness. They are capable of issuing real-time feedback to alert the driver if signs of inattention are detected.

Sensor-Less Technologies: Some solutions analyze the driver's interaction with the vehicle, such as steering patterns or braking behavior, without relying on direct physiological measurements.

Biometric Monitoring: Employing wearable devices or in-cabin sensors, these systems measure physiological indicators, such as heart rate or head pose, to infer the driver's attention level.

2.2.5 Commercial Solutions by OEMs

Commercially, several OEMs have integrated various forms of ADDW systems to enhance the safety features of their vehicles. Examples include:

- **General Motors (GM)**: GM's Driver Attention System leverages camera technology to detect drowsiness and inattention in the driver's behavior. « The Driver Attention System uses the Driver Monitoring System Control Module and Driver Monitoring System Camera to monitor the driver»[15].

- **Nissan**: Nissan's Intelligent Driver Alertness (I-DA) system is designed to monitor steering behavior to identify signs of inattention or fatigue. By analyzing the patterns and 'roughness' of steering inputs and comparing them to a standard driving pattern, the system can detect when the driving behavior deviates due to potential drowsiness. If erratic steering is observed at speeds above 60 kph (37 mph) [16], I-DA issues an audible alert and a visual message, suggesting the driver take a break, enhancing safety through proactive driver engagement.

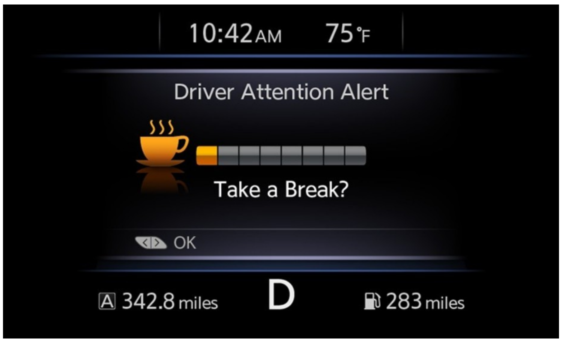

Figure 2.7: Nissan's Driver Attention Alert

- **Volvo**: Volvo has integrated advanced driver monitoring technology developed by SmartEye in their new Volvo EX90. This new system employs camera-based monitoring to evaluate driver alertness and detect signs of distraction, enhancing vehicle safety by ensuring driver engagement [17].

DMS in this example is managed using in cabin monitoring with camera, extracting the following features:

- **Face Recognition:** To identify the driver and potentially customize vehicle responses based on the driver's profile.

- **Occupancy:** To detect the presence of passengers and their positions within the vehicle.

- **Driver and Passenger State:** To assess the alertness and posture of both the driver and the passengers, which can be indicators of distraction or fatigue.

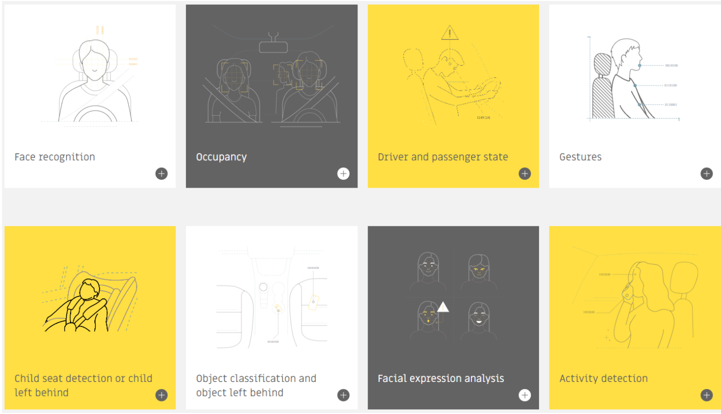

Figure 2.8: In cabin monitoring features used for detecting distraction (SmartEye for Volvo)

- **Gestures:** To interpret hand signals or other gestures that might control certain in-vehicle systems or indicate the driver's intentions.
- **Child Seat Detection or Child Left Behind:** To ensure that child occupants are safely seated and to alert if a child is left unattended in the vehicle.
- **Object Classification and Object Left Behind:** To recognize objects within the vehicle and remind occupants not to leave belongings behind.
- **Facial Expression Analysis:** To gauge the driver's mood and level of concentration through facial expressions.
- **Activity Detection:** To monitor and analyze the activities of the driver and passengers, which can include interactions with in-vehicle systems or with other occupants.

OEMs have taken significant steps in implementing ADDW technologies, with each bringing a unique approach to driver safety. As these technologies continue to evolve, they form a critical part of the broader effort to reduce accidents caused by distracted driving. This growth in this fiel is driven by all new regulations that impose on new production car the installation of this systems on the vehicle.

2.2.6 Experimental solutions

Within the exploration of experimental solutions for driver distraction detection, a variety of innovative approaches are being investigated, employing deep learning algorithms, model-based approaches, and hybrid systems. These methods leverage an array of data sources, including vehicular data and driver-specific information gathered from biometrics or camera observations, to enhance the accuracy and reliability of distraction detection systems.

Deep learning algorithms stand out for their ability to process and analyze vast amounts of data, learning complex patterns and behaviors associated with driver distraction. These algorithms can interpret camera feed to recognize facial expressions, eye movements, and head positions indicative of distraction. The strength of deep learning lies in its adaptability and continuous improvement through exposure to new data, making it highly effective in environments with diverse and unpredictable driving conditions.

Model-based approaches, on the other hand, rely on predefined models of driver behavior to assess distraction levels. One such method involves driver model identification, which analyzes the driver's interaction with the vehicle's controls, such as steering patterns, to infer attention levels. This approach is particularly adept at accommodating individual driver differences and can be finely tuned to account for various driving scenarios, thereby minimizing false alarms and enhancing system reliability.

Hybrid systems combine the strengths of machine learning and model-based approaches to provide a comprehensive solution that addresses the limitations of each method when used in isolation. By combining real-time data analysis with robust behavioural models, hybrid systems can provide a nuanced understanding of driver states, accounting for the complex nature of distraction and its manifestations.

As an example of possible implementations, "Driver Attention Level Estimation Using Driver Model Identification" exemplifies a model-based approach by employing driver model identification to estimate attention levels. This method's uniqueness lies in its use of vehicular inertial sensors and steering behavior analysis, sidestepping the need for lane information from cameras and emphasizing adaptability to individual driver differences .

In contrast, "Towards a Context-Dependent Multi-Buffer Driver Distraction Detection Algorithm" introduces the AttenD2.0 algorithm, a hybrid system that expands upon the original AttenD algorithm by incorporating multiple buffers to reflect context-dependent factors and visual time-sharing behavior. This innovative solution adjusts its parameters based on both static and dynamic driving conditions, showcasing its flexibility and applicability in real-world scenarios, particularly in ensuring driver attentiveness in automated vehicle control situations.

These experimental solutions illustrate the diverse and evolving landscape of driver distraction detection research, highlighting the potential of combining different methodologies to create more effective and reliable systems. In my thesis project, I aim to advance the exploration of driver distraction detection by adopting a novel approach inspired by the concept of a nominal driver model, as seen in the study "Identification of a Linear Parameter Varying Driver Model for the Detection of Distraction." This study successfully utilizes a Cybernetic Driver Model (CDM) through the application of an Unscented Kalman Filter (UKF) to recursively identify parameters indicative of driver distraction. The model integrates a deep understanding of human sensorimotor functions into a structured framework, dividing the driver's interaction with the vehicle into components such as visual anticipation, visual compensation, and neuromuscular response .

My project will maintain the foundational idea of using a nominal driver model but will diverge in the method of detection. Instead of solely relying on recursive identification through UKF, I plan to explore additional or alternative data-driven techniques that might offer enhanced sensitivity or specificity in detecting distractions. This could involve leveraging advanced machine learning algorithms or incorporating more granular biometric data to refine the detection of subtle changes in driver behavior or state that signify distraction.

The key motivation behind this approach is to capitalize on the strengths of the nominal driver model, particularly its basis in cybernetic theory which provides a comprehensive representation of driver behavior, while exploring new avenues for the detection phase that could offer improvements in real-time performance and applicability in diverse driving contexts. The goal is to achieve a more adaptable and robust system capable of accurately identifying distractions under various driving conditions and for different driver profiles.

2.3 Cybernetic Driver Model

As discussed in the previous section, understanding driver behavior to enhance road safety in preventing accidents detecting driving distraction represents a crucial aspect of this project. This can be done through the study of Cybernetic Driver Models. Cybernetic Driver Models, with their ability to replicate the complex interplay between a driver's cognitive processes and their interactions with the vehicle and environment, offer a powerful framework for this purpose. By modeling how drivers respond to various stimuli and manage driving tasks, these models help in identifying patterns that signify distraction. This chapter delves into the intricate world of Cybernetic Driver Models, unraveling how they serve as an essential tool for understanding and detecting driver distraction, thereby contributing to the development of safer driving environments.

The core idea behind cybernetic models is to represent the driver's interaction with the vehicle and the environment in a systematic way, often using control theory principles. These models consider the driver as a controller that processes inputs from the environment (e.g., road curvature, traffic density, vehicle dynamics) and generates outputs in the form of steering, braking, or acceleration commands to achieve desired driving goals (e.g., staying in lane, avoiding obstacles, reaching a destination).

Key components of cybernetic driver models typically include:

1. **Perception:** This aspect models how a driver gathers information from the environment, such as visual cues, sounds, and vehicle feedback. It addresses how this information is filtered, prioritized, and processed to inform decision-making.

2. **Decision-making:** This part represents the cognitive processes involved in choosing specific actions based on perceived information, driving goals, and potentially the driver's internal state (e.g., stress level, fatigue).

3. **Action:** This component models the physical actions taken by the driver, such as steering adjustments, braking, and accelerating. It considers the limitations and capabilities of human motor functions.

4. **Feedback Loops:** Cybernetic models often incorporate feedback mechanisms, where the outcome of the driver's actions influences their subsequent perceptions and decisions. This loop is crucial for modeling adaptive and dynamic driving behavior.

2.3.1 Lateral driver model

In the domain of human-vehicle interaction, the studies conducted by Mars and Chevrel [18] have established a comprehensive theoretical foundation that elucidates the complex mechanisms by which drivers perceive, decide, and actuate control over their vehicles. Mars and Chevrel reviews scientific studies on driver modeling emphasizing the importance of representing human visual and motor processes in steering control, incorporating current behavioral science knowledge. The model's structure and calibration using experimental data are key aspects, with applications in estimating driver state. This approach aims to enhance the synergy between human drivers and assistance systems, ensuring safety and efficiency.

They started studying each step for representing the evolution of an action-decision process of a driver starting from visual and motor control, driver behavior, and neuromuscular systems to improve the design and functionality of advanced driver assistance systems (ADAS). By simulating human steering behavior and assessing factors like distraction and visual degradation, these models contribute to the development of more intuitive and effective vehicle control systems that enhance safety and driver-vehicle interaction.

The cybernetic driver model by Mars and Chevrel meticulously dissects the driver-vehicle interaction into three pivotal stages: Vision, Decision-making, and Neuromuscular Action, each playing a distinct role in steering control.

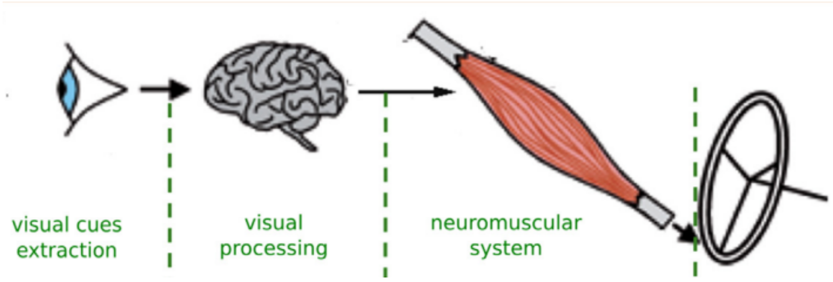

Figure 2.9: Cybernetic Driver Model: breakdown into basic stages

- **Vision:** This stage encompasses the driver's assimilation of environmental cues through visual perception, distinguishing between far visual information for forecasting road curvature alterations and near visual information for real-time corrections of lateral position errors. This dual-level visual processing is supported by numerous studies, underscoring the critical role of visual perception in steering maneuvers. The outcome of this stage is the formation of a steering intention, setting the stage for decision-making and action.

- **Decision-making:** Upon processing the visual information, the driver formulates a steering intention that reflects the cognitive aspect of driving. This intention is shaped by the need to align the vehicle's trajectory with the desired path, influenced by the anticipation of road curvature changes and the immediate need for lateral position correction. The steering intention serves as a vital link between perception and action, encapsulating the driver's response to visual stimuli and facilitating neuromuscular execution.

- **Neuromuscular Action:** The final stage translates the cognitive steering intention into physical motor output through the neuromuscular system. This system adjusts the force applied to the steering wheel based on the steering intention. Inspired by the work of Hoult and Cole [19], the model includes muscle co-activation, incorporating both feedforward and feedback control of movement execution. The output of this stage is the torque applied to the steering wheel, offering a realistic representation of the driver's physical interaction with the vehicle and directly affecting steering dynamics.

Each stage plays a distinct role in steering control, from the initial perception of environmental cues to the cognitive formulation of steering intention and the final execution of steering adjustments. The research spans from modeling human control of steering to assess driver distraction and the effects of visual degradation on steering control. These studies utilize a combination of anticipatory and compensatory control strategies, sensorimotor dynamics, and linear parameter varying models to capture the complexity of human driving behavior. Key themes include the integration of visual and kinesthetic feedback, the identification of driver model parameters for distraction detection, and the exploration of how visual impairment affects steering control. This analysis emphasizes the need for a multidisciplinary approach in the development of sophisticated driver assistance systems. Such systems aim not only to emulate or augment human driving capabilities but also to discern features indicative of the driver's responses to external road stimuli. This aspect is particularly crucial for this thesis, which focuses on detecting driving distractions, because it helps adding information about the intention of the driver and recognising functional drift that not corresponds to the nominal behavior.

Detailed Analysis of the Cybernetic Driver Model

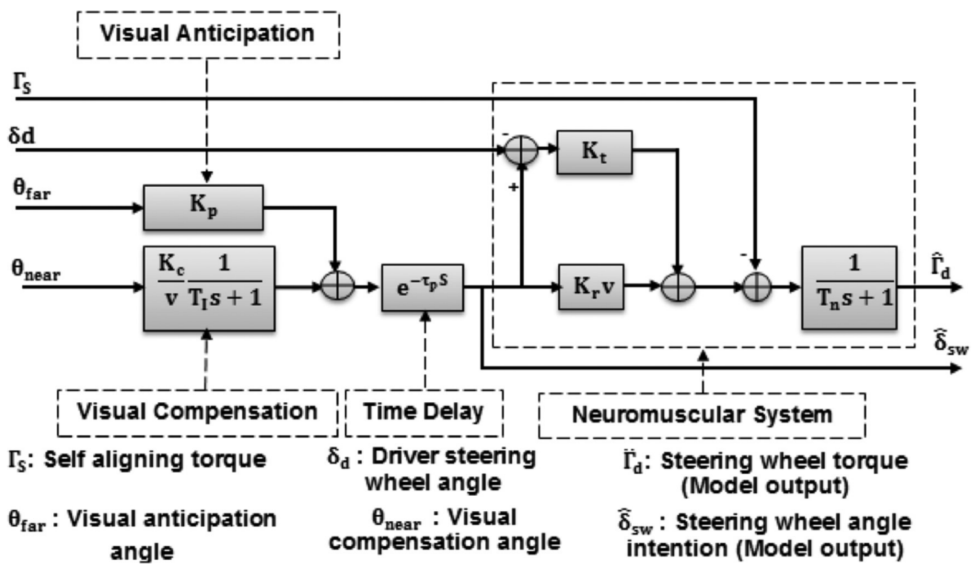

Figure 2.10: The cybernetic driver model (Mars et al., 2011; Saleh et al 2011)

In the cybernetic driver model conceptualized by Mars and Chevrel, the interactions between a human driver and the vehicle are encapsulated through a sophisticated framework. Figure 2.10 delineates the Cybernetic Driver Model, elaborated with parameters specified in section 2.3.1. This paradigm divide driver interaction into two sequential operations: the computation of a prospective steering-wheel angle derived from a two-point visual model, and the attainment of this angle via the vehicle's neuromuscular subsystem (NMS), which incorporates a slight processing latency.

Central to this approach is the two-point visual model, as depicted in 2.11, which integrates modules for both Visual Anticipation and Visual Compensation, following the groundwork laid by Savlucci & Gray [20]. It leverages the dual-point visual feedback obtained during driving from driver perspective: the 'far point' representing anticipatory vision towards road curvature changes, and the 'near point' indicating lateral road awareness close to the vehicle. The far point is elucidated through the road's tangent point data, obtainable from on-board cameras or direct visual assessment. The near point, situated a few meters ahead of the vehicle, reflects the driver's immediate visual field related to lateral lane positioning. The angles formed by the vehicle's trajectory with these points, θ_{far} and θ_{near}, respectively, serve as inputs for the visual model.

The entire model is articulated through various inputs, outputs, and parameters that work together to replicate the driver's steering behavior. The following subsections provide an in-depth examination of the model's main components.

Model Inputs The inputs to the model are critical as they initiate the control process:

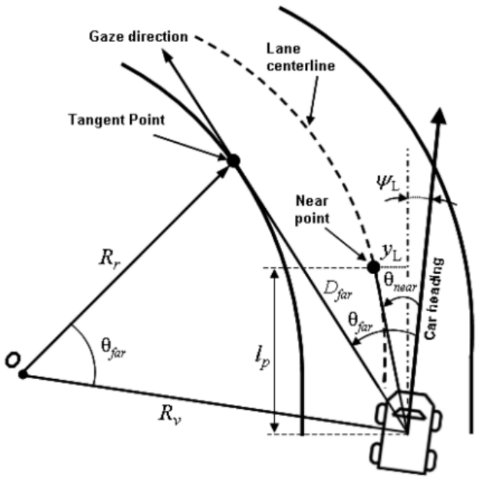

Figure 2.11: Visual representation of angles θ_{far} and θ_{near}

- $\boldsymbol{\theta_{\text{far}}}$ **- Visual Anticipation Angle**: This angle is used by the driver for visual anticipation, which involves observing the distant road to forecast upcoming changes in road curvature. The angle between the car heading and the tangent point, which is where the inside edge line appears to reverse from the driver's viewpoint, is particularly significant for visual anticipation. It is conceptualized that drivers use the angle θ_{far} to "read" the road curvature at the sensorimotor level, allowing for anticipatory adjustments in steering to navigate bends and turns.

- $\boldsymbol{\theta_{\text{near}}}$ **- Visual Compensation Angle**: The θ_{near} angle is utilized for the short-term corrections of the vehicle's lateral position. This is typically based on seeing road edge lines through peripheral vision, just a few meters ahead. It is associated with the immediate compensatory actions required to maintain the vehicle within the lane boundaries. This compensatory module allows the driver to regulate perceptual variables to minimize the difference between the desired vehicle state and the one that would be achieved if the current steering action was maintained.

  Both angles are processed to generate a desired steering wheel angle, δ_{sw}, which is then translated into actual steering commands by the neuromuscular system. The driver model accounts for visual information processing delays, represented by a time-delay term, τ_p, which approximates the time required for a driver to process visual cues and formulate a motor response. The model's ability to predict and compensate using these visual angles facilitates the design of advanced steering assistance systems that can operate harmonously with human drivers, maintaining the natural sensorimotor control loop.

