Autonomous emergency braking for highway trajectory planning

Ricardo André Pereira Ribeiro

Relatório do Projeto de Dissertação

Engenharia Informática

(2º ciclo de estudos)

Orientador: Prof. Doutor Nuno Gonçalo Coelho Costa Pombo

Abril de 2024

Declaração de Integridade

Eu, Ricardo André Pereira Ribeiro, que abaixo assino, estudante com o número de inscrição M11676 de/o Mestrado em Engenharia Informática da Faculdade de Engenharias, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o Código de Integridades da Universidade da Beira Interior.

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, que em particular atendi à exigida referenciarão de frases, extratos, imagens e outras formas de trabalho intelectual, e assumindo assim na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 11/04/2024

Assinado por: **RICARDO ANDRÉ PEREIRA RIBEIRO**
Num. de Identificação: BI15927615
Data: 2024.04.15 14.25.20 GMT Daylight time

CHAVE MÓVEL

Acknowledgements

In realizing this master's thesis, I would like to express my sincere acknowledgments to all those who contributed to making it happen.

I would especially like to thank my supervisor, Prof. Doutor Nuno Gonçalo Coelho Costa Pombo, for his disponibility, support, guidance, and professional judgment, which were of extreme importance to paving the way for all the work that this paper accomplished, but also thank you for all knowledge and education shared in all of the steps taken in the realization of this paper. I also want to appreciate the infrastructure, materials, and technologies offered by the professor and the institution to which he belongs. A special thanks to Universidade da Beira Interior, all the department professors, and all laboratory coworkers for allowing me to develop my technical skills and knowledge personally and professionally.

In addition, I would like to thank my parents and brothers for their unconditional support, encouragement, and wisdom, as well as for all the hard work they have done to help me reach this level.

Finally, thank my friends for all the discussions, the few good ideas about this project, and the days of distraction that allowed me to recharge my energies.

Resumo

Veículos autônomos (AV) exigem habilidades críticas em diversos cenários: consciencialização, inteligência na tomada de decisão e controle executivo. Sendo o aprimoramento destas características uma reação natural ao aparecimento de sistemas de reconhecimento como sensores, que apresentam medições cada vez mais precisas e com uma maior variedade de tipos de dados recolhidos, que combinado com o surgimento de novas tecnologias e abordagens matemáticas para resolver problemas existentes neste setor, bem como a evolução no setor da área de inteligência artificial permitiu uma rápida automatização da indústria dos transporte. Todo este crescimento, em diversas áreas levou a um rápido desenvolvimento deste setor nos últimos anos.

A ideia principal dos AVs é criar um módulo inteligente de tomada de decisão capaz de controlar todos os processos essenciais associados a um veículo, desde a criação de trajetórias ou controlo de direção até mesmo a análise de risco, como é o caso do sistema de Travagem Automática de Emergência (AEB). Este último, como ADAS, é realizado principalmente para mitigar erros humanos como distrações do condutor, análise de riscos com precisão matemática, combater deficiências na perceção humana em cenários com condições ambientais ou físicas adversas (fadiga, stresse, ansiedade) e prevenir acidentes automobilísticos. Isto reduzirá o tráfego rodoviário, minimizará as vítimas e os ferimentos humanos, e poupará milhões em perdas monetárias para todos os utilizadores das estradas. Os VAs são o futuro do fortalecimento e melhoramento das políticas de segurança em diversos cenários. No entanto, o seu elevado preço no processo de desenvolvimento e teste provou ser um impedimento significativo ao desenvolvimento destas tecnologias. Consequentemente, serão abordadas as soluções disponíveis na área da travagens automáticas de emergência e apresentadas todas as novas soluções e estudos, os seus defeitos e vantagens, o estado do processo de desenvolvimento, sistemas de teste, a qualidade e a fiabilidade dessas tecnologias.

Este trabalho visa criar AVs com planeamento de trajetória bem projetado utilizando um Modelo de Controlo Preditivo (MPC) adaptativo capaz de alcançar excelente desempenho em cenários críticos de autoestradas. Além disso, pretende também agregar um sistema de travagem de emergência que usa análises multidimensionais, incluindo deteção de colisão, tempo até colisão (TTC) e distância de travagem. Este estudo também mostra a necessidade de dar ênfase especial à verificação, validação e testes (VVT) na indústria automóvel, o que tem contribuído significativamente para o desenvolvimento de sistemas de automatização. Eles permitem que os desenvolvedores testem 'software' com baixo custo e baixo risco, permitindo encontrar falhas ocultas em fases preliminares e aumenta a confiança na segurança, funcionalidade e análise de transações para protótipos autónomos em redes rodoviárias existentes. Isto está a tornar-se cada vez mais uma norma na indústria automóvel graças à relação custo-benefício, permitindo a remoção de erros antes de chegar à fase de testes em ambientes reais, onde o custo dos erros, tanto monetário como humanitário, pode ser catastrófico. Para que os modelos fossem realizados via ambientes de simulação, este trabalho utilizou o Matlab Simulink.

Palavras-chave

Veículos autônomos, segurança crítica, verificação, validação, teste, simulação

Abstract

Autonomous vehicles (AV) require critical skills in several scenarios: awareness, intelligent decision-making, and executive control. The improvement of these characteristics is a natural reaction to the emergence of recognition systems such as sensors, which present increasingly precise measurements and a more significant collection of types of data, combined with the emergence of new technologies and mathematical approaches to existing problems in this sector, as well as the evolution in the artificial intelligence sector as raping the transport industry to a new level of automatization. In several areas, all this growth has led to this sector's rapid development in recent years.

The main idea of AVs is to create an intelligent decision-making module capable of controlling all essential processes associated with a vehicle, ranging from creating trajectories or steering control to even risk analysis, as is the case of the Autonomous Emergency Braking (AEB) system. The latter, as ADAS, is carried out mainly to mitigate human errors such as driver distractions, risk analysis with mathematical precision, combat deficiencies in human perception in scenarios with adverse environmental or physical conditions (fatigue, stress, anxiety), and prevent car accidents. This will reduce road traffic, minimize human casualties and injuries, and save millions in monetary loss for all road users. AVs are the future of strengthening and improving security policies in various scenarios. However, their high price in the development and testing process has proven to be a significant deterrent to developing these technologies. Consequently, available solutions in the area of autonomous emergency braking will be addressed, and all new solutions and studies, their strengths and weaknesses, the state of the development process, test systems, quality, and reliability, will be presented.

This work aims to create AVs with well-designed trajectory planning using an adaptive Model Predictive Control (MPC) capable of achieving outstanding performance in critical highway scenarios. Furthermore, it also aims to aggregate an emergency braking system that reacts to multidimensional analyses, including collision detection, time to collision (TTC), and braking distance. This study also shows the necessity of placing particular emphasis on verification, validation, and testing (VVT) in the automobile industry, which has contributed significantly to the development of automation systems. They allow developers to test software at a low cost and risk cycle, finding hidden faults in the preliminary phase and increasing confidence in security, functional, and transaction analysis for autonomous prototypes on existing road networks. This is increasingly becoming a norm in the automobile industry thanks to the cost-benefit ratio, allowing the removal of errors before reaching the absolute testing phase, where the cost of mistakes, both monetary and humanitarian, can be catastrophic. For the models to be carried out through simulation environments, this work used Matlab Simulink.

Keywords

Autonomous vehicles, safety-critical, verification, validation, testing, simulation

Contents

List of Figures

List of Tables

Lista de Acrónimos

ACC	Adaptive Cruise Control
AD	Autonomous Driving
ADAS	Advanced Driver Assistance Systems
AEB	Autonomous Emergency Braking
AI	Artificial Intelligence
AVs	Autonomous Vehicles
BBW	Brake-by-Wire
BFD	Braking Forces Distribution
BTN	Braking Threat Number
CAV	Connected Autonomous Vehicles
CAN	Controller Area Network Bus
CC	Cruise Control
CCRb	Car to Car Rear Braking
CCRm	Car to Car Rear Moving
CCRs	Car to Car Rear Stationary
CG	Center of Gravity
DENMs	Decentralized Environmental Messages
DiL	Driver-in-Loop
DOF	Degree of Freedom
EGG	Electroencephalography
EuroNCAP	European New Car Assessment Programme
ERP	Event-related potential
FCW	Foward Collision Warning
FB	Full Brake
GPS	Global Positioning System
HiL	Hardware-in-the-Loop
IMU	Inertial measurement unit
ICS	Inevitable Collision State
LC	Lane Change
LCF	Lead Car Following
LKA	Lane Keeping Assistant
LTI	Linear-Time-Invariant
MiL	Model-in-the-Loop
MIO	Most Important Object
MPC	Model Predictive Control
OEM	Original Equipment Manufacturer
PB	Partial Brake

PID	Proportional Integral Derivative
RLDA	Regularized linear discriminant analysis
RSS	Responsibility Sensitive Safety
SAE	Society of Automotive Engineers
SiL	Software-in-the-Loop
SMC	Statistical Model Checking
STN	Steering Threat Number
SoS	Systems-of-Systems
TTC	Time-to-Collition
UDP	User Datagram Protocol
USB	Universal Serial Bus
VTH	Variable time headway
V2I	Vehicle to Infrastructure
V2P	Vehicle to Pedestrian
V2X	Vehicle to Everything
VVT	Verification, Validation, and Testing
V&V	Verification and Validation

Chapter 1

Introduction

Autonomous Vehicles (AVs) are an emerging topic that has increasingly gained attention in recent years, with universities and automobile corporations worldwide investing significant sums to achieve a breakthrough in innovation in a highly competitive environment. The focus of the automotive industry, especially the research department, has shifted from Advanced Driver Assistance Systems (ADAS) to Autonomous Driving (AD), which are more complex and expensive, especially in actual scenario trials, where the complexity of the environment is so great that is entirely impossible to run all necessary scenarios to ensure the highest of standards in relation with VVT. AD aims to allow the vehicle to best respond to external conditions, presenting some degree of intelligence, which naturally requires high complexity at both hardware and software levels.

Furthermore, recent developments and technological advances in these fields allow AVs to be capable of sensing their environment (through a multitude of sensors with greater precision), with improved decision-making (through the evolution of Artificial Intelligence (AI)), and moving safely with reduced or no human interaction. With this last attribute in mind, safety is a priority in all transportation sectors, including autonomous airplanes, trains, boats, helicopters, and recently the arrival of drones, with the implementation of safe areas [1]. Who have their safety issues and challenges, but we will consider only automobiles for this paper. The main goal in improving road safety is to reduce accident rates, which may sound simple but has tremendous implications. The reduction of accidents will also reduce traffic congestion, which on its own improves the general driving experience in the city, as well as solving economic issues, not only for the people who own the car but also for road infrastructure managers, who can interfere in fixing road protection barriers, allowing the improvement of urban mobility and city design.

AVs are the next step in the evolution of the automobile industry, aiming to mitigate human error and provide a shallow intrusion way to optimize the driving process. Consequently, all these innovations, developments, and complexity require an improvement in how developers can carry out tests and experiences with low-cost and fully controlled test cases. Especially when verifying, validating, and testing are associated with safety-critical systems, it can affect the vehicle's integrity and the driver's life if it fails.

In a real-world scenario, too many variables are at play, exponentially increasing all the test cases necessary for software verification and validation (V&V). Automating the same test systems is essential to guarantee the software's quality, even before hardware integration. This creation of independent unit tests allows us to ensure a certain versatility in the development of AVs.

However, standard test cases have the problem of present scenarios, where the vehicle can operate safely in an autonomous mode without regard for geographic, environmental, traffic-related, and temporal limitations, comprising all the results and attributes hiding the AV's fundamental performance. A 2018 fatal accident involving a Tesla car and a truck showed the necessity of human supervision in autonomous vehicles at the current stage of evolution. The accident happened due to the inability of the car's artificial vision system in autopilot mode to detect a white truck on a cloudy sky background [2].

With that in mind, software verification and validation should be an extreme priority before investing significant sums of money in real-world tests. This process may also include dangerous situations and conditions that rarely happen in real traffic scenarios. This process is based on prevention and testing in the early stages of system development, as it is more accessible and affordable to find and fix such problems, even before writing a single line of code.

For this, the V-Model is gradually becoming the norm in all autonomous vehicle industries, but also in all the development and integration of electronic systems. A reason for this is the steady growth of technologies to test all the elements that make up the system, from the model-in-loop simulation to vehicles-in-loop simulations, passing by software-in-loop, hardware-in-loop, and other stages that may be necessary for the development of an AV. This way of decomposing the entire system into smaller parts and gradually integrating, testing, and validating them is why the V-Model is slowly gaining preference in the automobile industry.

1.1 Objectives

The main objective of this master's thesis is the proposal of an Autonomous Emergency Braking System, which belongs to the domain of Systems-of-Systems (SoS). It is a relatively new field of computer science that focuses on designing and operating large systems composed of autonomous constituent systems [3]. To achieve the established goal, a literature review was carried out on the current technology used, articles written, and worldwide theories and ideas associated with autonomous and connected vehicles and how to test these systems, acquiring sufficient knowledge of this matter to develop a new way to combine specific points of ideas to reach a unique solution.

Furthermore, a simulation of the braking system will be implemented, aiming to analyze the performance in previously created fully autonomous vehicles, which this work intends to continue, integrating a fundamental part of the system, the braking system, in case of emergency in critical highway scenarios. The goal is to prevent all car accidents in various scenarios, avoiding any collision and damage to the vehicle and ensuring that the passenger has minimal discomfort during the process. It is important to note that it is still an emergency braking, so removing all discomfort caused by the abrupt stopping is practically impossible. Nevertheless, it can be mitigated.

1.2 Document Structure

To reflect the work carried out, this document is structured as follows:

1. **Introduction** – Presents an introduction to the topic, the objectives, and the structure of this dissertation.

2. **State of the Art** – Describes the most important concepts within this project's scope and an overview of the subject, presenting several articles, ideas, and projects developed over time.

3. **Study of the test bench for Highway Path Planner with Autonomous Emergency Braking** – Explains with more detail the entire process of construction of the test bench, giving a more comprehensive exposition of the Path Planner, vehicle dynamics, the Adaptative Model Predictive Control and Autonomous Emergency Braking.

4. **Tests, Results, and Discussion** – The results obtained are described and analyzed by comparing two models. The overall performance of the model is discussed in detail.

5. **Conclusions and Future Works** – Summarise with an overall work accomplished on the dissertation and indicate possible future works around this topic.

Chapter 2

State of the Art

2.1 Introduction

This chapter will expose the origin and concepts associated with autonomous emergency braking. Namely, the studies and technologies developed that have proved to be an emerging topic have gained more notoriety in recent years.
It highlights some of the many methods and ideas exposed as a way of understanding a theme that, in one way or another, influences all people in society as the use of automobiles has gradually played a vital role in every person's life. It is necessary for almost all types of functions, and with a highly competitive industry where product diversification is increasingly associated with technology, especially software improvement (reliability and quality).

2.2 Automotive Industry, ADAS to AD

The automobile is a self-locomotion project that allows the transport of passengers, and this idea has existed for some time. However, in the last century, it has reached new levels of innovation in several areas, even being a pioneer in new technologies. Examples of this are the mass use of the production line that allowed Henry Ford to increase the productivity of the Model T construction, and it took little time for other companies to adopt this method and start mass-producing various types of cars on the market. The automobile became widely available to the consumer, with increasingly reduced prices. However, after some time, a greater variety of brands appeared, each betting on different types of innovations (namely mechanical components) or taking advantage of the fame resulting from sports activities associated with the automotive industry. In recent decades, with the emergence of digitalization, a change in the paradigm of the automotive industry has been noticed, with the different brands increasingly betting on more significant innovation in the technology and software used in the automobile. Figure 2.1 represents this evolution in car research.
The first type of system to appear was the Advanced Driver Assistance System (ADAS), an electronic system embedded into vehicles to automate them and assist drivers while driving. In ADAS, the primary objective is to help the driver by automating some of the multi-tasks associated with the driving process. Several examples include Adaptive Cruise Control (ACC), collision warning or avoidance, pedestrian collision avoidance or mitigation, driver alert to other vehicles, lane centering, automated parking, and more. ACC is at the forefront of these technologies, a system incorporated into road vehicles to adjust speed and keep them safe from the traffic ahead. Another is the Autonomous Emergency Braking System (AEBS), used to identify imminent collisions and prevent or reduce their severity. The AEBS has the principle of avoiding collisions, and for that, it is mainly used as a metric that defines the

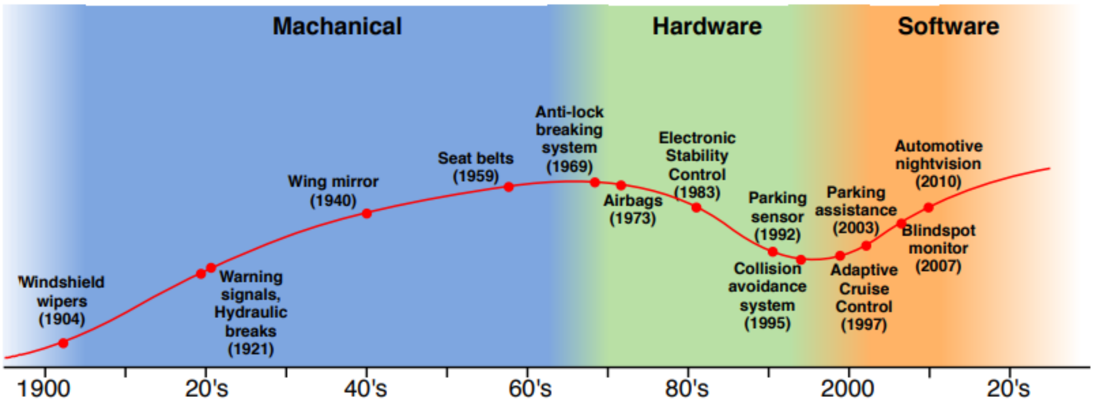

Figure 2.1: Research Evolution on Automobile Industry

shortest threshold time at which the vehicle can be stopped to avoid collision, called Time-To-Collision (TTC). Another critical factor is the Stopping Distance, which consists of the sum of the distance traveled by the vehicle at the same time that the driver reacts, plus the distance traveled by the car while decelerating to a stop. It is important to note that the ability of an AEBS to detect dynamic (moving) obstacles will directly influence collision avoidance, so sensor quality is of extreme importance for all AEBS because it is what will allow the precise calculation of TTC, which will initialize the braking process. There are several AEBS in the market, like Volvo City Safety, a safety protection system developed by Volvo in 2007 that relies on laser sensors to measure distances. Benz's Pre-Safe warning safety system uses ESP and BAS sensors to predict hazards and trigger mechanisms to protect drivers and passengers[4]. Nevertheless, the development of new technologies for AEBS, like the incorporation of an AEBS based on road friction, fuzzy controller based on pedestrian collision avoidance, deep reinforcement learning, Markov decision process, random forest, and dynamics learning tree are allowing the new systems to be more effective and to outperform existing systems. However, this system has a significant margin of improvement and evolution until it reaches its full potential as an AV.

To start standardizing all types of automation associated with the automotive industry, the Society of Automotive Engineers (SAE) defines five levels of automated driving systems adopted by the National Highway Traffic Safety Administration [5].

1. Level 0 refers to no automation, where all primary controls are with the driver.
2. Level 1 is called function-specific automation, where the vehicle has limited authority, like maintaining the speed via cruise control.
3. Level 2, at least two primary control functions are expected to work together, like controlling the steering and acceleration or deceleration by obtaining the driving environment information.
4. Level 3 refers to limited or conditional self-driving automation. It allows the vehicle to control safety-critical functions. However, driver control is occasionally expected,

especially in dangerous or complex situations.