- Γ_s **- Self-aligning torque:** It represents the mechanical feedback from the steering wheel, influenced by road and vehicle dynamics.

- δ_d **- Driver steering wheel angle** the actual angle at which the driver holds the steering wheel.

Model Outputs The outputs of the model are the results of the control process and represent the driver's response:

- $\hat{f}_d$: Steering wheel torque (model output), which is the simulated force that the driver applies to the steering wheel.

- δ_{sw}: Steering wheel angle intention (model output), which is the desired steering wheel angle computed by the model.

Model Parameters The parameters of the model define its behavior and response characteristics:

- K_p: Gain associated with visual anticipation, affecting how the visual information of the far environment influences steering.

- K_c: Gain associated with visual compensation, affecting the corrective steering based on immediate lateral position errors.
- T_s: Time constant associated with the self-aligning torque, representing the dynamics of the vehicle's response to steering.
- K_r: Gain related to the steering wheel's resistance, influenced by the vehicle's speed.
- T_n: Neuromuscular time constant, representing the delay and dynamics of the driver's neuromuscular system.
- τ_p: Processing delay, indicative of the time taken by the driver to process visual information and react.

Each component plays an important role in the accurate representation of the human driver's control strategy, enabling the development of advanced driving assistance systems that are attuned to the driver's natural responses.

Model adaptation to speed variation The dependency on vehicle speed is reflected in multiple components of the model:

- The compensatory action characterized by K_c is inversely proportional to vehicle speed, suggesting a decreased reliance on near visual cues at higher speeds.
- Steering system resistance, denoted by K_r, is directly influenced by speed, which impacts the force feedback experienced by the driver through the steering wheel.
- Neuromuscular response time, represented by T_N, indicates the latency in the driver's response, which also varies with the speed of the vehicle.

These elements collectively indicate a model that is sensitive to changes in speed, dynamically adjusting both the perception of visual cues and the execution of motor commands for steering control.

System Identification of the model The model depicted in Fig. 2.10 can be represented in the state-space framework as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \Pi) = \mathbf{A}(\Pi)\mathbf{x}(t) + \mathbf{B}(\Pi)\mathbf{u}(t) \tag{2.3}$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), \Pi) = \mathbf{C}\mathbf{x}(t) \tag{2.4}$$

where $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$ is the state vector, $\mathbf{u} = \begin{bmatrix} \theta_{\text{far}} & \theta_{\text{near}} & \delta_d & \Gamma_s \end{bmatrix}^T$ is the input vector, and $\mathbf{y} = \begin{bmatrix} \hat{\Gamma}_d & \hat{\delta_{\text{sw}}} \end{bmatrix}^T$ is the output vector. Π is the vector containing the parameters to be identified: $\Pi = \begin{bmatrix} K_p & K_i & T_i & \tau_p & K_r & K_t & T_n \end{bmatrix}^T$. The function $\mathbf{f}$ is a real analytic vector field on $\mathbb{R}^3$ and $\mathbf{g}$ is a real analytic vector field on $\mathbb{R}^2$. Once the time delay $e^{-T_p s}$ is replaced by a first-order Padé approximation, one gets from Fig.1 the state variables:

$$x_1 = \frac{K_c}{v}\frac{1}{T_i s + 1}\theta_{\text{near}} \tag{2.5}$$

$$x_2 = \frac{1}{1 + \tau_p s}(K_p\theta_{\text{far}} + x_1) \tag{2.6}$$

$$x_3 = \frac{1}{T_n s + 1}\left[(K_r v + K_t)x_2 - K_t\delta_d - \Gamma_s\right] \tag{2.7}$$

In the continuous time domain, we have:

$$\dot{x}_1 = -\frac{1}{T_i}x_1 + \frac{K_c}{vT_i}\theta_{\text{near}} \tag{2.8}$$

$$\dot{x}_2 = \frac{1}{\tau_p}(x_1 - x_2 + K_p\theta_{\text{far}}) \tag{2.9}$$

$$\dot{x}_3 = (K_r v + K_t)\frac{1}{T_n}x_2 - \frac{1}{T_n}x_3 - \frac{K_t}{T_n}\delta_d - \frac{1}{T_n}\Gamma_s \tag{2.10}$$

In the continuous time space it will be represented by: The matrices $A(\Pi)$, $B(\Pi)$, and C are defined as follows:

$$\begin{aligned}
A(\Pi) &= \begin{bmatrix} -\frac{1}{T_i} & 0 & 0 \\ \frac{1}{T_p} & -\frac{1}{T_p} & 0 \\ 0 & (K_r v + K_t)\frac{1}{T_n} & -\frac{1}{T_n} \end{bmatrix}, \\
B(\Pi) &= \begin{bmatrix} 0 & \frac{K_c}{vT_i} & 0 & 0 \\ \frac{K_p}{T_p} & 0 & 0 & 0 \\ 0 & 0 & -\frac{K_t}{T_n} & -\frac{1}{T_n} \end{bmatrix}, \\
C &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.
\end{aligned} \tag{2.11}$$

Assuming that the inputs are approximately constant during two consecutive sample times, the discretized model corresponding to e continuous time state-space model is given by:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{f}_d[\mathbf{x}_k, \mathbf{u}_k, \Pi] \\ \mathbf{y}_k = \mathbf{g}[\mathbf{x}_k, \mathbf{u}_k, \Pi] \end{cases} \tag{10}$$

where $\mathbf{f}_d[\mathbf{x}_k, \mathbf{u}_k, \Pi] = \mathbf{x}_k + T\mathbf{f}[\mathbf{x}_k, \mathbf{u}_k, \Pi]$, $\mathbf{x}_k = \mathbf{x}(kT)$, $\mathbf{u}_k = \mathbf{u}(kT)$, $\mathbf{y}_k = \mathbf{y}(kT)$ and T is the sample time.

2.4 Driving Simulators

Driving simulators are invaluable tools in the realm of automotive research and development, offering varying levels of complexity and immersivity. They enable researchers and engineers to simulate and analyze a wide array of driving scenarios, vehicle dynamics, and driver behaviors in a virtual yet realistic setting.

2.4.1 Complexity Levels in Driving Simulators

Driving simulators range from simple desktop-based applications focused on basic vehicle handling and maneuvering, to sophisticated full-scale systems that replicate the driving experience with high fidelity. For instance, low-complexity simulators like the basic versions of *TORCS (The Open Racing Car Simulator)* provide fundamental insights into vehicle handling. In contrast, high-complexity systems like *VIRES Virtual Test Drive* offer a comprehensive suite for simulating advanced driver-assistance systems (ADAS) and autonomous driving technologies in detailed virtual environments.

2.4.2 Immersivity in Driving Simulators

The level of immersivity in driving simulators is another crucial aspect, impacting the realism of the simulated driving experience and the validity of research outcomes. Immersivity ranges from 2D graphical interfaces to full 3D setups with motion platforms, surround sound, and panoramic visual systems. High-immersivity examples include *Daimler's Driving Simulator* in Fig. 2.12, which is one of the most advanced simulators, providing a 360-degree visual field and a motion platform capable of simulating various driving conditions realistically [21].

The selection of a driving simulator is heavily influenced by its intended application, balancing the need for complexity and immersivity against economic and practical considerations. While high-complexity, high-immersion simulators offer the most realistic and detailed research environment, their cost and operational requirements may not be justifiable for all applications. Conversely, simpler simulators may provide a cost-effective solution for basic research, or initial stages of development as in this project.

Figure 2.12: Daimler-Benz Driving Simulator

2.5 Simulation Environments

Simulation environments are the backbone of driving simulator systems, providing the necessary tools and frameworks for creating, executing, and analyzing simulations.

2.5.1 Matlab/Simulink

Matlab/Simulink stands out as a versatile and powerful environment for simulating and modeling dynamic systems for automotive applications. It is particularly valued for its comprehensive set of tools and libraries that cater to various aspects of vehicle dynamics and control system design.

Vehicle Dynamics Blockset

The *Vehicle Dynamics Blockset* in Matlab/Simulink includes fully assembled reference application models that simulate vehicle dynamics in a 3D environment. This blockset is instrumental in developing, testing, and fine-tuning algorithms related to vehicle dynamics, such as traction control, braking, and stability systems, in a virtual environment before real-world implementation.

The *conventional vehicle model* encapsulates a complex and technical representation, which includes a detailed account of the vehicle's physical characteristics, its control systems, and the assortment of sensors, as depicted in Figure 2.13 . Each component of the model is composed of variable subsystems that can be adjusted to meet specific requirements:

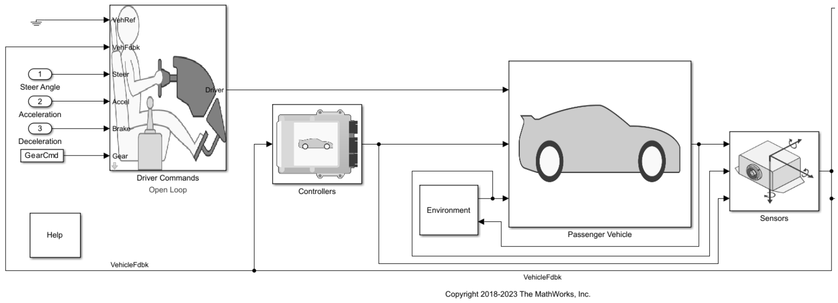

Figure 2.13: Conventional Vehicle Model - Matlab R2023B [22]

Driver Commands: Incorporates various driver models including open-loop, linear predictive, and longitudinal models for driver behavior. Controllers: Simulates essential vehicle controls such as the Electronic Control Unit (ECU), braking, differential, and transmission systems. Environment: Enables the simulation of external conditions like wind effects and surface friction. Passenger Vehicle Dynamics: Models the intricate dynamics of the drivetrain, engine, and vehicle body. Sensor Suite: Equips the model with a triaxial Inertial Measurement Unit (IMU). Visualization Tools: Provides integration with default maps from the Unreal Engine environment (refer to Section 4.16). Regarding the vehicle model's degrees of freedom (DOF), the toolbox presents two options: a 7 DOF model and an advanced 14 DOF model. The simpler version accounts for the body's three degrees of freedom and one for each wheel. The more complex model expands upon this by offering six degrees of freedom for the vehicle body and two for each wheel.

Widely recognized in both industry applications and academic research, the 14 DOF model in Fig.2.14 effectively balances the intricacies of real-world physical behavior and the mathematical constructs needed for more idealized scenarios (see [23]). The practical implementation of this model within Simulink has been validated as an accurate and reliable tool for vehicular research studies, as referenced in [24].

Automated Driving Toolbox

The *Automated Driving Toolbox* provides algorithms and tools for designing, simulating, and testing ADAS and autonomous driving systems. It includes features for sensor fusion, object detection, and path planning, facilitating the development of complex driving algorithms and the integration of sensor systems in a virtual simulation environment.

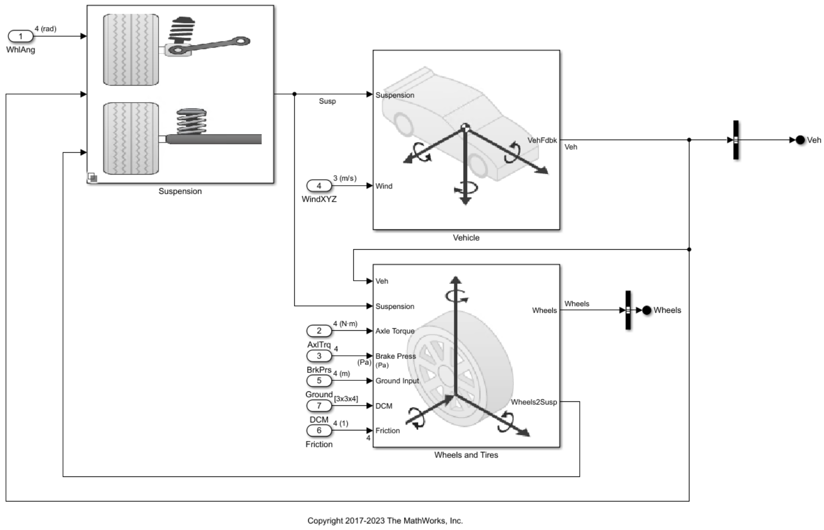

Figure 2.14: Matlab 14DoF Vehicle Model

2.5.2 Integration with Unreal Engine 4.0

The integration of Matlab/Simulink with *Unreal Engine 4.0* enhances the simulation environment by adding high-quality graphical rendering capabilities and expansive scenario customization options. This integration allows for the simulation of complex and realistic driving scenarios, including variable weather conditions, diverse traffic situations, and intricate urban and rural environments. By utilizing Unreal Engine's powerful rendering engine, researchers can create immersive and visually compelling simulations that improve the predictive accuracy and reliability of automotive research outcomes. The combination of Matlab/Simulink with its specialized toolboxes and the graphical prowess of Unreal Engine 4.0 enables highly realistic and complex simulations essential for advancing automotive safety and autonomous driving technologies.

Chapter 3

Background

This section explores the background knowledge, tools, and techniques utilized in the execution of this project.

3.1 Driving Simulator Hardware

The company's driving simulator, designed for previous projects, is equipped with specialized hardware to support software simulations. It includes a high-performance workstation, realistic control interfaces like a steering wheel and pedals for natural driver interaction, and a high-definition, curved main monitor for immersive simulation. Additional elements like a secondary screen for detailed simulation data, along with essential accessories like a seat designed for comfort and realism, contribute to a comprehensive setup. Future enhancements may include tools for assessing driver attention and generating distractions for data collection on driver behavior.

3.1.1 Main components

- **Control Device: Logitech G920 Driving Force** - High-precision steering wheel and pedals for a realistic driving simulation 3.1.
- **Cockpit: RSeat RS1 Red/Black** - Ergonomic racing simulator cockpit designed for comfort and realism.3.2
- **Display: Nilox Curved Monitor ELED 49" DFHD** - Immersive 49-inch curved monitor for an expansive field of view.3.3
- **Graphics Card: NVIDIA RTX 3070 GPU** - High-performance graphics card for rendering complex simulations.

Figure 3.1: Steering wheel and pedals

Figure 3.2: Cockpit

- **CPU: AMD Ryzen 7 5800x** - Powerful CPU to handle demanding simulation tasks.
- **Memory: Corsair Vengeance LPX 32GB DDR4 3200MHz** - High-speed memory for efficient data processing.
- **Storage: Samsung SSD 980 M.2 PCIe NVMe 1TB** - Fast storage solution for quick load times and large simulation data.

3.2 Vehicle Setup & Environment

The vehicle setup encompasses the various parameters to optimize the car's performance for specific applications, such as racing or comfort. This process involves fine-tuning mechanical resistances, sensitivities, and electronic control unit (ECU)

Figure 3.3: Curved monitor 49"

parameters to achieve the desired dynamic behavior. Environmental factors like air temperature and pressure also play a significant role in vehicle performance and must be considered during the setup process. In virtual models, these environmental conditions can be defined to simulate specific scenarios.

3.2.1 Driveline

The driveline constitutes an essential mechanism within a vehicle, tasked with the transmission of power from the engine to the wheels, thereby facilitating motion. This system encompasses key components including the steering geometry, transmission, and driveshaft, each playing a pivotal role in the vehicle's performance and efficiency.

This knowledge is utilized when making assumptions for the inputs of cybernetic models explained in section 4.2.3, in order to keep high fidelity with literature models. This approach is used to enhance the credibility of the simulations.

3.2.2 Sensors

In the era of advanced driver-assistance systems (ADAS) and the move towards autonomous vehicles, sensors have become indispensable in modern cars. These sensors enable vehicles to be more aware of their surroundings and include technologies such as radar, lidar, cameras, and ultrasonic sensors. Each type of sensor contributes to the vehicle's ability to navigate, detect obstacles, and enhance overall safety and performance.

The integration and calibration of these sensors are crucial in the vehicle setup process, ensuring the ego vehicle can accurately perceive its surroundings and respond appropriately.

The vehicle setup, therefore, involves a holistic approach, integrating mechanical adjustments with sensor calibration to tailor the vehicle's performance to specific needs and environmental conditions.

3.3 Kalman Filters

Kalman Filters constitute a series of mathematical equations that provide an efficient computational (recursive) means to estimate the state of a process in a way that minimizes the mean of the squared error. The standard Kalman Filter is designed for linear models, but real-world systems are often nonlinear, which led to the development of the Extended and Unscented Kalman Filters. Kalman filters, including their Extended (EKF) and Unscented (UKF) variants, are invaluable tools in state estimation and parameter identification for dynamic systems. These filters are particularly beneficial for estimating parameters of cybernetic models, such as the one proposed by Mars and Chevrel for driver steering behavior.

3.3.1 Extended Kalman Filters

The Extended Kalman Filter (EKF) serves as a robust tool for estimating the states of nonlinear dynamic systems, such as those encountered in cybernetic driver models. The cybernetic model encapsulates complex, nonlinear relationships between a driver's sensory processing and motor actions. The EKF addresses the challenge of linearization in these nonlinear environments by approximating the system's state at a point and expanding the nonlinearities around this estimate.

In the context of a cybernetic driver model, the EKF can estimate not only the immediate state of the driver-vehicle interaction but also infer hidden parameters that govern these interactions. This is achieved by augmenting the state vector with these parameters, thereby enabling the EKF to update its estimates as new data arrives.

Mathematical Description of the Extended Kalman Filter The EKF uses a two-step process: a prediction step, where the system's state is advanced from time $k-1$ to k, and an update step, where the prediction is corrected by the new measurement at time k. The equations governing these steps are as follows:

Prediction Step:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}) \tag{3.1}$$

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1} \tag{3.2}$$

where:

- $\hat{x}_{k|k-1}$ is the predicted state estimate.
- $f(\cdot)$ is the nonlinear state transition function.
- u_{k-1} is the control input at time $k-1$.

- $P_{k|k-1}$ is the predicted error covariance.
- F_{k-1} is the Jacobian matrix of $f(\cdot)$ with respect to the state.
- Q_{k-1} is the process noise covariance matrix.

Update Step:

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{3.3}$$
$$K_k = P_{k|k-1} H_k^T S_k^{-1} \tag{3.4}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - h(\hat{x}_{k|k-1})) \tag{3.5}$$
$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{3.6}$$

Where:

- S_k is the innovation covariance.
- H_k is the Jacobian matrix of $h(\cdot)$ with respect to the state.
- R_k is the measurement noise covariance matrix.
- K_k is the Kalman gain.
- y_k is the measurement at time k.
- $h(\cdot)$ is the nonlinear measurement function.
- I is the identity matrix.

EKF Advantages:

1. Computationally less intensive.
2. Extensively studied and understood in practice.
3. Requires less memory for operation.

EKF Disadvantages:

1. Can introduce significant errors in highly nonlinear systems due to linearization.
2. Requires the computation of Jacobian matrices, which adds to the complexity for certain models.

By using the EKF within the cybernetic driver model's framework a more responsive and accurate representation of the driving behavior can be obtained considering all non linear behaviours of the model.

3.3.2 Unscented Kalman Filters

For complex cybernetic models describing driver behavior, accurately estimating the state is paramount. The Unscented Kalman Filter (UKF) provides a powerful alternative to the EKF for such non-linear systems. The UKF applies the unscented transformation to predict and correct the state estimates, which is often more accurate than the linearization approach used in the EKF, particularly when dealing with highly non-linear systems.