5. Level 4 incorporates self-driving automation (high automation but not total control). The vehicle has complete control, performs all functions, controls all safety-critical driving functions, and monitors the driving environment, but is limited to a few scenarios.

6. Level 5 refers to fully autonomous vehicles or self-driving vehicles (i.e., automated driving or driverless mode) capable of driving under all scenarios

As established by SAE, these (sub)systems ADAS are positioned in SAE levels 0, 1, and 2. On the other hand, the result of automated driving is an autonomous car (with SAE Level equal to or superior to 3), where at this stage, no human driver needs to be active in the control of the vehicle. Nonetheless, the Autonomous Driving (AD) system can take control of the entire car and be in charge of the driving process when authorized. Indeed, implementing an AD is no more than the symbiotic relationship between existing ADAS in the vehicle, working like a central intelligence that commands all the different subsystems to accomplish the goal of self-driving. Nevertheless, why not only use ADAS? Or why is a more complex system needed to combine all independent sub/systems in one AV?
There are several reasons, but the principle, as stated by the article in Human Awareness versus Autonomous Vehicles [5] shows that even such devices classified as ADAS are not able to significantly reduce the number of accidents as humans require 0.3 - 3.5 seconds to respond to the stimulus appropriately. The variability of this value is usually dependent on human awareness. In a more complex situation, humans need to think out a strategy (which is not trivial in specific situations) and react with accurate behavior. Reports show that, in the laboratory, an upper value for the human reaction is 0.78 seconds. In an urban environment, it increases to 2.5 seconds, mainly due to information overflow that the driver receives from the environment. However, the study concluded that the average human reaction time is 12% slower than an automated vehicle's slowest steering reaction and even 2.6 times slower than the slowest emergency braking. That is why systems with low control or warning systems like ADAS are insufficient to prevent accidents and guarantee passenger safety. The necessity of AV is the need for safety-critical functions capable of ensuring the passenger's safety even if he is unaware of how dangerous that situation is or if he cannot prevent the accident (physical disability, distraction, stress, among others). Furthermore, considerable automation can also bring advantages like decreasing the driver's repetitive tasks and increasing productivity and satisfaction in the driving process.
However, a significant problem with automatization, as stated in the paper on Forward Collision Avoidance Systems [6] is that, despite progresses being made in achieving human-like vehicle controls for active safety by taking into consideration human drivers' driving behavior, many algorithms are either too complicated for real-time implementation, or lack accuracy in reflecting drivers' actual characteristics, adding even substantial expense in the development of these new types of AVs. Of course, it has increased the lack of trust and confidence between buyers and products [7]. It claims that Av's primary goal is to give an option for an agent to help achieve an individual's goals in a situation characterized by uncertainty

and vulnerability, classifying buyers' trust in automation products into three dimensions. Being them, performance (what the system does), process (how they build a system), and purpose (why the system does something). Studies have shown that 97% of the information people receive comes from vision and hearing, and the combination of speech and ambient light directly influences their average driving performance.

Furthermore, most participants indicate that the speech plus ambient light prompts significantly impact their driving choices. All this is extremely important to achieving the perfect relationship between the car and the user. Small details like the use of ambient light can increase trust in both AVs and ADAS systems, and both also have an impact on the reduction of time reaction time of the passenger when it comes to emergency braking. Other ways to improve their trust can be accomplished by using a more transparent development process like the V-model with several layers of testing and V&V.

The European New Car Assessment Programme (EuroNCAP) introduced the AEBS test in 2014, accelerating the penetration of ADAS in Europe. Also, the DARPA challenge started in 2004 and accelerated the research for autonomous driving. Several OEMs and universities have demonstrated autonomous driving cars [8]. There is a starting tendency to introduce AVs and ADAS after they fulfill specific standardized tests. This necessity for standardized test cases shows the rapid evolution of simulation and software tests, proving that simulation, virtual tests, and V&V of software is becoming a significant standard in the automobile industry, especially in the race to create a fully automatic car. The three major components are Mission Planning, Behaviors (Maneuver Planning), and Path/Motion Planning, in which each part of this system must be able to act automatically on a gigantic set of variables, and these must always guarantee the integrity and safety of the car and its passengers. This research, as imagined, can be extended for all means of transport, especially if we are considering the AEB. The VVT process can also be applied to trucks, plains, or drones (in case of emergency braking after landing) [9].

2.3 Connected and Autonomous Vehicle

Another critical term to the automated car industry is Connected Autonomous Vehicles (CAV), which adds new complexity to AV systems. One aspect of completely autonomous systems is the ability to communicate with other autonomous systems and the surrounding environment, including communication with infrastructure or pedestrians. In addition, the reasons for the existence of this same communication may vary. Still, it always will have the main objective of gaining extra information about situations or environments where it finds itself (for example, information about accidents or traffic or remote control may imply that some deceleration can be applied) and self-awareness.

The network communication technologies are divided into several classes: vehicle-vehicle (V2V), vehicle-infrastructure (V2I), vehicle-pedestrian (V2P), and more depending on the type of elements present in the established communication, being all incorporated in a large class of vehicle-everything (V2X) technology. They are used to facilitate the exchange of information between the car and the outside world. This communication can have a vital purpose of improving the gain of information on high-risk cases, like passing by a blind area as represented in Figure 2.2 or poor visibility area that contains a high risk of collision with potential darting-out pedestrians [10]. One of the problems with sensors and other systems used in the world is the detection of blind spots. This problem is because radio waves, laser beams, or camera visibility can be highly inefficient in detecting pedestrians passing by or behind much larger objects, where these objects act as barriers that obstruct pedestrian detection. As such, we can state that communication is vital to the vehicle to gain complementary information to improve his awareness rather than be limited by only peripheral perception. The communication between the traffic light and the car can force the latter to take preventive actions in case a crosswalk with pedestrians might be nearby.

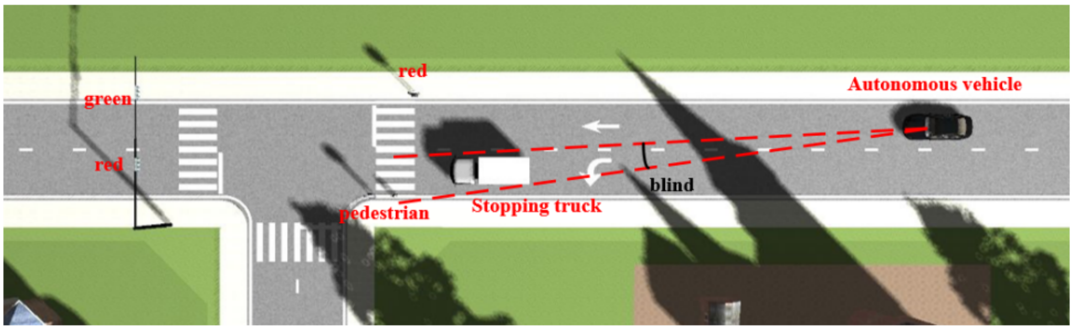

Figure 2.2: The blind positional relationship between pedestrian and vehicle

A more recent aspect of CAVs has gained increasing importance, as it is associated with Platooning systems. V2X can be used to deliver critical safety and positional information to nearby vehicles, as well as help shape traffic patterns on a regional scale. Applying that shared information allows for compacted cars in a group, where every member knows the data relative to the global position, acceleration, position, and speed of every member in the platoon. Generally, like in nature, there is a leader who is responsible for determining the planning, path tracking, and the determinations of best maneuver, and the follower's main objective is to provide the leader with the best and most relevant information but also find the best way to mimic the path used by the leader. Usually, the distance in vehicles between members of the same platoon is very short to avoid significant interference in the platoon's order and structure. With many cars packed closely in the platoon, it can easily provoke damage. A malicious vehicle could create a devastating pile-up by braking faster than following vehicles could react, making a considerable problem for platoon systems. Even with instantaneous detection and response, two cars traveling at 100 km/h (62 mph) with the preceding vehicle braking at 1G (approximately 9.8$m/s2$) and the trailing vehicle braking at 0.9G will require 4.32 meters of following distance to avoid a collision or to avoid the vehicle to harm its occupants. An even bigger problem is the action of malicious cars or software. For example, suppose the lead vehicle drives itself off a cliff. In that case, it is entirely outside the scope of platooning or contracts [11], and any interference, error, or malicious act in the front vehicles can generate a catastrophic snowball effect of events with enormous loss. Situations like that are prevented by defining protocols and contracts for the communication system with two primary objectives. The first is a process for recovery from a communications failure, security violation, or integrity assurance, and the second is defining principles of validation and security of information as well as having norms that the vehicle can implement in case of emergency or if it detects a violation of security. All this must be defined and clarified in contracts previously established and agreed upon by all platoon members, with many clauses covering all scenarios the vehicle may encounter.

In platoon formation, it is possible to achieve aerodynamic benefits when traveling at such close distances by using the vehicle in front as a shield. This uses the magnitude of aerodynamic forces on the platoon's behalf by reducing the air resistance. It has shown that the inter-vehicle distance needs to be reduced to 2.5 meters or less to enjoy this state, thus allowing reduced fuel/energy consumption and an easier way to maintain an aggregated platoon. However, there is the downside of making braking maneuvers even more dangerous due to the braking ability of a platoon, which corresponds to the vehicle with the worst AEB performance. If a third car joins a two-car platoon, with a maximum deceleration of 0.55G, then the whole platoon's braking capacity is now 0.55G [12]. However, if this third car can achieve a higher deceleration, then the entire platoon's deceleration is the same when braking. In this case, the vehicle's deceleration to enter a platoon has two necessities: achieving a higher or equal value than the vehicle in front and higher or equal deceleration than the worse vehicle in the platoon (in the perfect situation, this will be the leader). Indirectly, this implies that any vehicle in the rear (with a higher deceleration margin) has more time and distance to brake than required because it consistently outperforms the front vehicle. That also means vehicles with better braking systems should have a further back position. This type of implementation can only be achieved by non-passenger vehicles because only 2.5 meters to achieve a higher aerodynamic advantage can compromise human safety and the risk of injuries.