Mathematical Description of the Unscented Kalman Filter The UKF predicts the state and the variance of a non-linear system by taking a deterministic sampling approach. A set of points, called sigma points, are chosen so that their mean and covariance match that of the system's current state estimate. These points are then propagated through the non-linear system, and their new mean and covariance are computed to form the prediction. The steps can be described mathematically as follows:

Sigma Points Selection:

$$\mathcal{X}_{k-1} = \text{GenerateSigmaPoints}(\hat{x}_{k-1|k-1}, P_{k-1|k-1}) \tag{3.7}$$

Prediction Step:

$$\mathcal{X}_{k|k-1} = f(\mathcal{X}_{k-1}) \tag{3.8}$$

$$\hat{x}_{k|k-1} = \text{Mean}(\mathcal{X}_{k|k-1}) \tag{3.9}$$

$$P_{k|k-1} = \text{Covariance}(\mathcal{X}_{k|k-1}) + Q_{k-1} \tag{3.10}$$

Update Step:

$$\mathcal{Y}_k = h(\mathcal{X}_{k|k-1}) \tag{3.11}$$

$$\hat{y}_{k|k-1} = \text{Mean}(\mathcal{Y}_k) \tag{3.12}$$

$$P_{yy} = \text{Covariance}(\mathcal{Y}_k) + R_k \tag{3.13}$$

$$P_{xy} = \text{CrossCovariance}(\mathcal{X}_{k|k-1}, \mathcal{Y}_k) \tag{3.14}$$

$$K_k = P_{xy} P_{yy}^{-1} \tag{3.15}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}) \tag{3.16}$$

$$P_{k|k} = P_{k|k-1} - K_k P_{yy} K_k^T \tag{3.17}$$

Where:

- $\mathcal{X}_{k-1}$ represents the sigma points at time $k-1$.
- $f(\cdot)$ is the non-linear state transition function.

- $h(\cdot)$ is the non-linear measurement function.

- Q_{k-1} and R_k are the process and measurement noise covariance matrices, respectively.

- $\hat{x}_{k|k-1}$ and $\hat{x}_{k|k}$ are the a priori and a posteriori state estimates, respectively.

- $P_{k|k-1}$ and $P_{k|k}$ are the a priori and a posteriori estimate covariances, respectively.

- P_{yy} and P_{xy} are the measurement and cross-covariance matrices, respectively.

- K_k is the Kalman gain.

- y_k is the actual measurement at time k.

The UKF's ability to accurately capture the posterior mean and covariance without linearizing the process and measurement models makes it particularly suitable for systems with complex interactions and non-linearities, such as cybernetic driver models.

UKF Advantages:

1. Better at handling nonlinearities by using sigma points instead of linear approximations.

2. Potentially more accurate in systems with strong nonlinear characteristics.

UKF Disadvantages:

1. More computationally demanding, potentially unsuitable for real-time or resource-constrained environments.

2. Implementation complexity due to the management of sigma points.

3. Higher memory requirements due to storing multiple sigma points.

Given these considerations, the choice between EKF and UKF is done condidering that the systems that we want to obtain require simplicity and have limited computational resources because it should be implemented on the edge and EKF may be the appropriate choice.

3.4 Machine Learning Algorithms

Machine learning algorithms have revolutionized the way we approach problem-solving in various domains, including driver safety and vehicle automation. These algorithms allow computers to learn from data, identify patterns, and make decisions with minimal human intervention. One crucial application of machine learning is in the detection of driver distraction, a leading cause of accidents on the road. By analyzing a combination of vehicular data, environmental conditions, and driver behavior, machine learning models can alert systems to distracted driving, facilitating timely interventions to prevent accidents.

3.4.1 Supervised Machine Learning

Supervised machine learning is a subset of machine learning where the model is trained on a labeled dataset. This means that each training sample is paired with an output label. In the domain of driver distraction detection, supervised machine learning plays a pivotal role. The task of identifying whether a driver is distracted is essentially a classification problem, which is a quintessential application of supervised learning. Classifiers are algorithms that, once trained on a dataset where the 'distraction' states are labeled, can accurately predict the distraction state for new, unseen data.

The process involves feeding the classifier a set of features extracted from the data, which may include variables such as steering angle, braking patterns, eye movement, and even physiological signals like heart rate. The classifier then learns from this data, understanding the complex, multi-dimensional relationships that correlate with distraction.

Once the model is trained, it can then be applied to real-world scenarios. As it encounters new data, it uses the learned patterns to predict the driver's attention state. This prediction can trigger alerts or interventions to refocus the driver, thereby mitigating risks and enhancing road safety.

The effectiveness of these classifiers is critical since they are expected to function accurately in real-time, ensuring immediate and appropriate responses to prevent potential accidents caused by distraction.

Random Forest Classifier

One powerful supervised learning algorithm is the Random Forest Classifier, which is an ensemble learning method. It operates by constructing a multitude of decision trees during the training phase and outputting the class that is the mode of the classes of the individual trees. Random forests correct for decision trees' habit of overfitting to their training set, providing a more generalized and robust prediction.

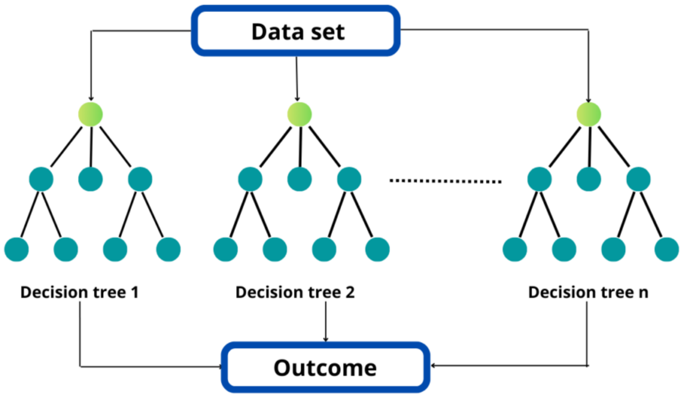

Figure 3.4: Logic of Random Forest Classifier

The Random Forest Classifier works particularly well for distraction detection tasks due to its ability to handle large datasets with numerous features, some of which might be irrelevant. It also provides a measure of feature importance, which can be insightful for understanding which factors contribute most to driver distraction. The key steps in the Random Forest methodology are:

1. **Bootstrap Aggregating (Bagging):** For each decision tree, a random sample of the data is selected with replacement, known as a bootstrap sample. Each tree is trained on its respective bootstrap sample.

2. **Random Feature Selection:** When splitting a node during the construction of a tree, a random subset of the features is considered for the split, rather than all features. This ensures diversity among the trees and is a key difference from a single decision tree.

3. **Building Multiple Trees:** Many trees are built independently using the above two methods of bagging and random feature selection. The number of trees built (`n_estimators`) is a hyperparameter of the algorithm.

4. **Majority Voting or Averaging:** For classification tasks, each tree in the forest votes for a class, and the class with the most votes becomes the model's prediction. In regression tasks, the average prediction across all trees is used.

5. **Model Output:** The final prediction output of the Random Forest Classifier is thus the aggregated result of many decision trees, which typically improves predictive accuracy and controls overfitting compared to using a single decision tree.

This classifier's ability to produce accurate and interpretable results, without the need for extensive data preprocessing, makes it an ideal choice for real-world applications like distraction detection systems in vehicles, where real-time processing and decision-making are crucial.

Support Vector Classifier (SVC)

Support Vector Classifier (SVC) is a powerful supervised learning algorithm used for classification and regression tasks. In the context of distraction detection, SVC excels by finding the hyperplane that best separates the feature space into distracted and non-distracted drivers. It handles both linear and non-linear data by using different kernel functions, thus providing flexibility in modeling complex relationships.This versatility is achieved through the utilization of various kernel functions, such as:

- **Linear Kernel:** Best suited for linearly separable data, where a straight line (or hyperplane in higher dimensions) can separate the classes.
- **Polynomial Kernel:** Allows the model to fit non-linear datasets by mapping the original features into a higher-dimensional space, where a hyperplane can then be used for separation.
- **Radial Basis Function (RBF) Kernel:** Also known as the Gaussian kernel, this is a popular choice for non-linear data, capable of handling complex relationships by mapping the features into an infinite-dimensional space.
- **Sigmoid Kernel:** Mimics the behavior of a neural network's activation function and can be used for non-linear problems.

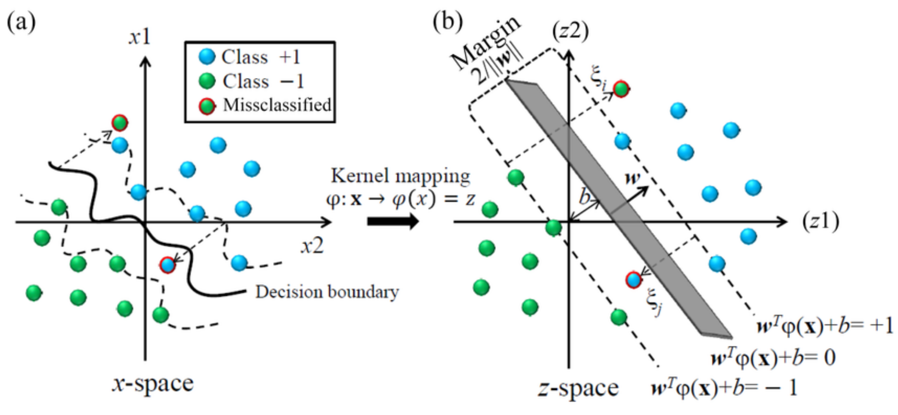

Figure 3.5: Support Vector Machine principle

In the context of distraction detection, the choice of kernel and its parameters (such as the degree for polynomial kernel, or gamma in RBF kernel) are crucial for the model's performance. The flexibility to tune these parameters makes SVC a powerful tool in modeling the intricate dynamics between various features that indicate driver distraction.

Furthermore, SVC introduces the concept of a 'margin' around the separating hyperplane, aiming to not only separate the classes but to do so in a manner that maximizes the distance between the closest points of the classes to the hyperplane. This concept, known as the *maximum margin classifier*, enhances the model's generalization capabilities.

The penalty parameter C plays a pivotal role in SVC by regulating the trade-off between achieving a low error on the training data and maintaining a wide margin. A higher value of C puts more emphasis on minimizing the training error, which can lead to a more complex model at the risk of overfitting. Conversely, a lower value of C prioritizes a wider margin and a simpler model, potentially at the expense of higher training error.

Given its robustness and adaptability, SVC stands out as an efficacious algorithm for discerning driver distraction, thereby contributing to the advancement of safety measures in automotive technology.

XGBoost

XGBoost stands for Extreme Gradient Boosting and is an efficient implementation of gradient boosting framework. This algorithm is known for its speed and performance and is widely used in machine learning competitions. XGBoost provides a robust way to handle a variety of data types, distributions, and the relationships between features, making it an excellent choice for identifying patterns that indicate distraction.

AdaBoostClassifier

The AdaBoostClassifier, short for Adaptive Boosting, is another ensemble technique that works by combining multiple weak learners into a strong one. Each learner in the sequence is adjusted to correct the mistakes of the previous one. In driver distraction detection, AdaBoost can incrementally improve the identification of nuanced behaviors associated with distraction.

GradientBoostingClassifier

Gradient Boosting Classifier is a predictive algorithm that builds an ensemble of decision trees in a sequential manner. Each tree attempts to correct the errors of the previous one based on the gradient of the loss function. Its strength lies in its ability to model complex relationships within data, which is crucial when different forms of distraction may manifest through diverse driving behaviors.

3.5 Time Series and Signal Processing

A time series is a sequence of data points collected or recorded at successive time intervals. The data points in a time series can represent a wide variety of phenomena, tracking changes over time in fields such as economics, meteorology, social sciences, and more. Time series analysis involves understanding these trends, seasonal variations, cyclical patterns, and other characteristics inherent in the data.

3.5.1 Discrete vs. Continuous Time Series

Time series can be categorized into discrete and continuous, based on the nature of their time intervals:

- **Discrete Time Series:** In a discrete time series, the data points are recorded at specific and often equally spaced time intervals. This type of series is common in scenarios where measurements can only be taken or are only meaningful at certain times, such as daily stock market closes or monthly unemployment rates.

- **Continuous Time Series:** A continuous time series, on the other hand, involves data that is recorded continuously over a period of time. This type of series could represent data like temperature measurements taken every millisecond or the fluctuating speed of a vehicle.

The choice between discrete and continuous time series depends on the nature of the data and the objectives of the analysis. In practice, even data from continuous processes are often discretized for analysis due to the limitations of data recording and storage technologies.

3.5.2 Signal Processing in Time Series Analysis

Before feeding time-windowed data into ML algorithms for classification tasks, applying signal processing techniques can be highly beneficial in terms of performances. Signal processing involves the analysis, manipulation, and interpretation of signals. It helps in enhancing the signal quality, extracting relevant features, and reducing noise, thereby making the data more amenable to analysis. The reasons for applying signal processing before classification include:

- **Noise Reduction:** Real-world data is often contaminated with noise. Signal processing can help filter out irrelevant or spurious noise, highlighting the underlying signal patterns.
- **Feature Extraction:** Certain characteristics of the signal, such as peaks, trends, and periodicity, might be crucial for the classification task. Signal processing techniques can extract these features, making them more explicit to the classifier.
- **Normalization and Standardization:** Signal processing can ensure that the data fed into the classifier is normalized or standardized, improving the classifier's performance by providing data in a consistent format.
- **Dimensionality Reduction:** High-dimensional data can be problematic for classifiers. Signal processing can reduce the dimensionality of the data, retaining only the most informative features.

In essence, preprocessing time series data through signal processing enhances the data's quality and structure, facilitating more accurate and efficient classification. This preparatory step is crucial for effective time series analysis, particularly when dealing with complex or noisy data sets.

3.5.3 Sliding Window

The sliding window technique is a crucial method employed in this project, particularly for processing sequential data or time-series information. In order to apply the inference model for distraction detection it is necessary to define two parameters: the window size and the window stride (or overlap).

Window Size

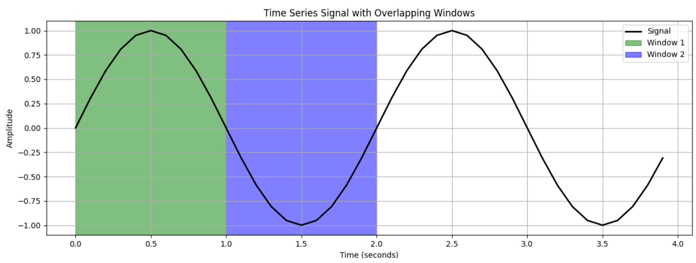

Figure 3.6: Time Window data segmentation

The window size determines the length of the data segment being considered at any given time. It is a critical parameter that impacts the granularity of analysis.

A larger window size encompasses more data points, providing a broader context but potentially diluting short-term variations. Conversely, a smaller window size offers a more detailed view of fluctuations within the data but might miss broader trends.

Window Stride

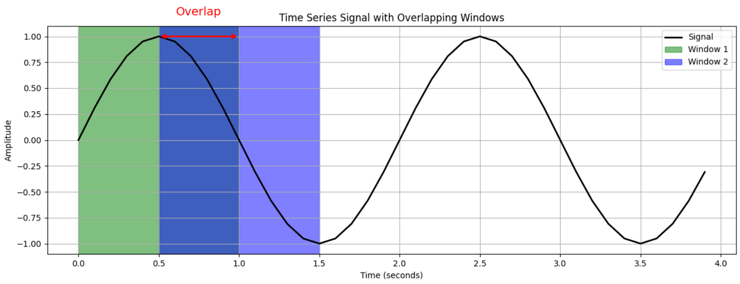

Figure 3.7: Window selection with overlapping

The window stride, or overlap, defines the step size between consecutive windows. A smaller stride increases the overlap between successive windows, leading to finer resolution in the analysis but at the cost of increased computational load. A larger stride reduces the overlap, enhancing computational efficiency but possibly overlooking subtle changes between windows.

The choice of these parameters is pivotal, as it balances the need for detailed data analysis against computational efficiency. In this project, the sliding window technique, with carefully selected window size and stride, facilitates the examination of temporal data, ensuring that the models capture both short-term dynamics and long-term patterns of distracted driving effectively.

3.5.4 Spectral Decomposition

To gain a deeper understanding of a signal, it's beneficial to examine it not just in its original time domain but also in terms of its frequency components. This is where the concept of frequency analysis comes into play, particularly through the application of the Fourier transform. This mathematical transformation allows us to convert a signal from its time domain representation, $s(t)$, into its frequency domain counterpart, $F_s(f)$, according to the following equation:

$$F_s(f) = \int_{-\infty}^{+\infty} s(t)e^{-2\pi ift}\,dt, \quad f \in \mathbb{R}$$

By doing so, the analysis shifts from observing changes over time (t) to understanding how the signal behaves across different frequencies (f).

Fast Fourier Transform The computational efficiency in conducting frequency domain analysis is significantly enhanced by the Fast Fourier Transform (FFT). This algorithm is an optimized version of the discrete Fourier transform (DFT) that quickly identifies the predominant frequencies within a given time window of data. Through FFT, it's possible to detect and analyze periodic patterns and anomalies within the signal, which are crucial for understanding its behavior. The insights gained from observing the frequency characteristics of a signal can reveal underlying structures and events, offering a more comprehensive view of its dynamics.

3.5.5 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a statistical procedure that utilizes an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance, and each succeeding component, in turn, has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. PCA results

- **Dimensionality Reduction:** PCA reduces the dimensionality of the data set, simplifying the dataset while retaining the variation present in the dataset to the maximum extent.
- **Visualization:** It is easier to visualize high-dimensional data sets when reduced to two or three principal components.
- **Noise Reduction:** By keeping only the most significant principal components, minor fluctuations or noise can be eliminated, leading to a more accurate analysis.
- **Feature Extraction:** PCA can be used to discover or reduce the number of variables in a high-dimensional data set.

Mathematical Formulation

The mathematical foundation of PCA involves a few key steps:

1. Standardize the data.
2. Compute the covariance matrix of the data.
3. Calculate the eigenvalues and eigenvectors of this covariance matrix.

4. Sort the eigenvalues and their corresponding eigenvectors.

5. Select k eigenvalues and form a matrix of eigenvectors.

6. Transform the original matrix.

Mathematically, if X is the original data matrix, and V is the matrix with k eigenvectors, then the transformed data Y is obtained as:

$$Y = X \times V \tag{3.18}$$

Applications in Distraction Detection

In the context of distraction detection in drivers, PCA can be utilized to reduce the dimensionality of the dataset derived from vehicle sensors and cameras monitoring the driver or as in this particular case from driver model parameters extracted. This dimensionality reduction can help in isolating the features that are most indicative of distracted behavior, thereby enhancing the performance of machine learning models designed to detect and predict such behaviors.

By focusing on the principal components that capture the most variance in the driver's behavior, the complexity of the predictive models can be reduced without a significant loss of information. This streamlined dataset can lead to more efficient and effective detection of distracted driving, contributing to safer driving environments and reducing the computational efforts in inference for real-time applications.

3.6 AWS Cloud

Figure 3.8: AWS cloud services

The development and deployment of machine learning models for tasks such as distraction detection require robust and scalable computational resources. Amazon Web Services (AWS) Cloud has been employed to store vast amounts of data and to train machine learning models efficiently.

Data Storage with Amazon S3

Amazon Simple Storage Service (S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. For our distraction detection project, Amazon S3 serves as the backbone for storing and retrieving any amount of data at any time. This includes raw data collected from simulations or real-world driving scenarios, pre-processed datasets ready for training, and the trained models themselves. The reliability and security of S3 ensure that our data is stored safely and is accessible whenever needed, facilitating seamless data management throughout the project lifecycle.

Model Training with Amazon SageMaker

Amazon SageMaker is a fully managed service that provides every developer and data scientist with the ability to build, train, and deploy machine learning models quickly. SageMaker offers various built-in algorithms and support for custom algorithms, making it a versatile tool for our distraction detection model development. Our models are trained on SageMaker using the stored data in S3, leveraging its scalable compute resources to expedite the training process. SageMaker's integration with Jupyter notebooks also allows for an interactive development environment, enabling our team to experiment with different models, evaluate their performance, and iterate quickly.

Deployment in Simulation Environment

Once trained, the machine learning models are deployed in a simulated environment for inference. This simulation environment replicates real-world driving scenarios, allowing us to test the effectiveness of our distraction detection models under various conditions. The models deployed in this environment use the trained weights and biases to make predictions on streaming data, simulating real-time inference. AWS provides the necessary tools and services to deploy these models efficiently, ensuring low latency and high throughput for real-time applications.

The integration of AWS services, from data storage with S3 to model training with SageMaker, and finally to deployment in a simulated environment, creates a cohesive and efficient workflow for developing and testing machine learning models aimed at enhancing driver safety through distraction detection.

Chapter 4

Material and methods

4.1 Driving Simulator

4.1.1 Requirements

The proposed solution necessitates the utilization of specific MathWorks toolboxes and versions, outlined in the table below:

Toolbox / Application	**Version / Package**
Matlab	2023b
Simulink	-
Automated Driving Toolbox	Automated Driving Toolbox for Unreal 4.0
Computer Vision Toolbox	-
Image Processing Toolbox	-
Requirements Toolbox	-
Simulink 3D Animation	-
Simulink Test	-
Vehicle Dynamics Blockset	Including the Unreal Interface package
Unreal Engine	4.27
Python	3.10

Table 4.1: Toolboxes and Versions Required for the Proposed Solution

Furthermore, the above-mentioned licenses were implemented via Matlab's Add-On Manager, ensuring seamless integration and compatibility, particularly with the specified version of the Unreal Engine desktop application for effective scenario generation and the specified version of Python compatible with Matlab R2023B (3.9, 3.10, 3.11) [25].

4.1.2 Driving Simulation Environment

The primary objective of developing a highly immersive simulation environment is to reproduce authentic driving responses and behaviors from the driver, facilitating the accurate recording of real distraction events. This immersive setting integrates a wide array of components, both within the simulation framework and external to the vehicle, such as other entities, potential impediments, roadway features, and diverse conditions of weather and illumination.

Matlab default scenario

The MATLAB default Curved Road scenario, as outlined in the MathWorks documentation [26], is particularly suited for collecting data on normal driving patterns. Illustrated in Fig.4.1 , this scenario's unique configuration offers a realistic setting that prompts drivers to exhibit a wide range of natural driving behaviors.

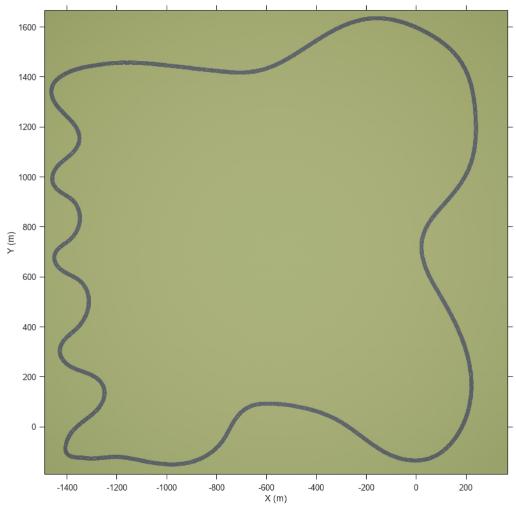

Figure 4.1: Illustration of the MATLAB default Curved Road scenario.

Characterized by its looped road with varying bends, the Curved Road scenario is essential for simulating real-world driving conditions that include both rapid and gradual cornering. This variety is crucial for reproduce a wide range of driver responses, from straightforward steering adjustments to more intricate navigational strategies. The scenario's capacity to provoke such a broad array of driving behaviors is a key factor in its effectiveness for data acquisition purposes, especially in research focusing on driver distraction. The resultant data sets are rich in typical driving dynamics, thereby providing a solid foundation for analytical studies aimed at enhancing road safety and driver assistance technologies.

Enhanced Scenario using Unreal Engine

Building upon the default Curved Road scenario provided by MATLAB, significant enhancements are introduced to create a more comprehensive and realistic driving environment. Utilizing the capabilities of the Unreal Engine editor, the base scenario is augmented to include additional vehicles, thereby introducing dynamic elements that mimic real-life traffic conditions. This modification aims to test and observe driver behavior in scenarios that closely resemble actual driving situations, enhancing the reliability of data acquired for distraction analysis.

The scenario is enriched with environmental details such as buildings, trees, and traffic signals to provide a fully immersive driving experience. These elements not only contribute to the visual realism of the simulation but also play a crucial role in simulating real-world distractions and obstacles a driver may encounter.

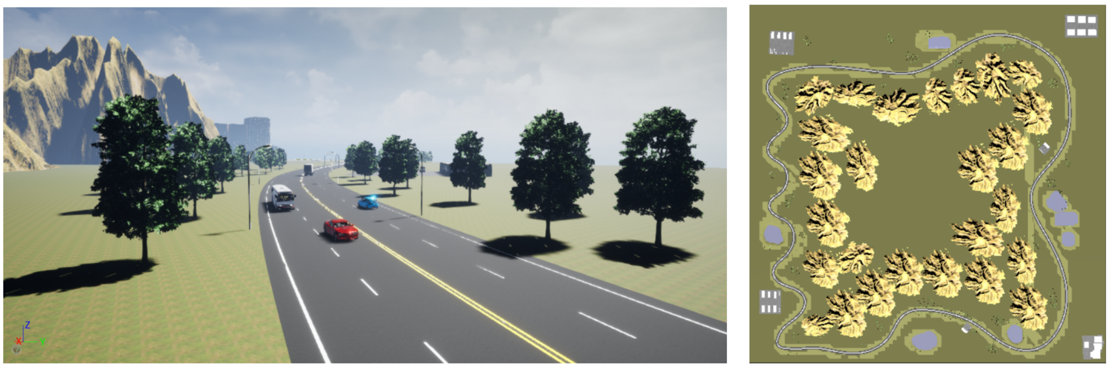

Figure 4.2: Unreal Scenario - Curved Road

To further augment the realism of the simulation and enhance the driver's sensory experience, audio mapping of engine noises was integrated, providing audible cues related to vehicle speed and performance. Additionally, a speedometer was displayed within the driver's dashboard view, offering visual feedback on the vehicle's speed. This setup is particularly effective in conveying the sensation of velocity, an essential factor in driving behavior and decision-making.

A critical feature implemented in this enriched scenario is the alert system on the dashboard. This system is activated when driver distraction is detected, serving as an immediate feedback mechanism to alert the driver and potentially mitigate risky situations. The integration of such alerts within the simulation environment is instrumental in providing in real-time feedback for distracted driving as requested by EU regulations reported in section 2.2.3.

Through these enhancements, the simulation scenario transcends its original design, offering a multifaceted platform for in-depth studies on driver behavior, particularly focusing on the detection and mitigation of distracted driving episodes.

4.2 Driver Model

A key aspect of the project lies in the employment and realization of a cybernetic model that is able to reproduce the driver's behavioral dynamics. The process of feature extraction is done through the application of a Kalman filter in the context of system identification. It is done introducing innovative solutions that stand in parallel to traditional literature. This approach not only challenges existing paradigms but also promotes the exploration of new ideas. In addition, we delineate specific assumptions pertaining to the simulated data employed in our analysis.

Following this foundation, the next section will explore the general structure of the cybernetic model and define the intended outcomes from its application. The focus will be on how this model encapsulates complex driving behaviors and the mechanisms through which it can advance the understanding of driver distractions. This is aimed at contributing to the creation of more intuitive and safer driver assistance systems.

4.2.1 Objectives

The primary objectives of this section include:

1. **Replication of Driver Model from Literature:** To replicate an established driver model for controlling the lateral dynamics found in literature, providing a benchmark for further exploration.

2. **Exploration of Integrative Solutions:** To explore additional solutions that can be integrated with the replicated models to enhance their descriptive and predictive capabilities.

3. **Identification and Selection of Distraction-Related Parameters:** To identify and select parameters within the model significantly influenced by driver distraction for in-depth analysis.

4. **Estimation of Distraction-Related Parameters:** To accurately estimate the values of identified distraction-related parameters using system identification with a Kalman filter.

5. **Convergence Analysis of Parameters:** To examine the convergence behavior of model parameters to ensure the model's stability and reliability.

6. **Development of a Final Solution:** To develop a refined model incorporating insights gained from the objectives above, offering a novel approach to understanding and mitigating driver distraction.

4.2.2 Cybernetic Models: Structural Insights

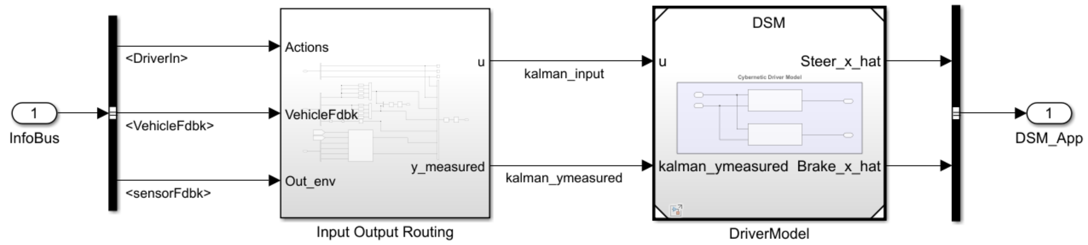

Figure 4.3: Simulink structure for cybernetic models

The Simulink diagram in Fig.4.3 illustrates the general scheme for cybernetic driver models created. The `InfoBus` serves as dispatcher for incoming signals, which are categorized into `<Driver Input>`, representing the driver's inputs or actions on steering wheel and accelerator and brake pedals; `<Vehicle Feedback>`, containing feedback from the vehicle's dynamics; and `<Sensor Data>`, which includes data from environmental sensors mounted on the vehicle in simulation. These signals are routed through the `Input-Output Routing` block, which processes and directs them to the `Driver Models` block.
Two bus signals emerge from this routing block: `kalman_input (u)`, serving as the control input for the Kalman filter, and `kalman_ymeasured (y_measured)`, the observed system measurements.

The `DriverModel` block incorporates two cybernetic driver models:

- **Lateral Driver Model**
- **Longitudinal Driver Model**

Inputs and measurements for kalman filter to estimate the driver's states are collected in the same bus and then, as shown in Fig.4.4, for each model are selected the right signals. Each model produces a bus signal containing state and augmented state signals.

4.2.3 Lateral driver model

In this segment, the implementation of the cybernetic driver model, previously discussed in section 2.3.1, is addressed. The system model is identified using an Extended Kalman Filter (EKF) block, essential for the dynamic estimation of parameters within the augmented state. The EKF is employed as a key mechanism, enabling the continuous adaptation and refinement of model parameters in response to observed driver behavior and vehicle dynamics.

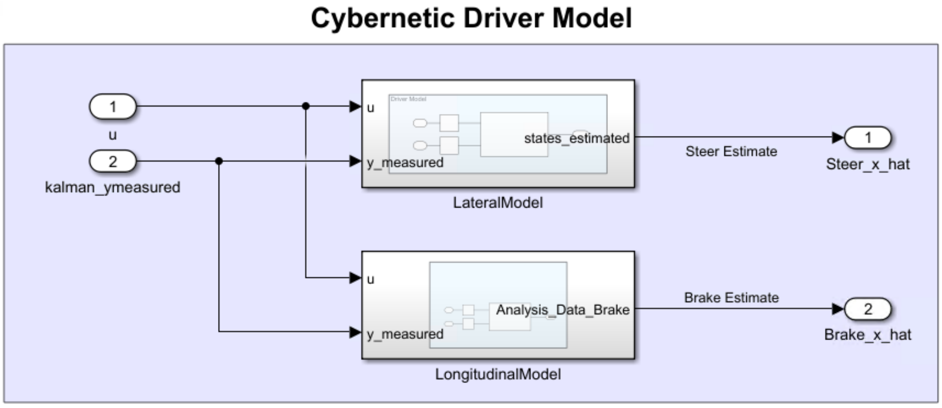

Figure 4.4: Simulink block - Driver models input management

Table 4.2: Inputs for the Cybernetic Driver Model

Input Source	**Input Variable**	**Description**
Driver Input	Steering Angle	The angle at which the driver is holding the steering wheel, indicating the driver's intended direction.
Vehicle Dynamic Feedback	Self Aligning Torque	The torque that aligns the steering with the vehicle's motion path due to tire-road interactions.
Vehicle Dynamic Feedback	Vehicle CoG Velocity	The velocity of the vehicle's center of gravity, essential for understanding the vehicle's motion state.
Vehicle Dynamic Feedback and Sensor Data	Theta_near	Near visual feedback angle from the road, aiding in immediate steering adjustments by the driver.
Vehicle Dynamic Feedback and Sensor Data	Theta_far	Far visual feedback angle from the road, assisting in anticipatory steering adjustments for upcoming road curvature.

Assumption on inputs & output meausured selection

Considering the model descripted in 2.3.1 in order to replicate the same model applying the recursive identification through the extended kalman filter shown in 3.3.1 due to available signals in simulation it is needed to make some assumption. In particular for the measure output "driver's steering intention" δ_{sw} is not measurable in physical terms; however, the use of the actual steering wheel angle δ_{sw} as output during the experiment phase is a common choice in literature (e.g., [27], [28]), and thus this method is adopted.

In the simulated environment, the self-aligning torque Γ_s and the steering wheel torque Γ_d are not directly available. Lacking the full array of parameters to reconstruct these signals, suitable substitutes must be identified. The self aligning torque Γ_s, as measured at the steering column corresponds to the measured torque at the wheels, modulated by a transmission factor and other variations due to the elastic and damping constants of the steering transmission system. This signal can be computed in our simulation environment as the sum of torques acting on the vertical axis of wheels and it is considered equivalent to Γ_s.

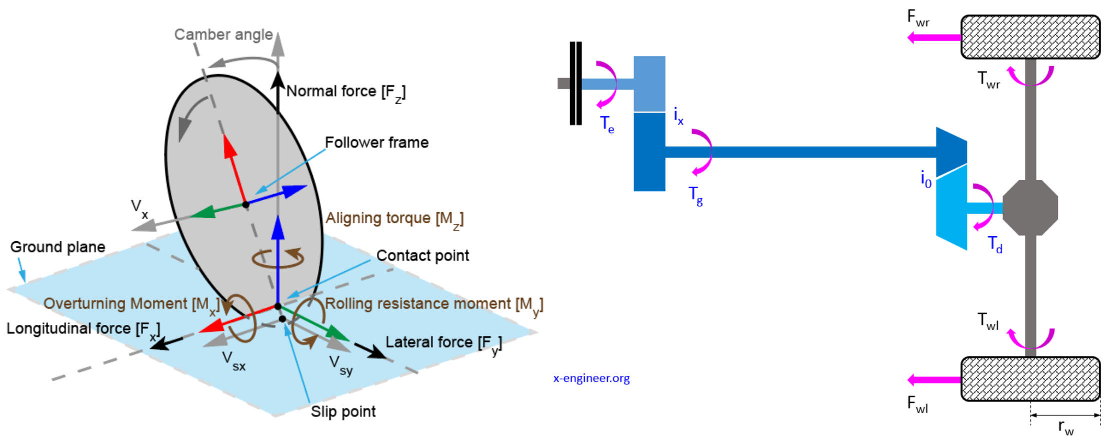

Figure 4.5: Self Aligning Torque acting on each wheel using Matlab Wheel Model and Wheel to Steering Wheel Torque transmission [29]

$$\Gamma_s = M_{z_1} + M_{z_2} + M_{z_3} + M_{z_4} \tag{4.1}$$

It is noted that Γ_d poses a greater challenge for substitution with an appropriate signal, as seen for the desired steering angle that represents the intention of the driver. In the initial stage of modeling, Γ_d is assumed to be equal to Γ_s. This choice is validated as no significant differences were observed between the two phases of the driver distraction detection process and it should be the torque that the driver is applying to counterbalance the torque that he is receiving on the steering wheel plus the additional part that allows him to steer at the desired angle.

Identification and Selection of Distraction-Related Parameters

Within the range of parameters governing the model, some of them has been considered as state variables in the augmented state vector, due to their significant role in distraction detection. To facilitate a clearer understanding and enhance the model's interpretability, these parameters have been assigned human-readable names that describe their function within the driving context:

- **Driver Sensitivity to Lane Margins** (K_c): This parameter quantifies the driver's responsiveness to the lateral distance from the vehicle to the lane margins, reflecting their ability to maintain lane position.
- **Driver Sensitivity to Vehicle/Road Curvature Misalignment** (K_p): It captures the driver's sensitivity to the misalignment between the vehicle's trajectory and the road's curvature, indicating how quickly and accurately a driver adjusts steering to follow the road.
- **Driver Steering Effort** (K_t): This reflects the effort a driver applies to the steering mechanism, which can correlate to their level of engagement or fatigue and potentially their susceptibility to distraction.

These parameters, embedded within the augmented state vector $\mathbf{x}_a$, are pivotal in the process of system identification aimed at detecting driver distraction. By differentiating between parameters that are less and more susceptible to variations due to distraction, the model gains the ability to provide a nuanced understanding of driver behavior under different cognitive loads.

These state variables, fundamental to the augmented state vector $\mathbf{x}_a$, are instrumental in system identification for distraction detection. The augmented state vector is formulated as:

$$\mathbf{x}_a = \begin{bmatrix} \mathbf{x} \\ \Pi \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ K_p \\ K_c \\ K_t \end{bmatrix}^T \tag{4.2}$$

where $\mathbf{x}$ represents the original state variables and Π embodies the parameters less susceptible to distraction-related variation, which have been previously identified and fixed based on literature to stabilize the identification process and cater to the detection mechanism's precision. The human-readable names assigned to the parameters within Π—namely K_c, K_p, and K_t—reflect the driver's sensitivity to lane margins, vehicle/road curvature misalignment, and steering effort, respectively. These parameters are directly associated with the driver's interaction with the vehicle and are thus essential to the analysis of driver behavior under varying conditions of attention and distraction.

In the domain of driver behavior modeling, it is a common practice to hold certain parameters constant to streamline the identification process and ensure the stability of the model. This practice is substantiated by extensive literature indicating that while some parameters exhibit significant variability across different drivers and driving conditions, others remain relatively invariant.

Following this convention, and based on precedents established by seminal studies such as the work of Ameyoe et al. [28], the parameter vector is partitioned into two subsets. Parameters less susceptible to variation due to distraction, such as T_r, τ_p, K_r, and T_n, which are fixed at commonly reported values in the literature (as shown in Table 4.3).

Table 4.3: Parameters less susceptible to distraction

T_r	τ_p	K_r	T_n
0.5	0.4	-0.35	0.04

These assumptions have been made for two reasons:

1. **Numerical Stability:** Fixing these parameters helps prevent numerical instabilities during the system identification phase, which could otherwise compromise the model's robustness and accuracy.