Other studies made aboard cases like autonomous vehicle following systems in an off-road environment. In this case, assuming that the front vehicle is man-driven and the following vehicle is an AV, using a combination of multi-sensor fusion (millimeter-wave radar and Lidar) and V2V, this system aims to be used in some rugged terrain, with low speed but a high variation of longitudinal, lateral position and yam rotation. The V2V system requires high accuracy and high precision for this type of data sent to the following vehicle, so both lead card data collection and transmission must be impeccable. Hence, the system adaption to the complex off-road terrain is challenging. Combining the detection capability of sensors with the data received by the communications and applying both sources of information on a fuzzy algorithm can create a system capable of following the front vehicle in more irregular terrain [13]. This technology also has the potential to be used by firefighters, military, and rescue teams, allowing them and the AV to transport equipment and materials through harsh terrain with minimal human resources required and freeing and allocating human resources in more urgent tasks. However, in this scenario, there are only two participants which is highly inaccurate for the highway scenario.

However, this proves that communication can also be essential in following and avoiding vehicle collisions. Both V2V and V2I can be used to improve emergency contingency maneuvers. The difference in emergency contingency is that the emergency action is accomplished by first reducing the vehicle speed to the possible point and second avoiding all possible vehicle trajectories ahead of the occupied area [14], creating an alternative motion path alternative to a braking maneuver (when these are not a variable option). This feature allows the AV's path planning to buy time crucial to safe driving performance. It is the primary reason to use a 3-stage AEB, where the two first stages will allow the AV to gain some precious time and analyze and calculate other alternatives before a full brake. If the scenario presents an alternative to full brake, the system will generate safe motion plans by computing and setting up emergency maneuvers with prior conditions. Sometimes, the awareness of these possible alternative trajectories is only present through vehicle communication, which is fundamental to a better understanding and perspective of the car ahead and road structure (numbers of vehicles ahead, blind spots, crosswalks, garage entrances, and more). However, this is limited to cases where an alternative safe to the braking process is possible, and simply slowing down the vehicle creates enough time for an emergency contingency trajectory to appear, if they are even possible. Even smooth emergency braking can be considered a better option for abrupt changes in trajectory. Alternatively, for scenarios where the vehicle has no time to compute all path planning, the road infrastructure could also plan these alternative paths and transmit them to vehicles in a collaborative effort, taking advantage of the position of the road infrastructure.

As in most computation science, the ability to send messages between various elements in a closed system can be challenging, and it is much worse in an open system like a highway that can have all types of communications, components, and all variables previously discussed, occurring at the same time and constantly changing adding a complex dynamism to the problem. For that, studies like Decentralized Environmental Messages (DENMs), [15] aims to solve this problem using asynchronous communication to alert vehicles to road hazards such as abnormal traffic conditions, road works or signal violations, and stopped vehicles. They repeatedly transmit this message in a relevant area, increasing message reliability during the event's lifetime. Compared with other cooperative awareness messages, the DENMs can provide more repetitions (between 5 and 10). Short-sensing allows sending large quantities of positive data with more extensive areas (above 250m). However, the improved other means of communication, as is the example of iDrive systems, connected through a series of communication layers, are presented as alternatives, consisting of three communication protocols: User Datagram Protocol (UDP), Universal Serial Bus (USB), and Controller Area Network Bus (CAN). The main goal of all these developments is to show that it is possible, especially for V2I, to create means to develop a system for better distribution of information to automobiles from a grid of roads within a city, allowing for an improvement in the city's management, namely in the reduction of congestion and road traffic, giving drivers knowledge in advance which behaviors they should adopt on certain roads.

The taxonomy of coordination and communications in a road environment can be divided into resource and task-oriented dimensions. The first is about acquiring access to a shared limited resource, whereas the second is about vehicles being able to accomplish a specific task. Other parameters must be defined, such as safety (what should never happen), liveness (what is necessary to fulfill the goal), and a quality measure, which is very subjective and varies from every problem interpretation. Furthermore, let us consider the most usual cases

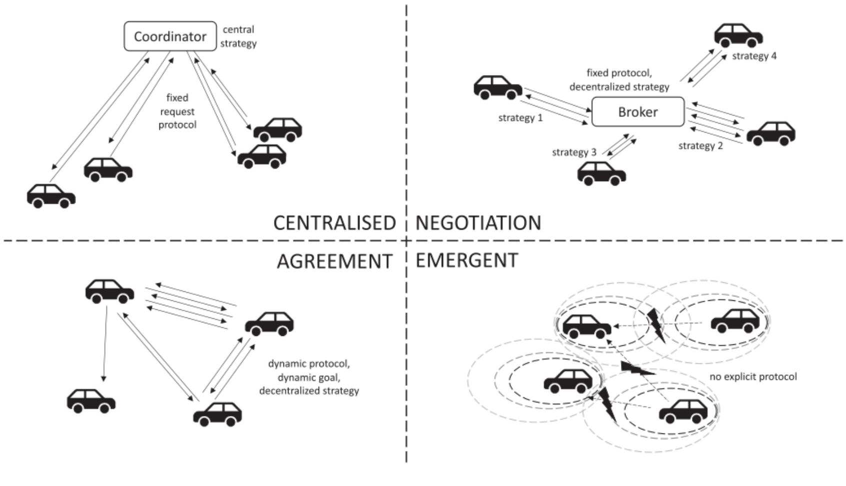

Figure 2.3: Four classes of coordination autonomy

of road communications, such as intersection management, smart parking, ride-sharing, ramp merging, platooning, and traffic flow optimization. In these scenarios, we can see these dimensions very clearly. For example, a platoon approach can seem like a task-oriented coordination problem, where each tries to replicate the leading vehicle's maneuvers. As such, the system is cooperative, as they need to work together to achieve platoon efficiency, where avoiding collisions is a measurement of safety, and finally, the ability to preserve the structure of the platoon defines the liveness and quality of the system. However, some studies may try a more competitive-based approach where each platoon member tries to compete for the best spot (where they can save more fuel, be close to the lead vehicle, or have a more aerodynamic advantage). Moreover, considering every dimension and every example, they can also be divided into four classes shown in Figure 2.3, centralized which exists an individual computational coordinator that streamlines all involved. The negotiation relies on the vehicle adhering to a specific negotiation protocol that usually requires a broker to mediate. Agreement necessitates vehicles to enter a dynamic protocol defined by themselves in a collective effort (adopting argumentation technologies). Finally, emergent vehicles do not explicitly engage in any coordination protocol and shared goal of reaching any agreement, focusing on individual perceptions to define their behavior [16], being the most personal approach and the norm to no CAV vehicles. They still have some communication capabilities, but this communication needs a high rule in the driving process.

2.4 AEB Systems

Research has been done in the vehicle active safety field to reduce these issues, which subsequently prompted the introduction of the already mentioned ADAS. Among these notable applications is the Autonomous Emergency Braking (AEB). There are several products implemented on the roads in the last five years, like Volvo City Safety, VW Front Assist by Volkswagen, and the 2010 Mercedes-Benz PRE-SAFER Brake [17]. Despite this recent technology, there is space to improve and innovate, as shown by a study of the AEB system integrated with the Potential Field risk assessment strategy. The newest ideologies on measuring risk and ensuring more efficient and successful collision mitigation are essential to achieving a minimum safe distance to the frontal obstacle after braking without risking passengers' lives. One problem is that most executed tests represent a forward-motion vehicle with few road lane scenarios and limited diversity in the complexity of the environment. However, many other factors and techniques exist for creating AEB systems, each with advantages and disadvantages. For this study, we must highlight that the objective of creating this AEB system is to be completely autonomous and able to be integrated into the AV, and it needs to be a more complex system than the ADAS standard AEB. Nevertheless, why implement an AEB system in the first place?

Conclusions point out that in car-motorcycle accidents, in 50% of cases, the cause of the crash was related to the failure of the car driver to observe the oncoming motorcycle and the realization of the actual dangerous situation. Moreover, studies reported similar results: the predominant collision partner was a passenger car (56%). Most crashes occurred in urban areas (62%), typically at intersections (48%). In 37% of cases, the errors were assigned to riders and "involved skills that were beyond those that typical drivers or operators might currently have" [18]. The study proves that 49 out of 90 crashes (54%) were potentially avoidable if AEB was present in the passenger car. However, the absolute number of crashes that could have been avoided directly depends on the exact initial position of the vehicles, which in most cases is not known with total accuracy without in-depth crash investigations. The impact speeds of real-world crashes did not differ substantially between those non-avoidable and avoidable for these cases. Most scenarios related to a specific area of AEB are related to collisions between 4-wheel and 2-wheel vehicles. Indeed, more cases may need an AEB intervention, such as collisions of cars on highways, animals crossing roads, random objects dropped on highways, roadblocks, and so forth. The main goal of an AEB system is to reduce the possible injury outcomes for the individuals involved to a minimum or, in the best-case scenario, avoid them entirely. It is also vital to prevent the integrity or structural damages to the vehicle (so as not to set the car on fire while breaking) and ensure the minimum of discomfort a passenger may have while performing an emergency braking if there are conditions for that process. In practical terms, the AEB triggering algorithm will typically follow these simple steps: First, obtain current state variables from V2V or sensor measurements; Second, define the state variables according to the rules used to describe a potential collision eminence (the definition of this rules is where occurs the significant diversity between the different products presented); Third, if occurs a rule violation and it is proven that the collision will occur, the AEB is triggered on the passenger car. They provoke the AEB to increase the braking action automatically in the event of an imminent collision. As defined, the system will be triggered only when a coming conflict leads to an inevitable collision, no matter what action of the host vehicle or what the opponent vehicle tries to implement [19]. This rule means that full braking should only be applied as a last-resort mechanism to avoid a collision. However, implementing other steps should give time for AV to find an alternative solution or even allow the road environment to change so that it is no longer necessary to apply the whole braking step. It is essential to mention that the majority of systems present in this section only focus on all three steps and try to mitigate or ignore the impact of the other two steps to help achieve a better braking system. The road leading to an emergency braking should also be analyzed.