2. **Empirical Evidence:** Studies, including those cited in reference [28], have consistently found no significant variation in parameters like T_n among different drivers, thereby validating the assumption of their constancy.

Consequently, the model focuses on identifying the parameters $\theta = [K_p, K_c, K_t]$ which are more reflective of an individual driver's behavior and susceptibility to distraction. This focused approach enables a more precise analysis of the elements that directly influence driving performance and distraction levels, yielding insights that are crucial for the development of targeted interventions and advanced driver assistance systems.

Observability is a critical attribute for the success of any state estimation technique, particularly within the context of Kalman filtering. It pertains to the system's ability to deduce its internal states from output measurements. The model's nonlinear observability has been substantiated through rigorous validation as noted in previous studies, specifically in reference [27]. This property is crucial when applying a Kalman filter since it relies on the premise that all necessary state information can be captured through external observations. Despite the presence of process noise v_k and measurement noise w_k, which are both assumed to follow a zero-mean Gaussian white noise distribution with respective covariance matrices Q_a and R_a, the system's design assures that the state estimations are robust and reliable.

Convergence Analysis of Parameters & EKF Initialization

For the successful application of the Extended Kalman Filter (EKF), precise initialization of the state covariance matrices and the noise covariance matrices is imperative. Initialization begins with the state vector $\hat{x}_0$ corresponding to $\begin{bmatrix} x_0 & \theta_0 \end{bmatrix}^T$, where θ_0 reflects common values retrieved from literature and then after several iterations.

Attention is then directed toward the state covariance matrix $P_{\theta,0}$ and the noise covariance matrices Q_θ and R_θ. In initializing the state covariance matrix $P_{\theta,0}$, it is posited that the system's actual state $\hat{x}_k$ changes more dynamically in comparison to the parameter vector $\hat{\theta}_k$. This reflects an understanding that the real-time state variables, representing the immediate physical states of the system, are subject to more rapid fluctuations than the parameters, which are typically more steady and evolve slower. Consequently, this leads to an initial state covariance matrix where the entries corresponding to the state variables are larger than those associated with the parameter estimates. This configuration of $P_{\theta,0}$ mirrors the expected behavior of the system's state dynamics, allowing the EKF to place more uncertainty in the estimation of the state variables initially, thereby accommodating their potential for quick changes. On the other hand, the parameters $\hat{\theta}_k$ are presumed to vary less and, hence, have smaller initial covariances. The precise values chosen for the initial covariance matrix are thus a direct consequence of these considerations, tailored to the system's characteristics as determined by empirical evidence and established practice.
It is posited that the real state dynamics $\hat{x}_k$ evolve more rapidly than those of θ_k, hence the first three elements of the main diagonal of $P_{\theta,x}$ are set significantly higher than those corresponding to θ_k, as was previously established in related work [28].

With these considerations in mind, the matrices are initialized as follows:

$$
\begin{aligned}
\hat{x}_{\theta,0} &= \begin{bmatrix} 0 & 0 & 0 & 2 & 4 & 6 \end{bmatrix}^T, \\
P_{\theta,0} &= \begin{bmatrix} \alpha I_{3x3} & 0 \\ 0 & \beta I_{3x3} \end{bmatrix}, \\
Q_\theta &= 10^{-5} \begin{bmatrix} I_{3x3} & 0 \\ 0 & I_{3x3} \end{bmatrix}, \\
R_\theta &= 10^{-3} \begin{bmatrix} 40 & 0 \\ 0 & 1 \end{bmatrix},
\end{aligned}
$$

where I_{3x3} denotes the identity matrix. The scalar multipliers α and β are chosen to yield desired performance, with optimal values found empirically; $\alpha = 30$ and $\beta = 1$.

The EKF proceeds to estimate the augmented state vector $\hat{x}_{a,k}$ at each sample time, refining the state and covariance estimates through its iterative process, which accounts for both the linear and nonlinear aspects of the system dynamics. The covariance matrices Q_0 and R_0 are indicative of the process and measurement noise characteristics and are crucial in achieving reliable parameter convergence. All technicalities regarding the EKF implementation and the reasoning behind the choice of α and β values are further expounded in Appendix A considering all state equations described in subsection 2.3.1.

4.2.4 Longitudinal Driver Model

In this section, we discuss the development of a longitudinal driver model that builds upon the cybernetic model framework commonly referenced in literature for lateral vehicle control. The aim is to extend this established model to predict brake pedal control. We maintain the assumption that the cognitive processing and neuromuscular execution aspects of the driving task remain consistent as in the lateral control model in the previous section 4.2.3.

As in the lateral control model, where anticipation and compensation are crucial elements captured by variables such as θ_{far} and θ_{near}, similar concepts are integrated into the longitudinal domain. The visual processing component, essential for predicting vehicle trajectory and modulating control inputs, is adjusted to incorporate distinct inputs significant to braking. While lateral control predominantly involves visual cues associated with road curvature and lane positioning, longitudinal control considers variables related to the distance and relative velocity of leading vehicles, yet keeping a focus on curvature and lane positioning.

In practice, θ_{far} and θ_{near}, which have been instrumental within the lateral control paradigm, continue to be employed as detailed in 2.3.1 for longitudinal modeling. This consistency is driven by two principal reasons. Firstly, it is an attempt to minimize the introduction of entropy into the overall system. By maintaining a simulation environment devoid of vehicles in the driving lane, the occurrence of distraction during data acquisition remains repeatable, avoiding the generation of random, non-systematic information. This approach is founded on the intent to keep the system as simple as possible, given the project's broad scope. The preference has been to start with a solid base, which then can be iteratively enhanced in various areas. This foundational approach allows for a focused initial analysis, setting the stage for subsequent, more complex developments.

This adaptation necessitates a careful consideration of the inputs to the model to ensure they appropriately capture the predictive and compensatory mechanisms drivers employ when modulating brake pressure. Through this approach, we strive to create a comprehensive model that not only reflects the driver's control strategy but also aligns with the underlying cybernetic principles that govern driver behavior in the context of vehicle dynamics and environmental interaction.

Identification and Selection of Distraction-Related Parameters

Within the range of parameters governing the model, some of them has been considered as state variables in the augmented state vector, due to their significant role in distraction detection. To facilitate a clearer understanding and enhance the model's interpretability, these parameters have been assigned human-readable names that describe their function within the driving context:

- **Driver Sensitivity to distance from leading car(K_c)**: This parameter quantifies the driver's responsiveness to the lateral distance from the vehicle to the lane margins, reflecting their ability to maintain lane position.

- **Driver Sensitivity to the leading car's speed (K_p)**: It captures the driver's sensitivity to the misalignment between the vehicle's trajectory and the road's curvature, indicating how quickly and accurately a driver adjusts steering to follow the road.

- **Driver Brake Effort (K_t)**: This reflects the effort a driver applies to the steering mechanism, which can correlate to their level of engagement or fatigue and potentially their susceptibility to distraction.

Identification and Selection of Distraction-Related Parameters

It is important to note that the first two elements, **Driver Sensitivity to Vehicle Distance (K_c)** and **Driver Sensitivity to Road Curvature (K_p)**, are not utilized in the distraction detection algorithm. The reason for their exclusion is the fact that these parameters will gain importance in scenarios where updates to the assumptions about θ_{far} and θ_{near} are made. Since these updates would incorporate a broader range of environmental interactions and anticipatory driving behaviors, K_c and K_p would then assume more significance in the detection process.

Consequently, while K_c and K_p are foundational to the overall driver model and provide valuable insights into a driver's interaction with the vehicle and road environment, they are presently not integrated into the distraction detection algorithm. Future revisions that expand the algorithm's capability to account for more complex driving scenarios will adopt the inclusion of these parameters.

Table 4.4: Inputs for the Cybernetic Driver Model

Input Source	**Input Variable**	**Description**
Driver Input	Brake pedal Angle	The angle at which the driver is acting on the brake pedal, indicating the driver's intended direction.
Vehicle Dynamic Feedback	Longitudinal Force on Wheels	The force acting on wheels due to acceleration/deceleration of body along the longitudinal axis of the wheels .
Vehicle Dynamic Feedback	Vehicle CoG Velocity	The velocity of the vehicle's center of gravity, essential for understanding the vehicle's motion state.
Vehicle Dynamic Feedback and Sensor Data	Theta_near	Near visual feedback angle from the road, aiding in immediate steering adjustments by the driver.
Vehicle Dynamic Feedback and Sensor Data	Theta_far	Far visual feedback angle from the road, assisting in anticipatory steering adjustments for upcoming road curvature.

Add tables for the description of parameters

4.3 Overall System Workflow

This part represents the main body of the thesis work, focusing on the comprehensive system developed to detect driver distraction and initiate alerts in real time on vehicle's dashboard in simulation. It outlines the sequential stages and integral components that collectively facilitate the monitoring of driver behavior, identification of distraction instances, and the subsequent triggering of alerts. Delving into the operational intricacies of each element, this section serves as the thesis's cornerstone, showcasing the research's depth and the novel approach to mitigating the risks associated with distracted driving.

4.3.1 Overview

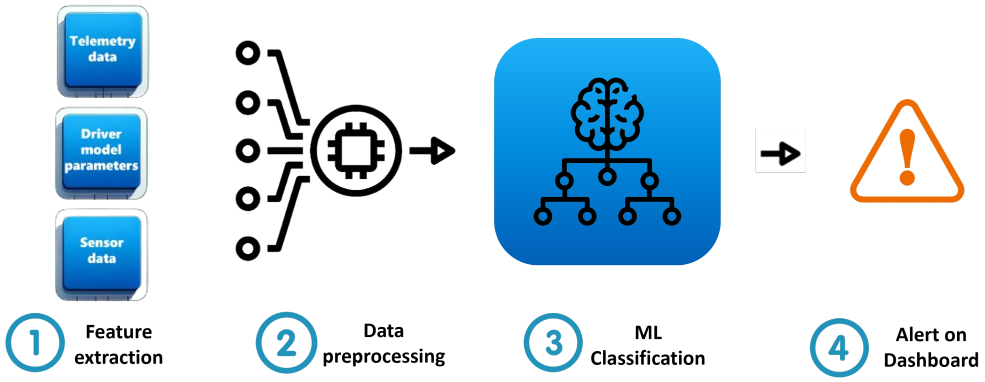

Figure 4.6: System workflow

The comprehensive workflow represented in Fig.4.6 of the algorithm designed for distracted driving detection encompasses several interlinked stages, each critical to the accurate detection and classification of driver behavior and state. The stages are as follows:

1. **Driver Model:** The driver's control strategy is identified through a Kalman filter as explained in section 4.2.3. This model effectively interprets the driver's operational patterns by estimating the latent state variables that determine driving behavior, such as control inputs for the steering wheel and pedals, which reflect the driver's decisions and actions.

2. **Feature Selection:** The next stage involves the extraction of relevant features for analysis. These features include parameters from the augmented state space of the Kalman filter, comprehensive vehicle data, and sensory input data, all of which contribute to creating a multidimensional understanding of the driver's state.

3. **Classification:** The process begins with a preprocessing phase where dimensionality reduction techniques and statistics extraction on time window of data, notably Principal Component Analysis (PCA), streamline the feature set to enhance the efficacy of subsequent classification. In the critical step of detecting driver distraction, the refined data is subjected to a classifier. The optimal classifier is selected from a suite of implemented algorithms, such as the Support Vector Classifier (SVC) and AdaBoost, to accurately distinguish between periods of driver distraction and normal attentive driving within time window considered.

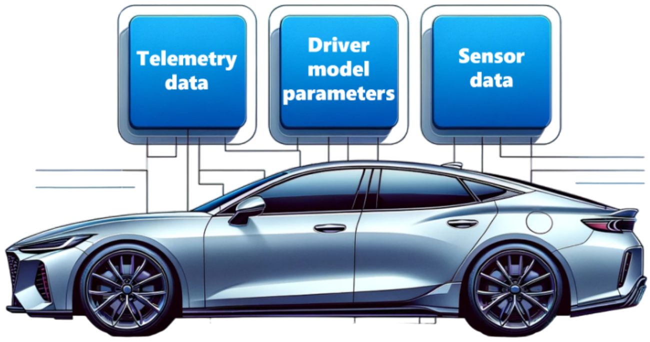

Figure 4.7: Data collected from vehicle telemetry, sensors and driver model

Each of these stages is critical in the overall functioning of the system and is explained in detail in the subsequent subsections.

4.3.2 Timing and Synchronization of System Stages

The effectiveness of the distraction detection algorithm is highly dependent on the precise timing and synchronisation of its individual stages. The stages operate as follows:

1. **Driver Model Data Extraction:** Operating at a constant frequency of 25 Hz, the driver model continuously processes driving behaviour data. This high frequency analysis ensures that even the smallest drifts in driving patterns are captured for a detailed characterisation of the state of the driver.

2. **Data acquisition and buffering:** In parallel with the driver model, vehicle dynamics and sensor data are collected at a general sampling rate of 25 Hz. These data, together with the parameters extracted from the driver model, are stored in a buffer of a size corresponding to the predefined window size (equal to 25 in the acquisition procedure done, sending data to cloud each second) .

3. **Window-Based Inference:** Upon filling the buffer, the algorithm initiates the preprocessing and classification stages. The preprocessed data from each window, encapsulating a specific segment of time, is then analyzed by the inference model. The ML classification model operates with a periodicity

equal to the window size, delivering outputs that classify the driver's state as either distracted or nominal for each evaluated window.

This structured temporal approach ensures that data is not only collected in real-time but is also evaluated methodically, allowing for consistent and timely predictions of the driver's attention state. Each step is meticulously aligned to facilitate a seamless workflow, critical for the dynamic environment of real-time driver monitoring.

4.3.3 Experimental Data Collection

The primary objective of the data collection during the simulation is to gather the necessary information for the training of the Machine Learning (ML) classification algorithm. The data essential for successful training include vehicle data, signals from simulated sensors on the vehicle, and the parameters of the cybernetic models estimated by Kalman filters. For the classifier's training, a ground truth of distraction is also required, which necessitates recording the temporal window of the driver's distraction. To facilitate manual acquisition, a button in the Simulink interface is used, which is pressed when the driver becomes distracted. To ensure the repeatability of the experiment and adhere to regulatory standards for distraction episodes, drivers contributing to the data collection campaign were instructed to look at an object placed in Area 1 defined by the normative guidelines represented in Fig. 2.4. For our data acquisition process, we have chosen to include only Area 1, spanning from $-55°$ to $+55°$ as viewed by the driver with $0°$ directed towards the center of the screen as in the figure below.

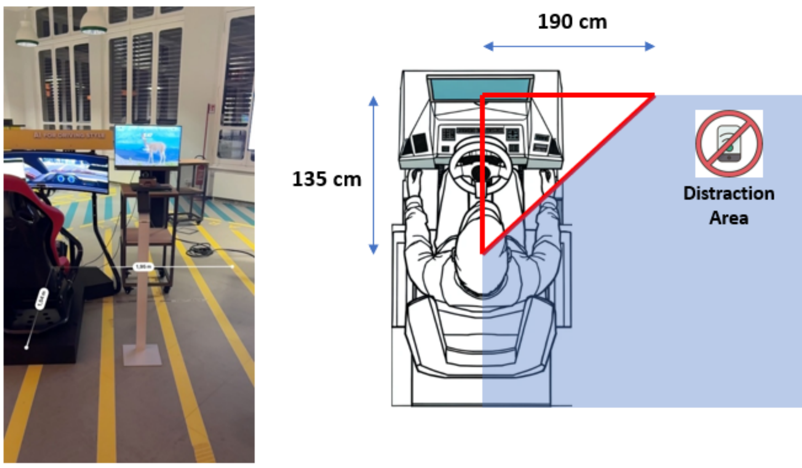

Figure 4.8: Definition of area for distracted episodes

Given $\alpha = 55°$ and $b = 135$ cm, calculate $a \approx 190$ cm as the distance to the center of the steering wheel. This calculation allows us to project the center of the screen and measure the orthogonal distance $a = 190$ cm, in order to define the Area to place the object to see, used for distraction episodes acquisition.

The object is placed within Area 1 as a reference point, dubbed the "Distraction Area" during driving. This setup is crucial for capturing driver behavior in relation to the simulated distractions, providing valuable data for refining the ML algorithm's accuracy in real-world scenarios.

Data Acquisition Procedure

During the data acquisition phase of the simulation, the following steps has been followed:

Explaining Objectives to the Driver

- For inexperienced drivers, conduct preliminary trial runs to familiarize with the simulator.
- Identify the visual distraction object placed within the simulation environment.
- Ensure that the driver performs the expected actions of the simulation naturally, attempting to get distracted as they would on an actual road while maintaining the lane for as long as possible.

Starting the Simulation

- Enter the driver ID (First and Last Name) for driver identification.
- Ensure the button for acquiring ground truth data of distraction is accessible and functional.
- To log a distraction, click once on the button on the control panel on the dashboard (Dashboard shown in figure 4.9).

Verifying Data Transmission to AWS

- Check that the acquired data has been correctly sent to AWS.
- If the data acquisition is deemed invalid, remove it from the system.

Figure 4.9: Distraction ground truth acquisition button in dashboard

Saving Simulation Information Record the following Notes:

- HASH Identifier Associated with the Driver: To ensure the privacy and security of the drivers' personal data, each driver is assigned a unique HASH identifier. This approach anonymizes the driver's identity when transmitting data to the cloud for processing and analysis. The use of HASH identifiers is a crucial privacy-preserving measure, preventing any direct or indirect identification of individuals from the dataset. This methodology aligns with best practices for data protection and privacy, especially in scenarios where sensitive information is collected and stored remotely.

- Duration of the driving session.

- Notes on the driving session (e.g., any technical issues encountered or information on episodes of driver distraction).

Data Collection and Storage Protocol

The protocol for data collection and storage is designed to efficiently handle and analyze data from various sources including Cybernetic Driver Model estimation, sensor data, driver actions, and vehicle signals as shown in Fig.4.11 in the orange box.

The process is as follows:

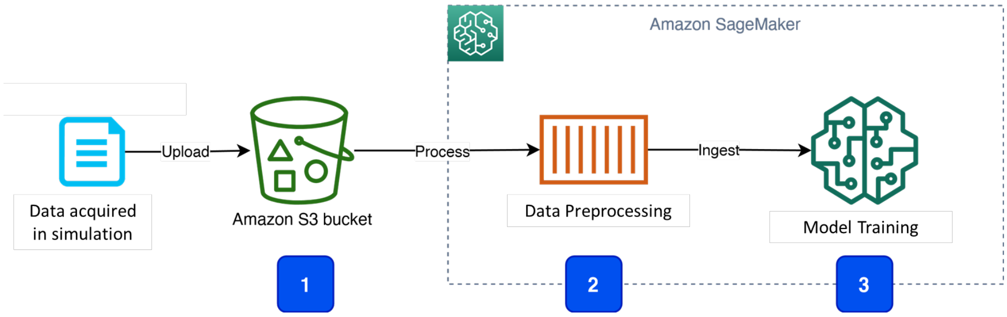

Figure 4.10: AWS Data Collection process

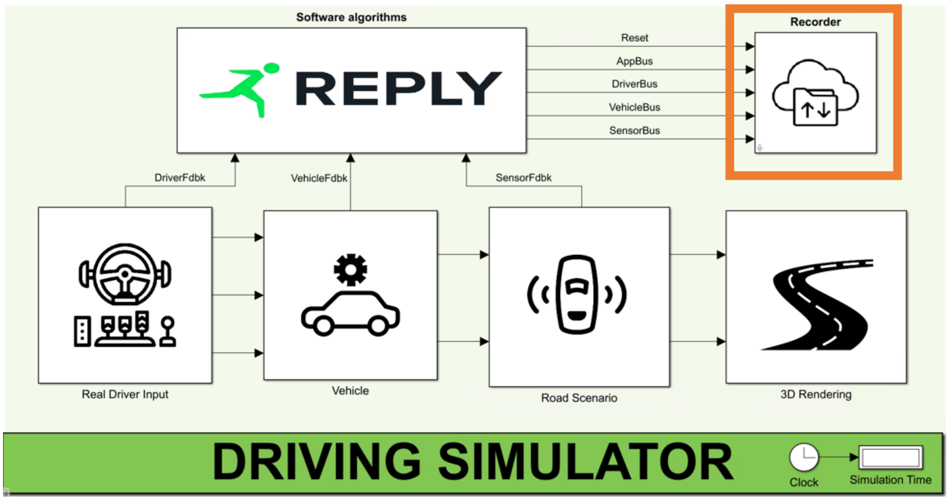

Figure 4.11: Recorder Block in General Architecture in Simulink

1. **Signal Acquisition:** Continuous monitoring of signals from different sources, including applications, vehicle-mounted sensors, driver inputs, and other vehicle-generated data. These signals represent a wide array of parameters like vehicle speed, engine conditions, driver behaviors, and app interactions.

2. **Buffering:** The collected signals are initially stored in a buffer within the Simulink environment. This buffering stage is crucial for managing the data flow, given the high frequency of data collection. The data is buffered at a frequency of 25 Hz, which implies that the data is captured and stored in the buffer 25 times per second.

3. **Cloud Transmission:** The buffered data is then transmitted to a cloud storage solution at a lower frequency of 1 Hz, meaning the data is sent to the cloud once every second. This ensures secure and accessible storage for further processing and analysis.

Dataset Characteristics:

- **High-Resolution Data:** The buffering frequency of 25 Hz results in a dataset with high-resolution, capturing detailed information of the monitored parameters. This high level of detail is essential for in-depth analysis applications such as predictive maintenance and real-time monitoring.

- **Cloud-based Storage:** Utilizing cloud storage provides benefits like scalability, reliability, and ease of access. The cloud-stored data is readily available for analysis, enabling applications such as remote diagnostics and advanced analytics.

- **Structured for Analysis:** The data is structured to facilitate efficient processing and analysis, allowing for the extraction of valuable insights that can inform decision-making processes, improve vehicle performance, and enhance user experiences. Data are stored according to the HASH code associated to the Driver ID and the journey that is generated every time a new acquisition is done and each time the driver during simulation go off the road.

This data collection and storage protocol is done through the AWS Cloud services reported in section 3.6

4.3.4 Distracted Driving Detection Algorithm

In this subsection, we introduce the development and implementation of a distracted driving detection algorithm, a critical advancement in driving safety technology. The process encompasses several key phases:

1. **Data Cleaning:** The initial step involves ensuring the data's integrity by removing inconsistencies, handling missing values, and filtering out irrelevant information. This refinement process is crucial for maintaining the quality and reliability of the dataset used for training the algorithm.

2. **Data Preprocessing:** Following data cleaning, the data undergoes preprocessing to transform it into a more suitable format for modeling. This includes normalization, feature extraction, and feature selection to enhance the learning capability of the algorithm and improve overall model efficiency.

3. **Training Phase:** Various machine learning models seen in section 3.4 are applied and trained on the preprocessed data. This stage involves parameter tuning, algorithm selection, and the application of cross-validation techniques to identify the model that best captures the nuances of distracted driving behavior.

4. **Model Evaluation:** After training, the models are evaluated to select the best performer based on metrics such as accuracy, precision, recall, and the F1 score. A set of models that exhibits superior performances in detecting distracted driving are chosen for inference.

5. **Inference in Simulation:** The selected model is then deployed in a simulation environment to assess its applicability in real-world driving scenarios. This crucial phase tests the algorithm's effectiveness in dynamic driving situations, ensuring its readiness for integration into safety systems and identifying any areas that may require optimization.

The entire pipeline from data preparation to simulation testing is essential for the distracted driving detection algorithm's reliability, accuracy, and effectiveness in enhancing driving safety.

Preprocessing

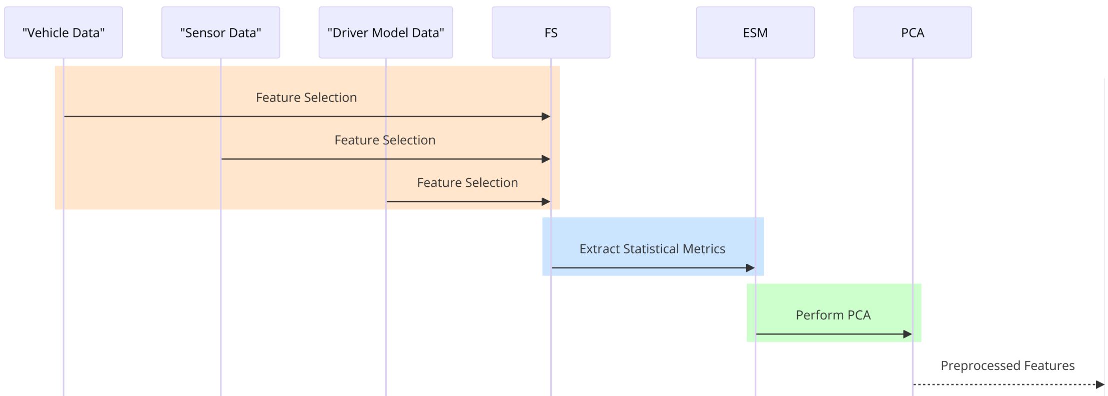

Figure 4.12: Signal preprocessing

In the context of distraction signal analysis, particularly within the data acquisition procedure for ground-truth signals, some preprocessing operation has been done. This limitation primarily arises from the method employed to collect distraction data, wherein participant involved in labelling phase cannot avoid generation of delay of distraction by pressing the mouse button used for acquisition of distraction groundtruth. This process introduces a notable latency in the recorded distraction ground-truth signals, attributable to human cognitive and motor response times.

To address this challenge and ensure the fidelity of the ground-truth data, a compensatory adjustment is implemented. This adjustment is necessitated by two key factors inherent in the human response to stimuli:

1. **Cognitive Processing Time:** The average human reaction time to visual stimuli is approximately 0.3 seconds. This interval encompasses the duration from the onset of the stimulus to the cognitive recognition of the distraction by the brain. [30]

2. **Motor Response Time:** Following cognitive recognition, an additional average duration of 0.2 seconds is required for the participant to convert this cognitive response into a physical action, such as pressing a button to indicate the perception of distraction.

Given the cumulative delay of 0.5 seconds introduced by these factors, the ground-truth signal for distraction is correspondingly shifted to accurately align with the actual moment the distraction was perceived, rather than the delayed instance of participant response. This temporal adjustment shown in Fig.4.13 is crucial for overcoming the limitations posed by the data acquisition procedure, thereby enhancing the reliability and accuracy of the distraction ground-truth data within the thesis research framework.

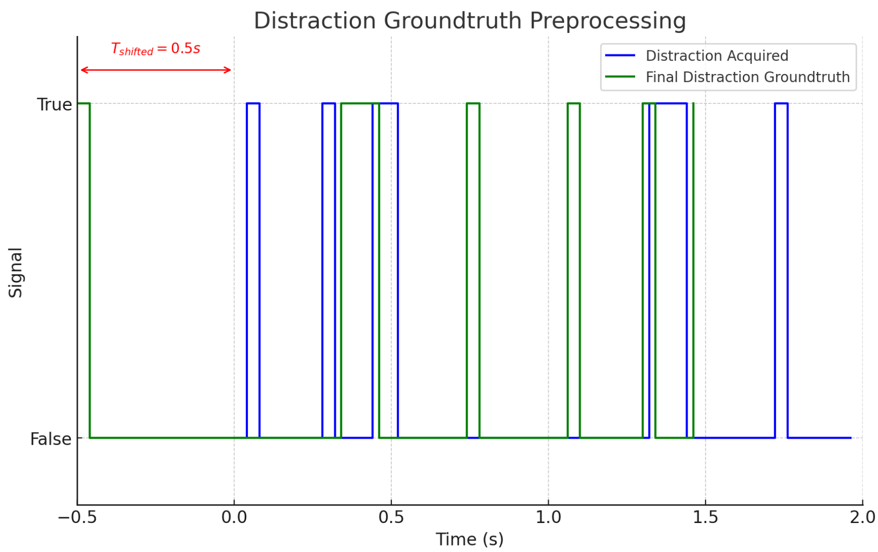

Figure 4.13: Shifting distraction ground-truth to compensate error in acquisition procedure

Feature selection

In the advancement of predictive modeling and machine learning, the critical task of selecting an optimal subset of features from a dataset has garnered significant attention. This step is instrumental in enhancing model performance, ensuring computational efficiency, and facilitating a deeper understanding of the underlying data patterns. To accomplish this objective, various techniques have been employed to process the selected signals. This procedure can be divided into two primary phases: the extraction of statistical metrics from chosen features, and the implementation of Principal Component Analysis (PCA), technique explained in section 3.5.5, to minimize the data volume required by the detection algorithm. This approach aims to maintain the same degree of phenomenon explainability with fewer signals, thereby reducing the computational effort required for the inference model that ideally tends working in real time during the simulation.

Table 4.5: Feature selected for distraction detection algorithm

Category	Variable	Size	Units
Control Inputs			
	Steering Input	[1x1]	[rad]
	Pedal Input	[1x1]	[-]
Vehicle Dynamics			
	Absolute Velocity	[1x1]	[m/s]
	Yaw Rate	[1x1]	[rad/s]
Environmental Variables			
	Left Lane Offset	[1x1]	[m]
	Right Lane Offset	[1x1]	[m]
	Heading Angle	[1x1]	[rad]
Cybernetic Driver Model			
	Sensitivity Lane Margin	[1x1]	[-]
	Sensitivity Road Curvature	[1x1]	[-]
	Steering Effort	[1x1]	[-]
	Braking Effort	[1x1]	[-]

Extraction of statistical metrics For each signal analyzed within the distraction detection algorithm (see Table 5.2, various metrics were extracted in both the time and frequency domains, utilizing the Fast Fourier Transform (FFT) for the frequency-based measurements. Metrics extracted in time domain are:

1. **Mean:** Represents the average value of the signal's features, offering a basic measure of the central tendency within the signal's data. The mean value is critical for understanding the signal's baseline level around which the features fluctuate.

2. **Variance:** Indicates the variability of the signal's features from the mean. A higher variance denotes a greater spread of the feature values, providing

insights into the signal's stability and consistency.

3. **Skewness:** Measures the asymmetry of the signal's feature distribution around the mean. Positive skew indicates a distribution with an elongated tail on the right side, while negative skew has the tail on the left. Skewness is useful for understanding the direction of the signal's deviation from the normal distribution.

4. **Kurtosis:** Quantifies the "tailedness" of the signal's feature distribution. A higher kurtosis implies more of the variance is due to infrequent extreme deviations, as opposed to frequent modestly sized deviations, indicating the presence of outliers or sharp peaks in the signal.

5. **Maximum Value:** The highest feature value in the signal, which is crucial for identifying the signal's peak amplitude and understanding the extremes within the signal's data.

6. **Minimum Value:** The lowest feature value in the signal, providing insights into the signal's deepest troughs and understanding the lower extremes within the signal's data.

In addition to the analysis of signals, the FFT is used as mathematical tool, particularly for decomposing a signal into its frequency components. For this type of application can result very efficient Working with FFT. In particular, various metrics can be employed to examine the spectral properties of a signal. Below are listed the metrics used as input to the detection algorithm:

1. **Amplitude Spectrum:** Provides the amplitudes of frequency components present in the signal. This metric is valuable for visualizing the intensity of various frequencies within the signal, offering insights into the signal's strength at different frequencies.

2. **Phase Spectrum:** Indicates the phase of each frequency component in the signal. This information is crucial for the original signal's reconstruction and for analyzing the temporal relationships among different frequency components, thus understanding how various parts of the signal are synchronized.

3. **Bandwidth:** Denotes the range of frequencies within which the signal contains most of its energy. This metric is significant for understanding the range of frequencies that characterize the signal, helping to determine the signal's overall spectral extent.

After evaluating statistical metrics for each feature in both, time and frequency domains of the considered time window, PCA is applied as explained in section

3.5.5 selecting a number of features corresponding to an explained variance ratio, defined in Equation 4.3 greater than 98%.

$$\text{Explained Variance Ratio} = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{n} \lambda_j} \tag{4.3}$$

Where:

- λ_i is the eigenvalue associated with the i^{th} principal component,
- k is the number of principal components considered,
- n is the total number of principal components.

Balancing Data: nominal and distracted

Balancing the data is not only crucial for preventing bias in machine learning models but also for reflecting the real-world scenario where nominal driving instances naturally occur more frequently than distracted ones. In practice, drivers are typically focused and attentive, with distractions being comparatively rare events. An unbalanced dataset would skew a model towards predicting nominal driving behavior, failing to alert on the critical yet infrequent distracted instances.

By employing balancing techniques, we can train our machine learning algorithms to recognize and classify distracted driving events with greater accuracy.

The Impact of an Imbalanced Data Imbalanced datasets are particularly problematic in distracted driving detection, where non-distracted driving instances significantly outnumber distracted instances. This imbalance can lead to detection models that are biased towards predicting the majority class, reducing their sensitivity to actual distractions.

Model Bias and Performance A model trained on imbalanced data might excel at recognizing safe driving but falter at detecting subtle or rare forms of distraction. This could lead to a high number of false negatives, where instances of distracted driving go undetected, compromising road safety.

Mitigation Techniques To combat imbalance, techniques such as oversampling distracted driving instances or generating synthetic examples of distracted driving can help balance the dataset. Employing specialized evaluation metrics like Precision-Recall Balanced Accuracy provide a more accurate measure of a model's ability to detect distracted driving, ensuring both high precision and recall.

Training Model & evaluation

The preprocessing of features sets the stage for the training phase, where shallow machine learning algorithms, as discussed in Section 3.4), are utilized. The training phase involves the application of a grid search to fine-tune the model parameters. This technique performs an exhaustive search over specified parameter values, cross-validating to find the combination that yields the best performance against our chosen metrics.

The performance of the model is evaluated by metrics such as precision, recall, and the F1-score, defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.5}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.6}$$

where TP denotes true positives, FP denotes false positives, and FN denotes false negatives.

Inference

The model with the highest precision, recall, and F1-score is chosen for deployment in the simulation environment as shown in Figure 4.14, enabling us to infer with high confidence during real-world applications. As detailed in Section 2.2.3, in accordance with the activation logic outlined in the ADDW (Advanced Driver Distraction Warning) regulations, the algorithm designed to predict driver distraction is configured to activate at speeds exceeding 20 km/h (see Fig. 4.15).

The prediction signal for driver distraction is integrated with additional alert signals emanating from various algorithms dedicated to the identification of improper driving behaviors, as well as other Advanced Driver-Assistance Systems (ADAS) implemented in the simulation.

This integration is governed by a well-defined logic that assigns priority levels based on the nature of the alert signal. Following this priority logic, visual and auditory alerts are activated on the simulated vehicle's dashboard (see Fig 4.17, sending this data to Unreal ensuring that the driver is adequately warned about potential distractions or hazardous driving behaviors, thereby enhancing overall driving safety.

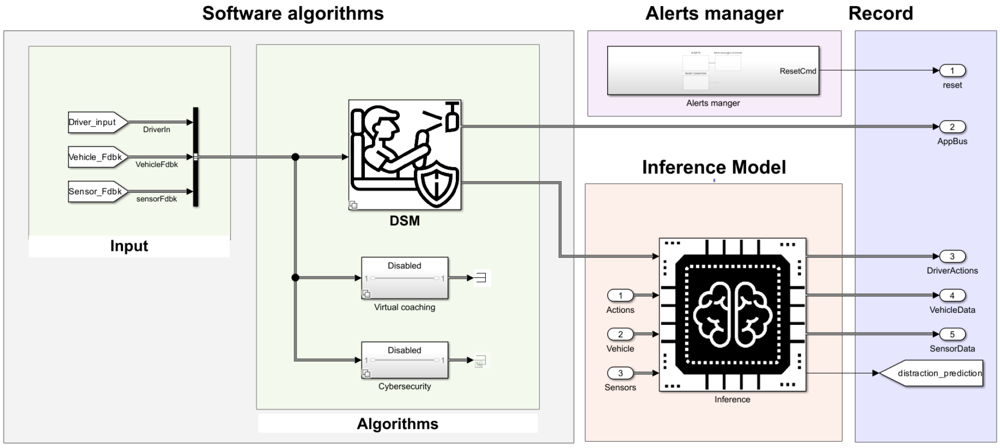

Figure 4.14: Inference Model in Simulink Architecture

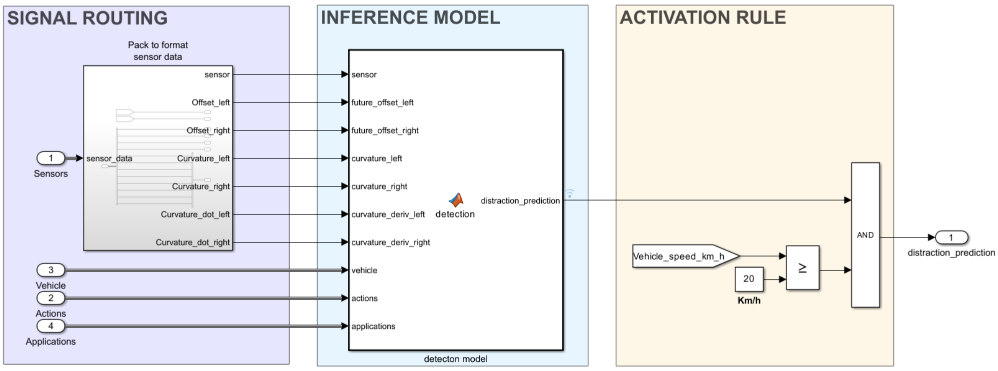

Figure 4.15: Inference Model Block

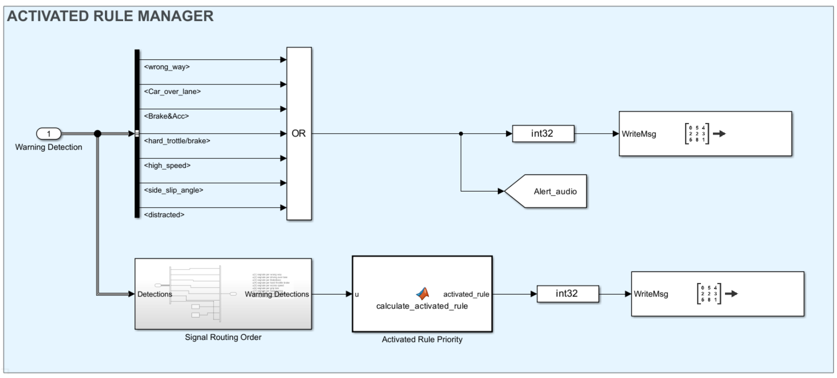

Figure 4.16: Alert manager

Figure 4.17: Distracted Alert on Dashboard

Chapter 5

Experimental results

5.1 Statistics on Experimental Data Acquisition

The experimental data acquisition process involved a group of participants, each contributing to the dataset with varying driving times as visualized in Figure 5.1. A total of ten drivers participated in the study, and the data collected ranged from short journeys to longer excursions, with a cumulative driving time amounting to 5.47 hours.