The most important aspect is the definition of the risk index, which combines a judgment on the forward collision risk and TTC or even other risk assessment metrics. Recent approaches divided the risk index into four steps. In the standard scenario, we have the normal phase, where other AV systems or subsystems control the vehicle's behavior. Then, the first step, which occurs when there is a possibility of a collision, starts with alarming the passenger through warning signals. The second step is to apply a smooth deceleration or partial braking to reduce the overall velocity, allowing the system to compute and define course actions or alternatives to the complete braking process. The third step, omitted sometimes, is similar to the second, but the deceleration value is superior (another alternative is to combine both and make a gradual deceleration). Finally, if the collision is unavoidable and imminent, apply full braking [20]. However, in some scenarios, standard systems could not accurately extract the relative distance, a parameter required for risk calculation, because it could not reflect the geometrical factor of the curve. This error shows another vital problem related to non-linear roads, specific roads with angle degree variations or roads where distance calculation is not so simple or linear, but this isn't a norm for this branch of study. Every solution presented compromises itself to solve one specific scenario from a large number of possible scenarios in a road environment because too many variables can influence the whole AV, and there needs to be an optimal solution and more test cases for all scenarios.

Another important variant in the braking scenario is the environmental behavior around the vehicle during and after the activation of an AEB because this is a high-risk maneuver that should only be activated as the last resource to avoid an accident. The behavior of other vehicles, specified in a concise future, must be considered to prevent accidents between cars and the primary vehicle. For that, there is a branch for AEB control strategies [21] that presents a mechanism to prevent collision with a leading car close by, which may occur when the ego vehicle performs autonomous emergency braking. Generally, the AEB system performs braking at a very high deceleration at the last point when collision avoidance is possible to prevent a rear collision with the leading car, where most systems design a predefined deceleration of emergency braking at a very high deceleration rate of more than 0.8G, however, is possible to see the relationship between deceleration and distance in Figure 2.4. That value can vary between approaches, and if the rear vehicle is an average car without an AEB system, this vehicle may not be able to avoid a frontal collision in this situation with the ego vehicle when this performance a full brake the collision. Furthermore, the scenario of low friction or frozen road can lead vehicles to skid out of control and severely affect the road friction and wheel attraction, and this can provoke an uncontrollable variation of lateral and longitudinal movement of the ego vehicle, leading it to collide with cars traveling parallel to this or in front. These are more examples of the necessity for more studies related to this case scenario and an urgency to improve the VVT of software related to the automobile industry, including ADAS, because to reach a fully autonomous vehicle, there are an enormous number of variables and test cases that are needing to be addressed. In creating a good AEB with low prices, it is vital to use VVT software to transform all processes into a more independent and agile system. Many of these AEBs, due to their complexity, are extremely expensive for developing countries, namely Asia, South America, and Africa (the primary consumer of cars for the next generations), where it is necessary to produce more low-cost or cost-efficient systems [22]. However, AEB technology is only available on 21% of all vehicles sold in Europe despite having the potential to reduce the number of accidents by up to 27%. In this case, in most of Europe, it is necessary to standardize this technology with a need for more investment in the research and development of these systems.

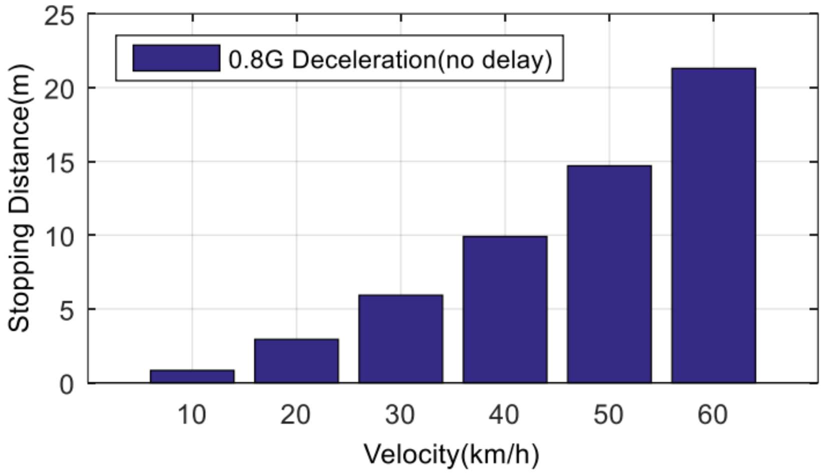

Figure 2.4: Braking distance for 0.8G deceleration with perfect conditions

High-speed accidents, especially on highways, are more likely to be fatal than low-speed collisions, even though they are less recurrent. Automatic braking systems can save lives and reduce the amount of property damage that occurs during a crash. Crash dummies are the most iconic method of VVT correlated to the automobile industry. Their behavior in test cases of high risk provides a highly reliable source of information without compromising human life. Although crash dummies can provide vital information about the severity and type of injury that the human body may suffer following involvement in a collision, they offer no information regarding the interactions within the vehicle before the crash. The focus should be placed upon the information network that governs 'who' (driver or automation subsystem) owns 'what' information and 'how' this information is communicated across the network as a whole [23]. The driver's importance is more irrelevant according to the greater autonomy of the vehicle (as defined by the SAE scale). The interactions between multiple agents can be system described as "cooperative" because, under the right circumstances, automated systems can become "a vital non-human member of the driving crew." One importance of dummies to all processes of creating an AEB is that even in the early stages, they can be used to define some essential variables, like, for example, the range of deceleration that provokes less damage in the dommies or body dislocation during the full emergency braking.
Is it possible to guarantee that AEB systems are more efficient than simple braking systems or even more efficient than only warning systems? Several studies use various methods to prove this affirmation is true, the most important being those associated with the analyses of the passenger reaction time or regularized linear discriminant analysis (RLDA) based brak-

ing intention detection algorithm using electroencephalography (EEG) signals. In January 2018, an autonomous-driving Tesla Model S rear-ended another vehicle on the highway, resulting in the driver's death. According to the investigation, the driver did not step on the brake pedal to take over the vehicle in time when he found that the autopilot system failed. Moreover, driver's facial images have been used to recognize distractions or drowsiness while physiological activity measurements were employed to detect fatigue [24]. Like any other human action, cognitive processes precede any brake pedal deflection, which makes the method based on brake pedal deflection unsatisfactory in terms of time performance, where braking intention detection can reach an accuracy of nearly 90%. Still, the advancement only goes 189 ms. However, it can still correlate with the idea that subtasks negatively impact the driver's attention to the road condition. ADAS has an irrefutable paper in reducing the driver subtasks, consequently improving driver concentration and reducing accident risk. Usually, these studies demonstrate this correlation using event-related potential (ERP) and machine learning techniques to analyze and classify the signals of road events. Shows results where the responses are 454 ± 234 ms before the reaction (actual button-pressing response), and the average recognition accuracy of the RLDA classifier reached 95.81% [25]. Some of them state that the EEG signals of a passenger can reflect the emergency road event and can be utilized to control the vehicle in an emergency to prevent collisions. It can present itself as a complementary data source to obtain information about the environment (like sensors), but its true potential has two significant main obstacles. First and more evident is that EEG signals are utterly dependent on the passenger's awareness. Suppose he cannot see or understand that he is in a situation of danger (bind spots or passenger falls asleep). In that case, EEG signals will be completely normal, so they will not be enough to provide a valid risk assessment. It is necessary to incorporate this with other systems, such as radar, cameras, or other sensors, to achieve the desired performance. Secondly, the response time is only in the house of 454ms, with an error margin of 234ms. In a worst-case scenario, the reaction will only occur 200ms before the action, which, for the expensive account, the cost of training machine learning and cost with EEG sensors can not be economically viable for the market, being easily surpassed by systems with a better cost-efficiency ratio, such as radar or LIDAR technologies. One crucial fact correlated with these studies is that age and gender effects are not critical. The relation between driving experience and response time has proven that this could affect the performance of the brain-computer interface system related to the control of the autonomous vehicle.

AEB systems have great importance, not only to outperform the human reaction and capability of awareness (commonly limitation associated vision or human fatigue) but to be able to transform the driving experience from a more risky activity to an enjoyable experience, and this is primarily a norm for people whose jobs depend on large amounts of time invested in driving vehicles over long distances and for long hours. Using these technologies can be a safe bet to improve transportation companies, whether these are goods or products or the transport of passengers.