This considerable amount of data provides a basic foundation for the analysis of driving behaviors and distraction patterns. The variance in driving times, as demonstrated in the histogram, indicates the durations captured, enhancing the generalizability of the study findings.

Table 5.1: Average and Variance of Distraction Episode Durations per Driver

Driver	Average Duration (seconds)	Variance (seconds)
Driver 1	2.29	0.83
Driver 2	1.22	0.61
Driver 3	2.50	1.58
Driver 4	0.84	0.09
Driver 5	2.17	1.38
Driver 6	1.27	0.41
Driver 7	3.15	7.92
Driver 8	1.63	0.56
Driver 9	2.21	0.81
Driver 10	2.76	1.33

During the data collection phase, we evaluated the average time of distraction

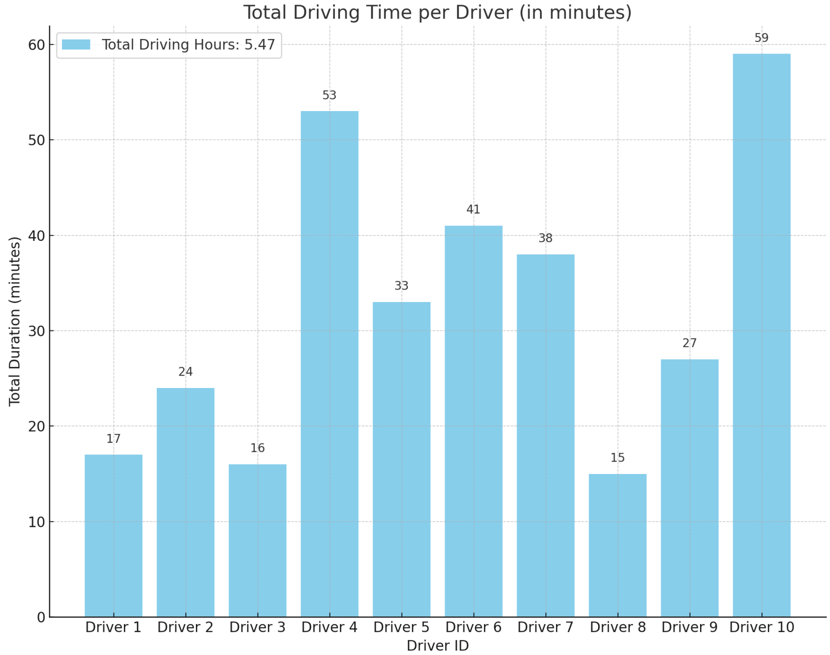

Figure 5.1: Statistics on Data Acquisition per Driver

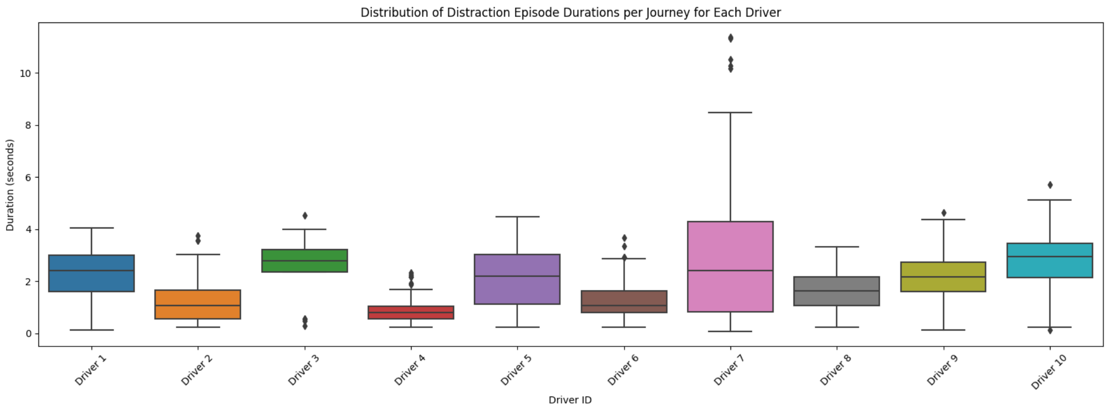

Figure 5.2: Distribution of Distraction Episode Durations per Driver

episodes for each driver, as depicted in Figure 5.2. The boxplot illustrates diverse statistics reflecting the individual variability in driving behavior and attention patterns, considered as studies suggest [31] that drivers often exhibit 'lizard glances,' a behavior characterized by rapid and frequent shifts in gaze that can indicate divided attention or cognitive load. It is important to consider that realism of the simulation about the surrounding environment while distracted and the challenging final section of the track have impact on driver focus diminishing distraction.

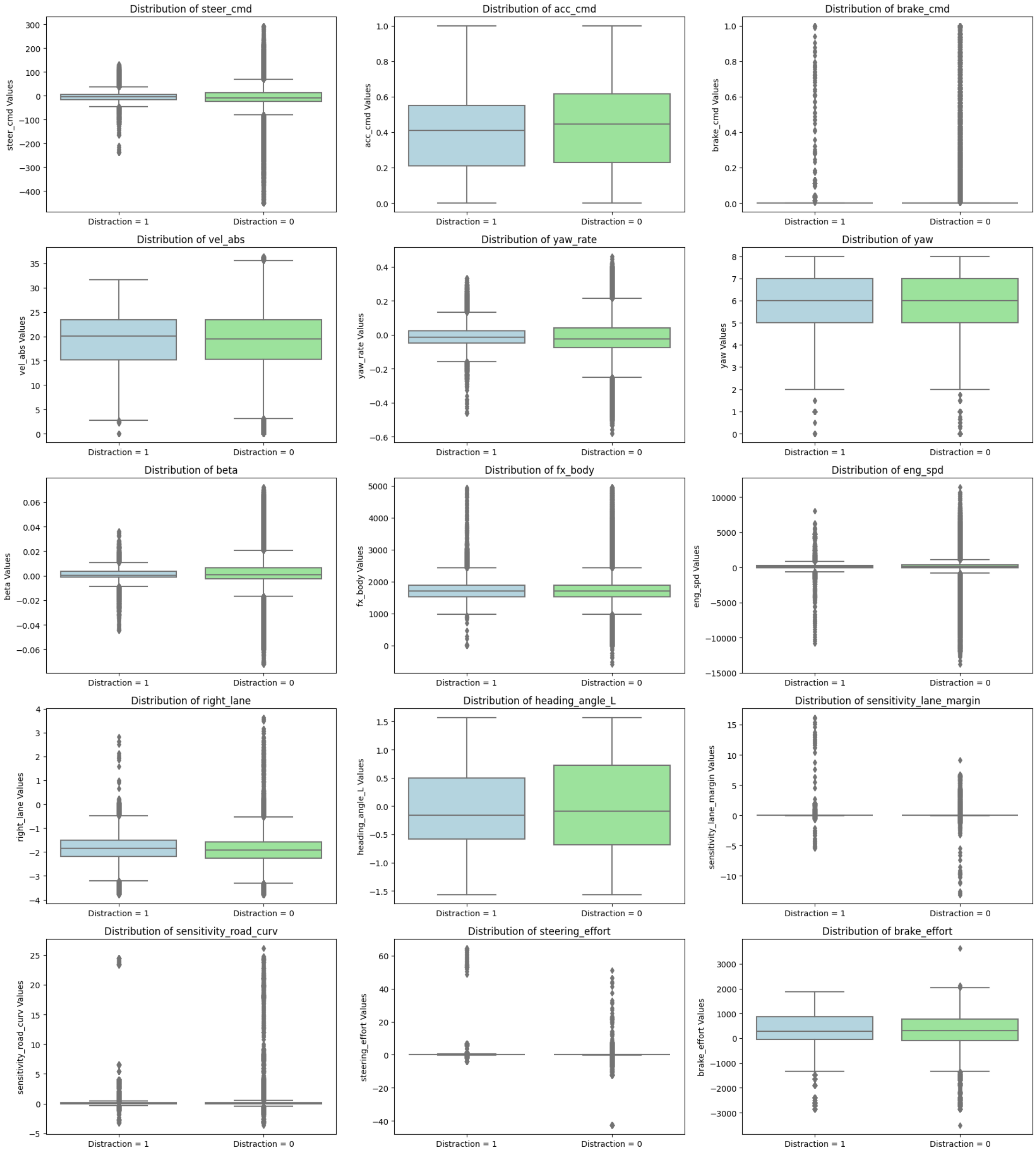

Figure 5.3: Boxplot: Nominal vs Distracted driving

The box-plots in Fig. 5.3 present a comparative analysis between distracted and nominal data acquisition, offering insights into the impact of driver distraction on vehicle control parameters. Notably, significant differences are observed in the steering angle, yaw rate, and slip angles, which are critical indicators of vehicle stability and orientation. The steering angle, for example, highlights the driver's input in response to distractions, while the yaw rate and slip angles give a sense of how vehicle trajectory and adherence to the road's curvature are affected during such episodes.

The heading angle is particularly telling, as it represents the direction of the car relative to the road. Deviations here suggest that distraction can significantly influence the driver's ability to maintain a consistent heading, leading to potential safety risks.

In contrast to these point-in-time measurements, the last four parameters related to the cybernetic model of driver behavior — including sensitivity to lane margin and road curvature, steering effort, and brake effort — require an analysis that extends beyond mere static values. For these parameters, their evolution over time is of greater interest. It's crucial to observe how these variables fluctuate during episodes of distraction, as they embody the driver's adaptive responses and overall control strategy. Understanding the dynamic changes in these parameters can provide deeper insights into the mechanisms of driver behavior under the influence of distraction.

Signal	**Unit**	**Description**
steer_cmd	Degrees	Steering command indicating the steering angle.
acc_cmd	-	Acceleration command to the vehicle's throttle.
brake_cmd	-	Brake command indicating the braking force applied.
vel_abs	m/s	Absolute velocity of the vehicle.
yaw_rate	rad/s	The rate of change of the vehicle's yaw angle.
yaw	Degrees	The vehicle's yaw angle, indicating its orientation.
beta	rad	Slip angle of the vehicle.
fx_body	N	Longitudinal force acting on the vehicle's body.
eng_spd	RPM	Engine speed in rotations per minute.
right_lane offset	m	Indicator of whether the vehicle is in the right lane.
heading_angle_L	Degrees	Heading angle relative to the left lane marker.
sensitivity_lane_margin	-	Sensitivity to the lane margin.
sensitivity_road_curv	-	Sensitivity to the road's curvature.
steering_effort	-	The effort on steering action.
brake_effort	-	The effort on braking action.

Table 5.2: List of signals with their respective units and brief descriptions.

5.2 Distraction detection algorithm results

The development of the distraction detection algorithm aimed to balance high recall rates for distraction instances against the need to minimize false negatives. The optimal model emerged as a compromise solution that effectively recognizes instances of driver distraction without becoming excessively intrusive during normal driving conditions.

This trade-off is particularly important due to the significantly higher proportion of time drivers spend in a nominal state compared to being distracted. A model with a high rate of false alerts could lead to driver irritation, reducing the system's usability and potentially compromising its adoption in real-world scenarios.

5.2.1 Impact of Model Parameters on Performance Metrics

In this section model parameters are analyzed according to their influence on the performance metrics. Parametrs considered are the following:

- **Window Size:** The temporal window of driving data that the algorithm analyze for potential distractions. Larger windows may encapsulate more comprehensive behavioral patterns, yet they risk omitting transient yet critical distractions.

- **Threshold for Majority Voting:** The decisive boundary that governs the classification of driving data as distracted or nominal. The threshold's magnitude is instrumental in calibrating the sensitivity and specificity of the detection.

- **Balance Ratio:** The proportional parameter used to balance dataset in the training dataset facing the problem discussed in Sec.4.3.4. This ratio is crucial for training the model to discern distractions effectively without being over-inclined towards the majority class.

These parameters serve as the fulcrum for training the distraction detection algorithm, each with significant influence over precision, recall, and F1 score, collective metrics that epitomize the model's accuracy and robustness. A granular analysis follows, presenting the effects these parameters have on the model's performance and the consequent trade-offs encountered during the optimization process.

As a result of different model trained of the distraction detection algorithm, the **F1 score** increases with an increase in window size and the threshold for majority voting. However, it decreases when the distraction windows are balanced with nominal driving windows.

Precision increases with the window size but is not significantly influenced by the threshold for majority voting used to classify a window as distracted or nominal. Higher precision is observed for balance ratio values equal to 5.

Recall improves with lower balance ratios as the model becomes better at recognizing distractions. It also increases with larger window sizes.

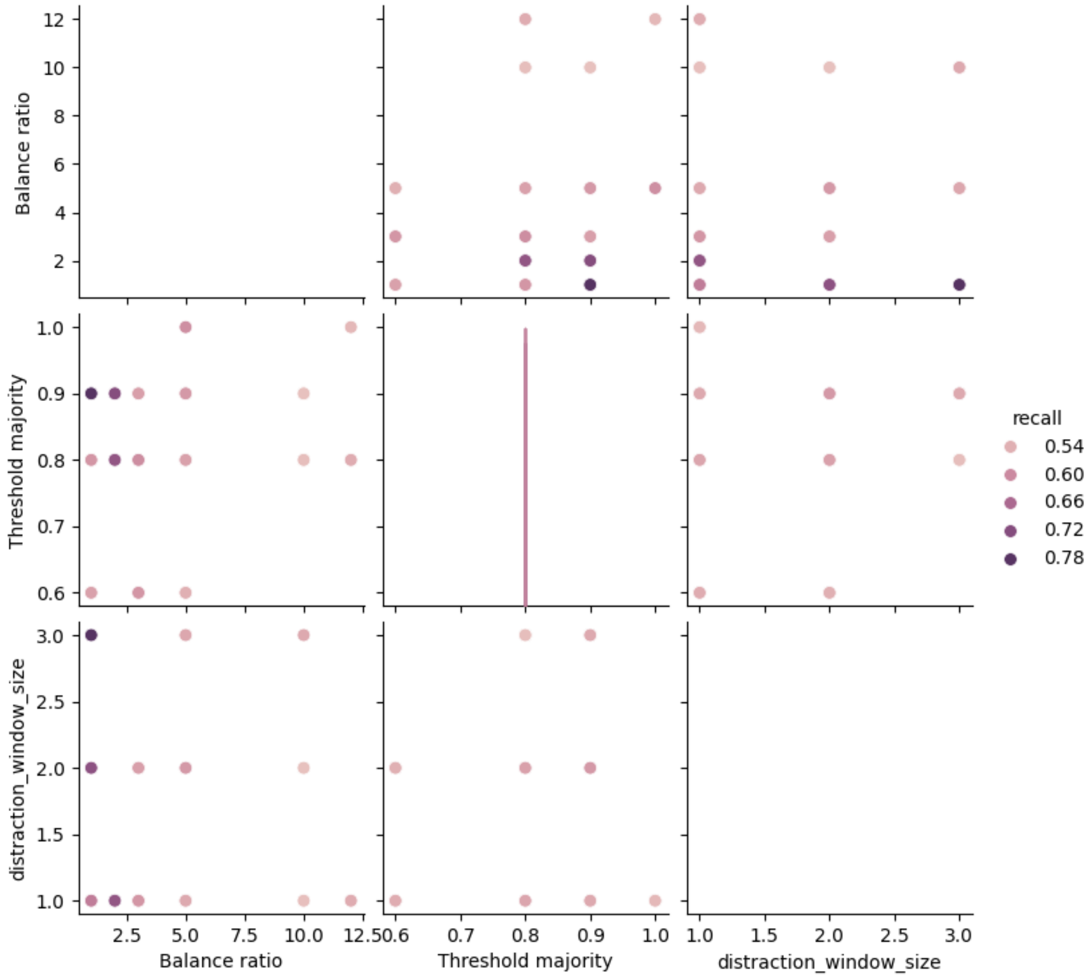

Figure 5.4: Recall performances according to parameter values

The challenge lies in selecting an optimal window size. While larger windows capture broader patterns of driver behavior, they risk missing shorter-duration distractions that could still pose a significant safety threat. According to regulatory standards, the system should recognize distractions within 6 seconds at speeds below 50 km/h and within 3 seconds above this threshold. Thus, while tuning the model parameters to optimize performance metrics, it is crucial to ensure the algorithm does not overlook brief yet potentially hazardous distractions.

5.2.2 Best Models: Model 1

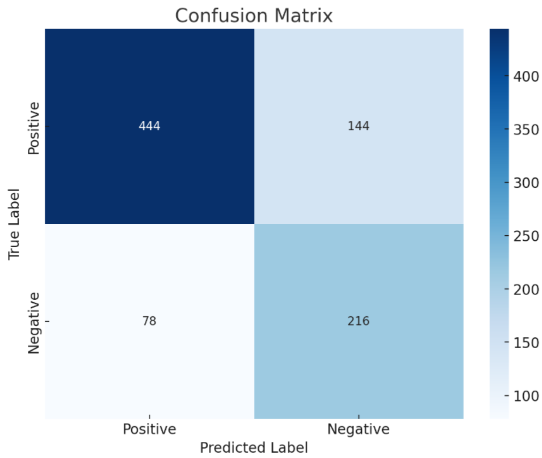

Figure 5.5: Model 1: Confusion Matrix

	precision	recall	f1-score	support
nominal	0.85	0.76	0.80	588
distracted	0.60	0.73	0.66	294
accuracy			0.75	882
macro avg	0.73	0.74	0.73	882
weighted avg	0.77	0.75	0.75	882

Figure 5.6: Model 1: Metrics

Parameters: Window size = 1s, window stride = 1s , balance ratio = 1 , majority voting treshold = 0.8

5.2.3 Best Models: Model 2

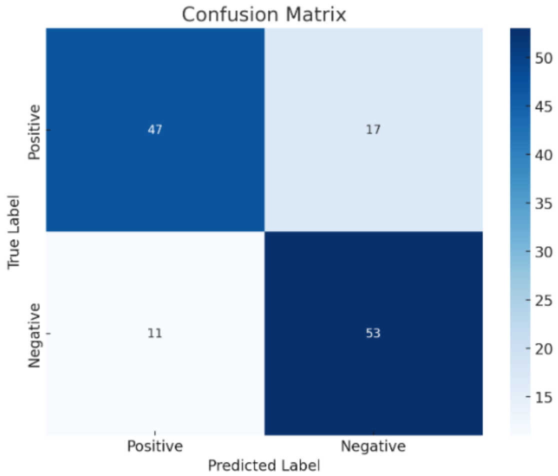

Figure 5.7: Model 2: Confusion Matrix

	precision	recall	f1-score	support
nominal	0.85	0.76	0.80	588
distracted	0.60	0.73	0.66	294
accuracy			0.75	882
macro avg	0.73	0.74	0.73	882
weighted avg	0.77	0.75	0.75	882

Figure 5.8: Model 2: Metrics

Parameters: Window size = 3s, window stride = 1s , balance ratio = 1 , majority voting treshold = 0.9

5.2.4 Best Models: Model 3

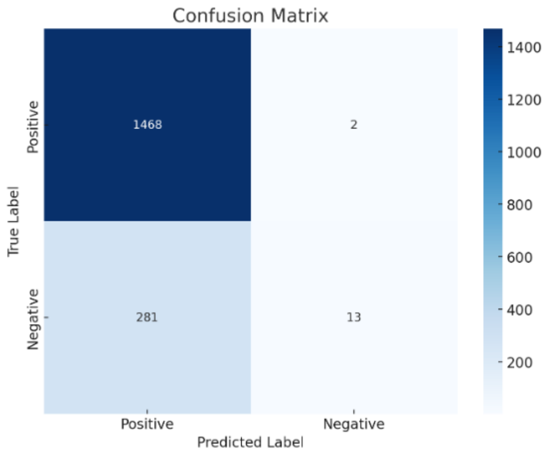

Figure 5.9: Model 3: Confusion Matrix

	precision	recall	f1-score	support
nominal	0.84	1.00	0.91	1470
distracted	0.87	0.24	0.08	294
accuracy			0.84	1764
macro avg	0.85	0.62	0.50	1764
weighted avg	0.84	0.84	0.77	1764

Figure 5.10: Model 3: Metrics

This matrix reveals a high number of true positives and a small number of true negatives, with no false positives and a moderate number of false negatives. This ca be caused by small amount of data as in model 2 due to small amount of distraction window after selecting a larger value for the window size.

Parameters: Window size = 2s, window stride = 1s , balance ratio = 5 , majority voting treshold = 0.9

Among the three models considered for inference, model 3 performed the best. Despite the first model appearing superior due to potentially higher aggregate metrics, it has a too high number of false positives. This is a significant issue because false positves represent instances where the model fails to correctly identify nominal driving raising the alert too many times without presence of distraction.

Model 3, although it may have slightly lower overall metrics compared to the first model, shows a better balance between false positives and false positives, making it less "annoying" or problematic in practical use. The reduced annoyance of model 3 could refer to its higher reliability in not overlooking positive cases, which is often more critical than avoiding false alarms (false positives) depending on the application.

The chosen model demonstrates an acceptable level of recall for distracted driving, ensuring that most distractions are detected, while maintaining a low enough false negative rate to avoid frequent false alerts.

Chapter 6

Discussion

6.1 Sensor-Based and Sensorless Algorithm

Driver distraction is a multifaceted issue, deeply intertwined with the unpredictable nature of human behavior. It encompasses a range of activities that divert the driver's attention from the primary task of driving, potentially leading to hazardous situations. The imminent regulatory mandate underscores the need for effective technological solutions to address this challenge. In this context, the debate often revolves around two primary approaches: sensor-based and sensorless algorithms.

Sensor-based systems, such as those employing cameras to monitor eye movements, offer direct, real-time insights into the driver's focus and alertness. These systems are capable of detecting instances where the driver's gaze deviates from the road for prolonged periods, signaling potential distractions. The advantages of sensor-based solutions include their precision and responsiveness in capturing specific indicators of distraction, such as eye movement patterns, blink rates, and head positioning.

However, sensor-based systems are not without their drawbacks. Issues such as privacy concerns, the need for continuous line-of-sight, and the potential for false positives in varying lighting conditions or with different eyewear can pose challenges. Moreover, the reliance on physical sensors may increase the complexity and cost of the vehicle systems.

On the other hand, sensorless algorithms offer an alternative approach by analyzing indirect indicators of driver behavior, such as steering patterns, braking behavior, and vehicle speed variations. These systems infer the driver's state by monitoring the vehicle's operational parameters and driving dynamics, which can provide a broader context of the driving situation. Sensorless solutions can be less intrusive and may circumvent some of the privacy and technical challenges associated with sensor-based systems.

Nevertheless, sensorless algorithms might lack the immediacy and specificity in detecting distraction episodes compared to their sensor-based counterparts. The indirect nature of the data they analyze could result in a delayed or less accurate assessment of the driver's attention level.

6.2 The Importance of Data Preprocessing

Data preprocessing is an essential step in developing models for distracted driving detection. This process ensures the data collected from various sensors, such as cameras and accelerometers, is clean, consistent, and ready for analysis. The quality of preprocessing directly influences the model's ability to accurately identify instances of distracted driving.

Data Cleaning

In the context of distracted driving, data cleansing involves filtering out noise from sensor data, correcting errors, and handling missing values. For instance, camera footage may require correction for varying light conditions, while accelerometer data might need filtering to remove non-driving related movements.

Normalization and Standardization

Given the diverse range of sensors and the different scales of data they produce, normalization and standardization are crucial. This ensures that each input feature contributes equally to the analysis, preventing any one sensor from disproportionately influencing the detection model.

Feature Extraction and Selection

Feature extraction in distracted driving detection involves identifying key indicators of distraction from raw sensor data. This could include analyzing patterns in steering wheel movement, facial expressions, or eye gaze direction. Effective feature selection helps isolate the most predictive indicators of distracted driving, improving model accuracy and efficiency.

6.3 Distraction groundtruth

The acquisition procedure of ground truth data is crucial for developing distracted driving detection algorithms, as it benchmarks their accuracy. This data can be gathered through in-lab simulations, which offer controlled conditions but may not fully represents the complexities of real-world driving. It can be important to introduce variability in distraction behavior recorded that can challenge data analysis.

6.4 Sim vs. Real-World Driving Perspectives

One of the pivotal aspects of evaluating driver behavior and distraction mechanisms involves understanding the disparities between simulated environments and real-world driving conditions. These differences have profound implications not only for the manifestation of distraction episodes but also for the overall comfort of the driver and the utilization of external reference points during driving.

Distraction Episodes

In simulated environments, distraction episodes can be precisely controlled and replicated, which is essential for systematic data collection and analysis. However, real-world distractions are often more dynamic and unpredictable. Factors such as sudden noises, movements outside the vehicle, or unanticipated actions by other road users can lead to distraction episodes that are difficult to replicate in a simulated setting. Consequently, while simulations are invaluable for studying specific distraction triggers and responses, they may not fully encapsulate the complexity and spontaneity of real-world distractions.

Driver Comfort

The comfort level of a driver in a simulated environment can significantly differ from that in a real vehicle. Simulators, depending on their design and realism, might not accurately replicate the tactile feedback, seating ergonomics, and overall spatial awareness experienced in a real vehicle. These discrepancies can influence a driver's stress levels, fatigue, and consequently, their susceptibility to distraction. The physical and psychological comfort of the driver plays a crucial role in how distractions are perceived and managed, potentially skewing the data acquired from simulated environments.

External Reference Points

Real-world driving relies heavily on external reference points, such as landmarks, road signs, and the behavior of other vehicles, to navigate and make driving decisions. These reference points provide contextual cues that are integral to anticipatory driving skills and situational awareness. In a simulation, despite advances in visual and environmental realism, certain subtleties and depth cues from the real environment might be absent or less pronounced. The lack of these real-world reference points can alter the driver's interaction with the environment in the simulation, potentially affecting the naturalness of their driving behavior and their response to distractions.

Implications for Research and Training

Understanding the differences between simulated and real-world driving conditions is crucial for interpreting data related to driver distraction and behavior. It underscores the importance of complementing simulated studies with real-world observations and experiments where feasible. Moreover, these insights are invaluable for refining simulation technologies and methodologies, aiming to bridge the gap between simulated environments and real-world driving experiences for more accurate research outcomes and effective driver training programs.

Chapter 7

Conclusions

7.1 Forthcoming Regulations

With the advent of the General Vehicle Safety Regulation, a significant stride has been made towards augmenting road safety within the European Union. Starting from mid-2024 [32], it becomes mandatory for all new vehicle types to be equipped with an advanced driver distraction warning system. This regulatory requirement, as detailed in an EU initiative, aims at enhancing the detection and prevention of driver distractions through real-time monitoring of driver eye movements, thereby contributing to a reduction in road accidents .

7.2 Future Perspectives

Given the complexity of human behavior and the diverse scenarios encountered on the road, neither approach is unfailing. Therefore, a fusion becomes essential. Integrating sensor-based and sensorless technologies can leverage the strengths of both to create a more comprehensive and effective driver distraction warning system. Such a hybrid approach can combine the direct, real-time monitoring capabilities of sensor-based systems with the broader contextual analysis offered by sensorless algorithms. This integrated solution can enhance the accuracy and reliability of distraction detection, thereby maximizing the potential for preventing accidents and ensuring safety on the roads.

In conclusion, as we navigate the intricate landscape of automotive safety regulations and technological advancements, the synergy between sensor-based and sensorless algorithms presents a promising pathway. By harnessing the complementary benefits of both approaches, we can develop sophisticated systems capable of safeguarding drivers against the perils of distraction, thereby contributing to a safer and more secure driving environment.

7.2.1 Driver Models Improvements

Force feedback

Incorporating force feedback that reflects vehicle dynamics and interaction with the road surface, rather than just steering angle, would significantly enhance simulation realism and it will help relaxing the assumption done on inputs and output measured in the estimation with the kalman filter. This upgrade would improve the sensitivity and accuracy of cybernetic model parameters, which are crucial for replicating driver reactions based on vehicle responses. This nuanced approach to force feedback would provide a more authentic driving experience, essential for accurately modeling and understanding driver behavior in various scenarios.

7.2.2 Cloud Retraining with AWS

A potential avenue for enhancing the driver distraction detection algorithm, is the implementation of a cloud retraining mechanism, specifically utilizing Amazon Web Services (AWS) as the cloud platform. This approach would involve periodically retraining the machine learning (ML) algorithms that underpin our distraction detection capabilities, using the vast computational resources and scalable infrastructure provided by AWS.

Process Overview

The retraining process would be initiated by collecting a diverse set of driving data from vehicles over time, capturing various driver behaviors, responses to stimuli, and potential distraction indicators. This data, once accumulated to a certain threshold, would be securely transmitted to AWS, where it would undergo analysis and be used to update the ML models.

Personalization and Adaptation

The key advantage of cloud retraining lies in its ability to personalize the distraction detection system to individual drivers. By continuously incorporating new driving data, the ML algorithms can adapt to each driver's unique driving style, enhancing the system's accuracy and reducing false positives. This personalized approach acknowledges the subjective nature of driver distractions and tailors the system to better meet individual needs.

Deployment of Updates

Once retraining is complete, the updated ML models would be deployed back to the vehicles as a system update. This deployment process would ensure that each vehicle's distraction detection system remains state-of-the-art, benefiting from the latest data and insights. The integration with AWS facilitates a seamless and efficient update process, leveraging its robust cloud infrastructure.

Conclusion

Incorporating cloud retraining into our system represents a forward-looking strategy to maintain the efficacy and relevance of our driver distraction detection technology. By leveraging AWS for cloud retraining, we can ensure that our system evolves with our drivers, providing personalized, adaptive, and increasingly accurate detection capabilities that promise to enhance road safety significantly.

List of Tables

List of Figures

Appendix A

Cybernetic Driver Model

State Transition Functions

The state transition functions used in the system's model are provided below. These functions are implemented in MATLAB and used to compute the state transitions for the cybernetic driver model.

Listing A.1: State Transition Function

```
function x = state_transition_fcn(x,u)
% Euler integration of continuous-time dynamics x'=f(x) with sample
    time dt

dt = 0.04; % [s] Sample time
x = x + state_transition_fcn_cont(x,u)*dt;
end

function xk1 = state_transition_fcn_cont(x,u)
% value of the fixed parameters
tau_p = 0.4;
T_i   = 0.5;
K_r   = -0.35;
T_n   = 0.04;

if u(3) < 10e-04 % to avoid numerical errors
     xk1 = [-1/T_i * (x(1)-x(5)*u(4)/(u(3)+10e-04));
             1/tau_p * (x(1)-x(2)+x(4)*u(5));
             1/T_n * ((K_r*u(3)+x(6))*x(2)-x(6)*u(1));
             0;
             0;
             0];
else
    xk1 = [-1/T_i * (x(1)-x(5)*u(4)/u(3));
            1/tau_p * (x(1)-x(2)+x(4)*u(5));
```

```
            1/T_n * ((K_r*u(3)+x(6))*x(2)-x(6)*u(1));
            0;
            0;
            0];
end
end
```

Listing A.2: EKF Initialization Script

```
%% EKF support script

% initialization and covariance matrix definition
kalman = [];
kalman.x0 = [0 0 0 2 4 6]';
kalman.P0 = diag([30, 30, 30, 1, 1, 1]);
kalman.Q = 10^5 * diag([10^-5, 10^-5, 10^-5, 10^-5, 10^-5, 10^-5]);
kalman.R = 10^-3;

assignin('base', 'kalman', kalman);
```

Bibliography

[1] S. Santia. «Driving Style Estimation for Impaired Driving Detection: An AI-based solution using Reinforcement Learning». MA thesis. Politecnico di Torino, 2022 (cit. on p. 1).

[2] N. Chiapello. «Driving style estimation for driver state monitoring». MA thesis. Politecnico di Torino, 2021 (cit. on pp. 1, 13).

[3] E. Yacometti. «Driving Style Estimation for driver state monitoring». MA thesis. Politecnico di Torino, 2023 (cit. on p. 1).

[4] *Volvo Cars introduces Driver Alert Control - world-first technology to alert tired and distracted drivers.)* Volvo Car Corporation. 2007 (cit. on p. 3).

[5] *Ford and MIT Project Uses LiDAR Sensors and Smart Devices to Measure Pedestrian Traffic and Predict Demand for New, On-Demand Electric Shuttles.)* Ford Media Center. 2016 (cit. on p. 4).

[6] *Driver Attention Monitoring System)*. Toyota Motor Corporation. 2016 (cit. on p. 4).

[7] Yanchao Dong, Zhencheng Hu, Keiichi Uchimura, and Nobuki Murayama. «Driver Inattention Monitoring System for Intelligent Vehicles: A Review». In: *IEEE Transactions on Intelligent Transportation Systems* 12 (2009), pp. 596–614. URL: `https://api.semanticscholar.org/CorpusID:9323978` (cit. on p. 4).

[8] «National Highway Traffic Safety Administration (NHTSA) Distracted Driving». In: (2021) (cit. on p. 5).

[9] Llerena LE et al. «An evidence-based review: distracted driver». In: (2015) (cit. on p. 5).

[10] European Road Safety Observatory. *Distracted Driving Statistics 2023*. Dec. 2023. URL: `https://road-safety.transport.ec.europa.eu/document/download/3aa20fd9-b969-416a-9e43-094d445f6254_en?filename=ERSO-TR-Distraction_2023-12-19.pdf` (cit. on p. 5).

[11] National Highway Traffic Safety Administration. *Countermeasures That Work: Distracted Driving*. 2021 (cit. on p. 5).

[12] SAE Driver Metrics, Performance, Behaviors and States Committee. *Operational definitions of driving performance measures and statistics*. SAE International. J2944. 2015 (cit. on p. 8).

[13] European Parliament and Council. *Regulation (EU) 2019/2144 of the European Parliament and of the Council of 27 November 2019 on type-approval requirements for motor vehicles and their trailers, and systems, components and separate technical units intended for such vehicles, as regards their general safety and the protection of vehicle occupants and vulnerable road users*. 2019 (cit. on pp. 11–13).

[14] Commissione Europea. *Regolamento Delegato (UE) 2023/2590 della Commissione del 13 luglio 2023 che integra il regolamento (UE) 2019/2144 del Parlamento europeo e del Consiglio stabilendo norme dettagliate concernenti le procedure di prova e i requisiti tecnici specifici per l'omologazione di determinati veicoli a motore per quanto riguarda i sistemi di avviso avanzato della distrazione del conducente e che modifica tale regolamento*. 2023 (cit. on pp. 11, 13, 14).

[15] General Motors. *GM Driver Assistance Systems*. `https://www.gmparts.com/content/dam/gmparts/na/us/en/index/technical-resources/technical-documents/gm-driver-assistance-systems-aid.pdf.html`. 2021 (cit. on p. 16).

[16] Nissan Motor Corporation. *Intelligent Driver Alertness*. `https://www.nissan-global.com/EN/INNOVATION/TECHNOLOGY/ARCHIVE/IDA/`. 2021 (cit. on p. 16).

[17] SmartEye. *Smart Eye's market-leading driver monitoring software included in new Volvo EX90*. `https://www.smarteye.se/news/smart-eyes-market-leading-driver-monitoring-software-included-in-new-volvo-ex90/`. 2022 (cit. on p. 16).

[18] Franck Mars and Philippe Chevrel. «Modelling human control of steering for the design of advanced driver assistance systems». In: *Annual Reviews in Control* 44 (Oct. 2017). DOI: `10.1016/j.arcontrol.2017.09.011` (cit. on p. 20).

[19] Will Hoult and David J. Cole. «A neuromuscular model featuring co-activation for use in driver simulation». In: *Vehicle System Dynamics* 46.sup1 (2008), pp. 175–189. DOI: `10.1080/00423110801935798` (cit. on p. 21).

[20] Dario D Salvucci and Robert Gray. «A two-point visual control model of steering». In: *Perception* 33.10 (Oct. 2004), pp. 1233–1248 (cit. on p. 23).

[21] *The driving simulator: A journey through time*. `https://group.mercedes-benz.com/company/magazine/technology-innovation/100-things-driving-simulator.html`. 2020 (cit. on p. 27).

[22] *Vehicle Dynamics Blockset - MATLAB*. https://it.mathworks.com/products/vehicle-dynamics.html. MathWorks (cit. on p. 29).

[23] D. Setiawan, M. Safarudin, and A. Singh. «Modeling, simulation and validation of 14 dof full vehicle model». In: *International Conference on Instrumentation, Communication, Information Technology, and Biomedical Engineering 2009*. 2009, pp. 1–6 (cit. on p. 29).

[24] J. Setiawan, M. Safarudin, and A. Singh. *Modeling, simulation and validation of 14 dof full vehicle model*. Dec. 2009 (cit. on p. 29).

[25] MathWorks. *Python Compatibility with MATLAB*. https://www.mathworks.com/support/requirements/python-compatibility.html (cit. on p. 48).

[26] MathWorks. *Curved Road*. https://www.mathworks.com/help/driving/ref/curvedroad.html (cit. on p. 49).

[27] A. Ameyoe, P. Chevrel, E. Le-Carpentier, F. Mars, and H. Illy. «Identification of a linear parameter varying driver model for the detection of distraction». In: *IFAC-PapersOnLine* 48.26 (2015), pp. 37–42 (cit. on pp. 53, 56).

[28] A. Ameyoe. «Estimation de la distraction fondée sur un modèle dynamique de conducteur: principes et algorithmes». In: (2016) (cit. on pp. 53, 56, 57).

[29] MathWorks. *Magic Formula Tire Force and Torque*. https://it.mathworks.com/help/sm/ref/magicformulatireforceandtorque.html. 2021 (cit. on p. 54).

[30] T Madl, Bernard J Baars, and S Franklin. «The timing of the cognitive cycle». In: *PLoS One* 6.4 (2011). Published 2011 Apr 25, e14803. DOI: 10.1371/journal.pone.0014803 (cit. on p. 69).

[31] Shiyan Yang, Brook Shiferaw, Trey Roady, Jonny Kuo, and Michael Lenné. «Drivers Glance Like Lizards during Cell Phone Distraction in Assisted Driving». In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 65 (Sept. 2021), pp. 1410–1414. DOI: 10.1177/1071181321651147 (cit. on p. 77).

[32] Nick Bowyer. «EU Regulation on Advanced Driver Distraction Warning Systems Published». In: (Dec. 2023). URL: https://www.interregs.com/articles/spotlight/eu-regulation-on-advanced-driver-distraction-warning-systems-published-000260 (cit. on p. 90).