2.5 AI vs MPC

Many approaches are related to AEB, but it is possible to divide them into two main components; the first and most used among the two is Model Predictive Control. MPC is the core of AV technologies, essential to implement all ADAS types, from ACC to collision avoidance, passing by lane centering, and pathfinding. Thanks to using models capable of predicting the future by combining the physics of the primary car, environment behavior, and expected behavior in an imminent future (based on acceleration, velocity, movement, and laws of forces and motion). The second component is Artificial Intelligence techniques like fuzzy algorithms, which are being developed and implemented in some ADAS functionalities like ACC. AI determines what values will be inserted in the MPC. However, IA has a prominent rule chiefly to enhance sensor analyses, mainly object detection, especially by cameras and decision-making. At the same time, the MPC transforms this pretended behavior into action to control the vehicle dynamics. Despite operating in different sectors of an AEB system, both methods are vital components for its creation.
State of the art in planning methods roughly falls into three groups. First are the sampling-based methods, where the conditions and input variables are discretized or randomly sampled in lattices, and then efficient solutions for deterministic or stochastic searching. The second group implements decoupling schemes for planning the global path and calculating the speed for local obstacle avoidance. A third group of planning algorithms involves mathematical constrained optimization formulations that offer guaranteed consideration to conditional existence (physics of motion and force of dynamic objects) and optimality solution based on the problem formulation [26]. This project mainly belongs to the last group. It offers a convenient and optimal planning method that can accommodate dynamic changes in the environment of the ego vehicle via frequent updates on its input information and formulate the optimal solution based on predicted near-future states and their respective costs and reliability.
The obstacle avoidance methods of intelligent vehicles are summarized into two categories. Firstly, if the trajectory planning judges that it can avoid obstacles by changing lanes, it will change lanes if the risk assessment determines any possibility of collision and any alternative maneuver to avoid that collision. Secondly, if the vehicle judges it cannot avoid obstacles, it will adopt the braking strategy. As we can see, the wheel's desirable behavior is determined in all AEB systems based on judgments that the Path Planning Process effectuates over an optimization problem where the main variables are typically associated with risk assessment, setting constraints, and objective function. Then, this information is given to MPC, which converts all measures to guarantee that the vehicle dynamics (wheels, brakes, pedals) perform in a way intended for that situation. For that, measurements like acceleration, velocity, TTC, tire data, vehicle mass, inertia properties, distance, time gap, road friction, object detection, headway, and other determinations of that object's measurements are central to achieving the optimal solution for the defined multi-constrains problem [27]. They also require high update rates consistent with the dynamics, with an excellent requirement for high-quality sensors for the MPC to compute the feasible and optimal control inputs for the controlled vehicle.

One of the factors responsible for the variable of solutions present is associated with metrics chosen to send to the Planning Algorithm and MPC, with many systems opting for TTC or distance gap as the principal font of risk assessment. However, alternatives like variable time headway (VTH) [28] exist. Like the others, they share the same variables like acceleration, velocity, distance, and time, and the correlation between these variables differentiates all approaches. Some prefer to give more importance to the time gap between vehicles, while others use distance. Some prefer to focus mainly on the states of the ego vehicles, others on the leading or following vehicle or MIOs behavior. The relation between these proposals varies according to the specification of the desired product. For example, the standard deviation of deceleration with the VTH model is 1.855 m/s^2. In contrast, the constant time headway model reaches 2.245 m/s^2 because VTH is more considerate with time variation between states. Although the time headway model sends out the collision warning signal earlier than the VTH model, the earlier collision signal will affect the premature sending of the braking measures.

Generally, the vehicle safety and failure protection mechanisms are implemented in one of three stages (perception, decision-making, and execution) during the self-driving operation controlled by MPC. Responsibility Sensitive Safety (RSS) is a mathematical model of AV safety that functions after the decision-making flow before the execution process. This algorithm indicates the safe longitudinal and lateral distance during the self-driving period to ensure that the final decision-process result is the one with less possibility of a collision accident [29]. The overall architecture of the implementation of RSS is in Figure 2.5. Even though the RSS concept has been mainly and widely applied in perception and decision-making, its design works more like a diagnostic and failure protection system by the driving path and visual-based road information for autonomous vehicles, and typically, its structure developed parallel to the original decision-making process. Working as a parallel stage that complements the other three main steps, it is highly efficient in executing the strategy to confirm in real-time if self-driving actions are safe and comply with road rules. However, implementing a parallel system has increased computation costs, making this approach slightly less appreciable than other methods presented.

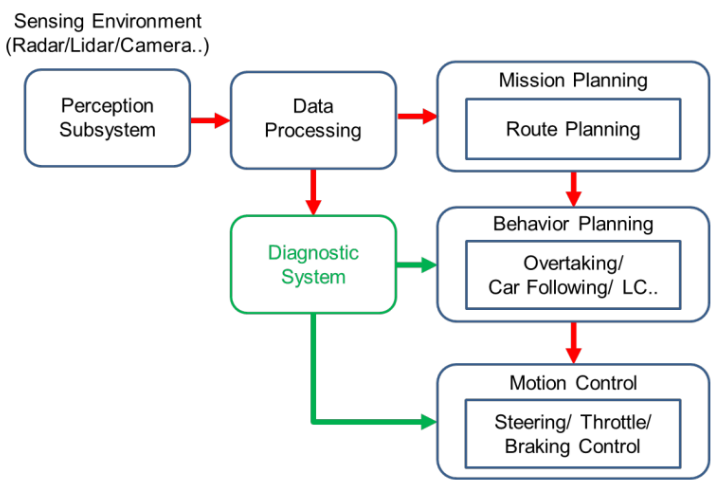

Figure 2.5: Architecture using the diagnostic system for AV

Autonomous Emergency Braking System

All the AEBs developed have proposed different methods for path planning, such as the smooth curve planner, neural-network-based and fuzzy-based control, artificial potential field, and optimal control methods [30]. Soft curve planning methods implement different techniques for path smoothing and curve generation, such as spline curves, Bézier curves, and more, with the primary goal of generating obstacle-avoiding paths according to safety parameters. In contrast, some methods tend to amylase the state condition and classify it into a determined group, according to the data collected by the perception module, that provides the necessary environmental information for accurate state classification, acting according to rules or steps pre-defined for belonging to a clustered group. Even though the two methods share the same stages, perception, decision-making, and execution, their work is very different. Where one prefers to work with states and the possible near future of these states according to pre-defined rules and optimization algorithms, the other approach prefers to work according to a dataset analysis and classification of a state according to that group's characteristics, learning and making the actions necessary to execute according what they know of his environment and the typical outcomes of that states. Nonetheless, there are some limitations to how both systems work. The first has a higher computational cost because it must repeatedly generate many future states. However, a more frequent problem present by second is when a state can present itself with a significant number of variables in enormous and ambiguous test scenarios, especially in specific study cases. It is challenging for AI to cluster all the datasets correctly, especially in very fluid and complex scenarios like roads. This problem can provoke a wrong scenario classification and cause a lousy action, leading to a passenger fatality. The environmental specificity associated with the test case can be so excellent that it is almost impossible to have all scenarios with significant quantities in the data sets. This ambiguity problem also affects the first method but is more alarming in this AI system. All the examples in this section had always recorded to virtual test case realization as shown in Figure 2.6. These executed tests serve to evaluate smooth curve planning methods.

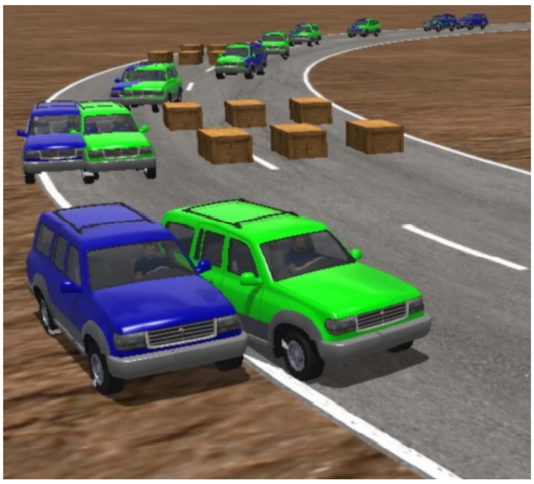

Figure 2.6: 3D virtual test using accurate dynamic models

Planning systems are responsible for decision-making involving route planning, behavioral planning, and trajectory planning, where all information needs to be defined for the MPC to act accordingly. Then, it processes this information and sends the commands to the vehicle. The planning process solves an optimization problem by minimizing a cost function of the states and defining the best vehicle outputs within imposed constraints [31]. Their complexity is based on the definition of constraints and the dynamic relationship between states and vehicle actions. However, for a system related to AEB, there is not much of a question of optimization like planning but is more of a question of risk evaluation of the near future, making MPC currently more relevant. Furthermore, the need for significant datasets of automobile accident reports, especially in the specifications of the ego vehicle and the surrounding environment, led to this same problem of insufficient data available to train AI. The lack of that information is the principal reason for a lack of solution based on AI in contrast to simple methods of state prediction, even though this presents a high computational time.

We see the differences and similarities aboard in cases. Firstly, a particular system for emergency collision avoidance when evasive maneuvers are preferred instead of traditional braking with this approach has some good ideas, namely the creation of path-replanning methods using an artificial potential field, rapidly explored random trees and timed elastic bands, which make collision-avoidance decisions and safety-oriented motions. It ends, as usual, with evaluating the optimal path to maintain vehicle safety and stability during the avoidance process [32]. However, it is more of a complementary assistant to the AEB system than a proper alternative. We can see a combination of MPC with AI, as one of the problems presented is the necessity to reduce computational effort and time. For that, measures like circle decomposition of the vehicle shape can decrease the computational cost by decomposing the vehicle into safety areas represented by risk circles, as mentioned in the safety area for drones. Even when simplifying the vehicle model in places of danger, the complexity associated with the approach has a high computational cost.

Secondly, another difficulty presented by traditional Proportional Integral Derivative (PID) controllers, as an alternative to MPC, is that they cannot effectively control the AEB system with nonlinear solid characteristics. Therefore, back-propagation was shown as a solution for this case, taking advantage of his proprieties of distributed storage, self-organizing, self-adaption, and self-learning, allowing the PID parameters to adjust in real-time, according to states and environments where the vehicle encounters. In other words, instead of using an MPC, this approach tries to combat the limitations of PID in the Planning process using AI. Back-propagation gives the AEB system great versatility to combat PID deficiency, allowing it to adapt to a changing state, presented as a possible solution for the problems previously mentioned [33]. This system focuses on finding the best time for braking by combining a controller to maintain a safe distance, a back-propagation neural network PID controller, and an inverse dynamics model, where each of these is responsible for generating the desired acceleration, control acceleration and the braking decision, respectively. The system is capable of effectively generating an appropriate deceleration for every scenario. Meanwhile, it must be tested in more complex scenarios to see its potential and implementation in other systems, especially with systems that contain path planning or cross-road scenarios. These are some examples and comparisons between the different methods presented to AVs; with AI technologies recently reaching a more critical status in the automobile industry, a new branch of approaches and techniques is appearing and challenging more traditional views on the problem. However, as proved by the next section, AI also has a relevant rule of the sensors, which on its own makes them more influential because it will impact the first stage of AV, perception, where MPC is more relevant in decision making, which is dependent on his sensor data to generate the best decisions.

2.6 Sensors

Any system of AVs contains three sequential stages: perception, decision-making, and execution. The process of the previous one will inevitably affect the operation of the next one. This emphasis puts perception systems like LiDARs, radars, cameras, and others under severe scrutiny, as it does not matter how efficient a system is at decision-making if its perception of the world has flaws. For that reason, sensors are becoming a relevant part of all sectors of technology development, especially in the branch of the Internet of Things and the Automobile Industry, as will be shown in the several types of research associated with sensors and techniques used by AEB systems.

Of the most acclaimed sensors is LiDAR, which detects the position and speed of objects around the acquisition environment with an outgoing laser beam (it uses light in the form of a pulsed laser), which is costly and widely used in the military. Ordinarily used are cameras with a large angle of view, high distribution rate, simple construction, low cost, and small size. Moreover, one of the most popular is radars, which use a beam of energy (radio waves) emitted from an antenna as they strike objects in the atmosphere. The energy scattered in all directions, some reflected directly to the radar. The main idea of various sensors on one vehicle is for them to work together to mitigate the weakness presented by each of them on their own. Commonly alone, they must be more robust to withstand environmental changes, rain, snow, haze, and other phenomena that reduce their recognition ability. Forward collision detection systems usually depend on a combination of sensors (radar, LiDAR, or camera) to detect objects or any possible obstacles from any vehicle side. However, they mainly focused on the front, with some less potent systems in the rear for parking high-speed vehicles approximation. A warning algorithm has two categories: kinematic-based and perception-based. The Perceptual-based algorithm depends on the risk calculation, such as TTC, which depends on the data like the range and speed. However, the kinematic-based approach focuses more on the minimum distance required to stop before the collision. It presents a more advanced method and requires more detailed information about speed, range, and relative deceleration rates [34]. According to the chosen plan, the input necessary may vary, but some inputs are standard and mandatory. They are velocity, acceleration, distance, and time. However, other information can be required depending on the method used and additional information obtained by correlating the mandatory inputs. We can see many sensors and optimal places to achieve the highest perception capacity in Figure 2.7.

As mentioned, sensors can have some limitations, especially in harsh weather conditions, but there are other problems. For example, automotive radar systems like iron bridges can present issues in highly cluttered road environments. The transmitted signal bounces back

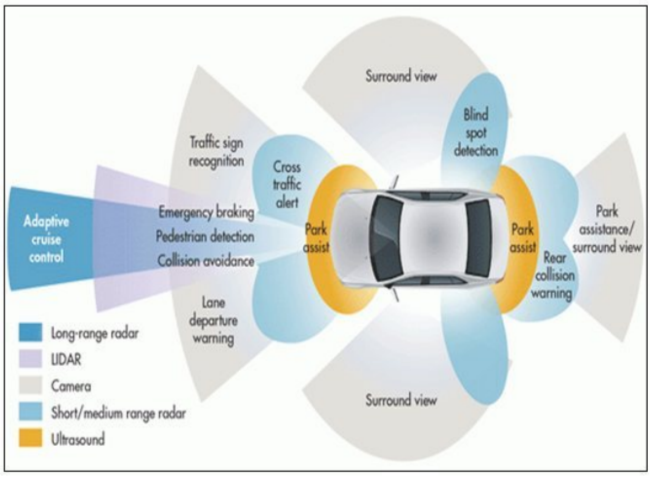

Figure 2.7: Types of Sensors and optimal position

from several metal reflectors, for example, iron pillars, guard rails, or other metallic superficies, which can reflect the wave and create a clustering challenge. Fixing this problem requires the implementation of the density-based spatial clustering of applications with noise algorithm, which uses the guard rails and iron pillars condition of being stationary and with similar relative distances from the radar-equipped vehicle and each other to cluster them in two different groups appropriately. These two groups, one on each side, represent the bridge's left and right guard rails and iron pillars. With this automatic clustering, only the points in the middle remain, which can automatically be considered the various vehicles or obstacles on the bridge road [35]. This method is highly efficient for this type of situation. However, in some cases, like when a random object falls off the truck onto the highway that can cause a severe accident, some things, depending on size and material constitution, can be invisible to radar systems. For that reason, they combine him with other sensors.

Another area for improvement with sensors is their ability to identify or classify a surrounding object. Contrary to what most people believe, converting radar waves into Micro-Doppler produced from pedestrian and bicyclist targets at the 77 GHz band can give much information about the object. The Doppler effect happens when a target has translational movement relative to the signal source where Doppler frequency modulation on the radar echoes is induced by micro-motion. This effect is called as Micro-Doppler [36]. When a pedestrian walks at a determinate velocity, the radar will only consider the body's mass center, as it will detect the object's velocity but cannot classify it. However, when analyzing the Micro-Doppler effect, it sees the different parts of the human body because they have different velocities, acceleration, and behavior profiles to mass center. Since pedestrians move their arms and legs in a pendulum movement during walking or running, the radar responses from legs and arms have a different Doppler frequency shift than that of the more static torso. It is possible to use radar to do more than merely detect objects; it can also classify them. Furthermore, the walking and running movements can also be classified because the legs' and arms' motion and velocity differ accordingly, and they can even distinguish a walking pedestrian from a runner based only on the different pendulum movements of arms and legs. This method also has applications for separating the cyclist's body movement from that of pedestrians.

However, this is unnecessary in ordinary cases because the bicycle velocity is superior to that of a running pedestrian, which is superior to walking. However, this method can also have applications when the cyclist reduces velocity in the crosswalk.

Radar is the core technology used among the various sensors for the ACC system. Radar is the device equipped on a vehicle to measure the distance from the front car or obstacle and relative velocity using Doppler frequency shift between transmission wave and reception wave [37]. Even simple things like the colocation of the radar in a car can have consequences because of the vertical field of view and azimuth field of view. A proper place for good detection is if the radar is placed over 60cm from the ground with a balanced horizontal level. Some sensor errors emerge from improper placement of sensors on vehicle parts or the lack of security against physical damage. As such, it is imperative to be aware of placement in the vehicle because the potion and the data collected by this are directly dependent on the sensor positions and angle of view.

Other techniques combine using an infrared camera that can detect heat and LiDAR. These are a combination design to combat the weakness. Some atmospheric particles like rain, fog, and dust can reflect the LiDAR laser, leading to malfunctions and wrong assessment of the environment. The radar is also prone to malfunctions, as previously mentioned in scenarios with irregular metal objects with solid reflections [38]. The infrared camera detects obstacles when it is snowy and rainy, in bad weather conditions, with many particles in the air; with these conditions, it is given more importance to the thermal infrared, which can easily detect vehicles and people in cold weather. Otherwise, the weight of LiDAR input gradually increases over the data collected by the infrared camera in hot weather because, in warmer climates, there are fewer particles (except sand storms) and an increase in overall temperature, especially on roads. A couple of environmental sensors can determine the type of day and what weights to use to regulate the input values of sensors and their importance to the overall system, and an IA algorithm can be trained to handle with a higher precision the weight to give to each sensor according with meteorologic data.

Relative to using cameras as vehicle sensors, many systems using AI allow image or video recognition. However, on the road, some specifications need to be addressed. An example is a multi-sensor platform, which supports using a co-aligned RGB/thermal camera and a related calibration technique. Most large-scale datasets are mainly based on RGB-based images. They thus are only feasible in will-lit conditions as opposed to ill-lit environments such as dawn, sunrises, sunset, and nighttime when the RGB has significant variations from standard data from common datasets [39]. It is possible to reflect the visible spectrum (RGB) and transmit the thermal scope using a unique device called a Beam Splitter, which appears in Figure 2.8. The focus was on the improvement and calibration of Beam Splitter, the fusion of RGB, which allows a better identification and classification of images in the video with the thermal photograph, allows a better understanding of object depth because thermal photos depend on the omitted energy rate from objects, not the visible colors (this has some limitations on hazard days, where thermal variations are irrelevant). Combining both methods can recognize an object's depth and actual form even when using various optical illusions that can commonly fully affect the RGB cameras and even the human eye to a false sense of reality. This advantage is because thermal vision primarily aims to complement the spatial awareness usually lacking in a traditional RGB system. This characteristic makes this approach more complex and much more challenging to calibrate. However, the results are extraordinary, returning to the principle that a reasonable perception system must use multiple collaborative sensory systems, thus mitigating their failures.

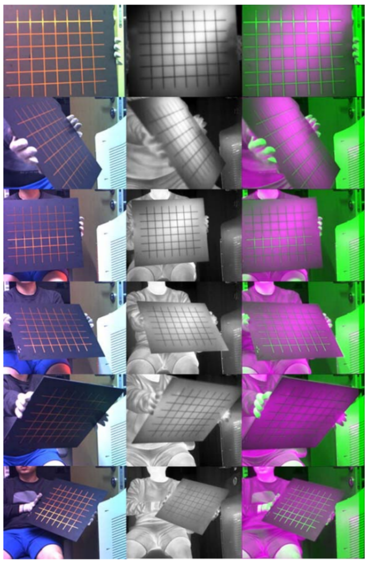

Figure 2.8: (left) RGB images, (middle) thermal images, and (right) fused images

There was mention of what types of values were collected by the various sensors, but how do they relate to different safety indicators, and how do these indicators relate to the car-following situations? In the past, other authors have focused on various safety indicators to analyze car-following problems. For example, time gap and gap distance are variables used in metrics like TTC, time headway, safety margin, and combinations of two or three, such as a union between TTC and time headway. Data such as relative angle to nearby objects, gap distance (distance from the ego vehicle's front bumper to the lead vehicle's back bumper), and relative speed to the lead vehicle arrived via the millimeter-wave radar. After being collected, all this data was processed, analyzed, and grouped, and only then will the Planning process use it. In some instances, drivers are mainly at a time gap of fewer than four seconds or a gap distance of fewer than 80 meters when in car-following situations. These time gap and gap distance values could also be used as a threshold in defining and identifying car-following events [40] but to analyze and start to standardize this behavior in test cases. Furthermore, studies showed that the spread of gap distance was found, as imagined, to be bigger at high speed compared to that at low. Similarly, the disparity of gap distance is more prominent on expressways than highways, with a massive gap between this and national roads.

Finally, it is essential to mention the third stage of any AEB system relative to execution. In this case, there is a necessity for brakes that manage to perform the decisions made by the controller (MPC) with the most incredible precision and accuracy mechanically possible. Some AEBs require a specific amount of force to apply to brakes, so the development of the Brake-by-wire (BBW) system is drawing significant attention and possibly becoming the eventual solution for the brake system used in high-level futuristic autonomous vehicles. This approach features an active braking function, and this mechanism uses a regenerative braking control strategy consisting of two sections, including braking forces distribution (BFD) and hydraulic pressure regulation. When the driver's braking force is not heavy, and the braking intensity of the vehicle is relatively low, not more than 0.1g, the sole front wheel regenerative braking torque is applied to the vehicle's braking actuation. As the braking demand increases, with the driver pressing the brake pedal, the rear wheel frictional braking forces start to add a fixed ratio to the front wheel braking forces (allowing the distribution of the pressure). If the braking intensity reaches 0.5g, the anti-lock brake system function is immediately activated, and the BFD gets close to the ideal BFD curve; better braking performance is obtained in emergency brake due to the perfect BFD curve [41]. By defining the best way to distribute the forces associated with the braking execution and preparing measures to achieve that ideal balance between rear and front wheel braking force, it is possible to achieve a more efficient way to brake with a much higher percentage of precision and accuracy.

2.7 Verification, Validation, and Testing

All the cases presented so far constantly showed the necessity of a virtual use of tests that allow the validation and verification of software. This necessity is mainly due to the risk of injury or material damage that real-world tests can provoke without due preparation because accidents or collisions during verifying, validating, and testing AV components can be monetarily expensive and can put the physical safety of personnel at risk. The risk taken is exceptionally high during the tests, especially in edge cases. The equipment and personnel costs are too high to afford all the necessary tests to V&V an AEB system. V Model, Figure 2.9 is already a standard, reliable system development process. This cascade process allows the division of all development processes into four stages. We can create a different group test for every step, giving great independence and flexibility between stages. Firstly, Model-in-the-Loop (MiL) is for rapid algorithm construction. This stage uses typically unitary test cases relative to the lowest software level, and they are simulations close to unit design. Secondly, the central focus of this study is Software-in-the-Loop (SiL), which is used for software and firmware checking, determining whether testing of software segments independently or the entire software corresponds to the expected behavior. This method allows the detection of bugs and errors in the software before using expensive hardware. Thirtly is Hardware-in-the-Loop (HiL), primarily actuators that will be added in the loop of simulation, figuring out whether there are issues in the hardware response and the relationship/response to software, finally, the actual car test and assessment [42]. Furthermore, when designing a test scenario, it is relevant to define several elements properly. The character of the test scenario could be a vehicle, pedestrian, animal, or any traffic object. The character's behavior may vary between acceleration, turn, curve, braking, or other maneuvers. It is also essential to define road characteristics like if it is a crossroads, the number of lanes or downside roads, and pre-define street making like size, color, and length. Furthermore, we have traffic signs, lights, and shadows, and the last and critical weather can be a significant factor in many ADAS failures, as already shown.

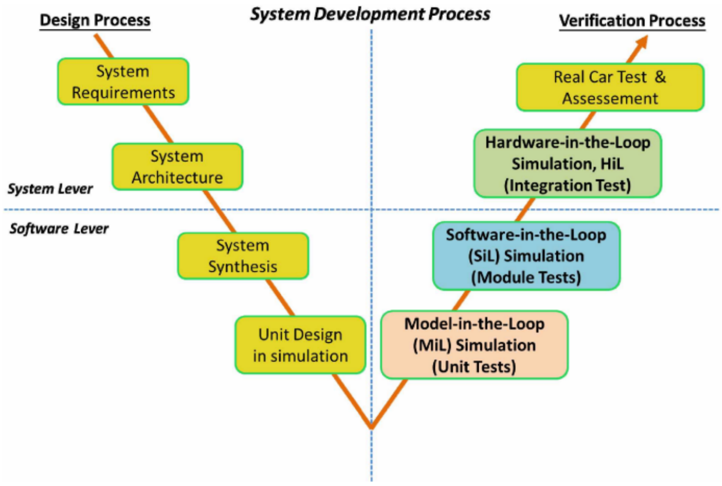

Figure 2.9: V Model Diagram for System Development Process

In the U.S., there were 5.7 million police-reported motor vehicle crashes and 30057 fatal crashes in 2013, while the vehicles traveled 2.99 trillion miles. It is possible to mitigate his problem through the global use of software testing, which would allow the general expansion of the production and improvement of AEB. Some proposed accelerated evaluation approaches exist for AVs generating motions of the other vehicles and MIOs to accelerate the verification of AVs in simulations and controlled experiments. The cut-in scenarios are developed based on skewed statistics of collected human driver behaviors, which generate risky testing scenarios while preserving the statistical information so that the safety benefits of AVs in nonaccelerated cases can be accurately estimated [43]. In this article, the key to accelerated evaluation is to skew the Monte Carlo samples' statistics while maintaining statistical precision and accuracy. Compared with other non-altered Monte Carlo, this method achieves higher performance because the skewed act allows the sampling process to reach more test cases of interest. Meanwhile, cross-entropy enables the search for the optimal skewing parameters recursively. With accelerated simulations, only driving for 1000 miles will expose the AV and generate many challenging scenarios that can take about 20 million miles of real-world driving. This step drastically reduces development and validation time for AVs. It dramatically increases the variety of scenarios designed for testing this but also enables generating a higher gamma of viable test cases, consuming less time and money possible.

However, the previously used method uses Monte Carlo to generate many test cases with fewer traveled distances on a minor one-driving experience. The following process pretends to combat a different aspect when presented with a great dataset. This technique will try to correctly choose the most likely test cases that can provoke a car accident. This AI approach categorizes the test scenarios according to collision type, location, time, weather, and road type, resulting in 36 distinguished categories. Then, select from each cluster whether, among the characteristics occurring there, one existing case that is less frequent in the identified scenarios. These scenarios will likely be located in boundary areas between two clusters, giving a more balanced distribution of tests and simultaneously trying to cover less regular collision scenarios. Next, these test cases are aggregated to be representative of each cluster, allowing a few characteristic scenarios, which, on the one hand, cover the entire performance space and, on the other hand, also consider different collision types and, as far as possible all sectors of the road network [44]. This method can determine the optimal parameterization for the test cases to implement.

Thoroughly validating and verifying automated or autonomous driving functions is inevitable for meeting quality criteria for safety-critical systems. Being able to generate and compare test cases is crucial to selecting the best model to implement and using the conjugation of combinatorial testing and search-based testing techniques. It's possible to create a virtual testbed to carry out many tests faster than in real time. Also, it reduces costs by using a genetic algorithm to evolve an initial population of candidate solutions (individuals), holding a finite set of parameters (chromosomes) that each comprises specific parameter values (genes), like several genetic artificial intelligence systems. The process of generating new test scenarios using genetic algorithms is what sets this technique apart from others. Here, the optimization problem attributed to a higher score for lower TTC, thus maximizing the number of risk scenarios. After the AI algorithm generates new and better individuals (test cases), the focus shifts to search-based testing to come up with concrete values of the parameters in a particular scenario, and finally, the combinatorial testing approach will find critical interactions between parameters, forming a specific method [45]. Then, it just needed to test his veracity using different AEB techniques, where they could use the test cases capable of generating accidents. This shows the fastest way to create multiple tests for various AEB approaches and that artificially generated scenarios can demonstrate the need for rectifications and upgrades in some AEBs in a few minutes, fully automated.

One problem that provokes AEB failure is the need for more requirements definition, a crucial part of any VVT process. In automotive software systems, the threats to safety include not only external factors from outside the automobile but also internal factors such as a driver's erroneous operation. It is possible to use a Bayesian network to quantitatively evaluate the effect of safety requirements by comparing and correlating the accident probability before and after, applying mitigation use cases based on the initial safety requirements and the safety patterns. These extracted safety requirements from real-case scenarios that result in accidents will define the new standard for developing the system [46]. This automatic early requirements extraction is an essential first step in the V model because it will serve as the starting point and base for all the cascade processes.

There are two main challenges in the validation of algorithms for autonomous driving. First, the complexity and variety of scenarios the vehicles can face are more extensive than in Advanced Driver Assistance Systems (ADAS), and second, there is a necessity to consider the constant unpredictability possibility of interaction between multiple systems. For that, it is possible to implement Statistical Model Checking (SMC), which is a method to provide an intermediate between testing and exhaustive verification by relying on statistics [47]. It is important to have well-defined two components: intention and expectation. The first is the drivers' maneuvers, whereas the second is what the system should do regarding the current situation and traffic rules. The definitions of these two are fundamental prerequisites in any test case, and the comparison between them (what is intended and what is expected) is the best global indicator to measure the system's performance under a loop in the V-Model. Every single test case should have well-defined desired behavior before running the test and drawing the intention of the maneuver. With both intended and expected maneuvers defined, SMC is used to evaluate statistically the difference between both processes. If showing that the description between the intended action and the desired behavior is too large, the model needs to be reconsidered.

Also, according to the European New Car Assessment Programme (EuroNCAP) standard, AEB is divided into three working conditions: AEB City, AEB Inter-Urban, and AEB Pedestrian. For highway scenarios, the AEB finds itself in the Inter-Urban situation. The class of tests is also subdivided into three subclasses of scenarios: Car to Car Rear Stationary (CCRs), Car to Car Rear Moving (CCRm), and Car to Car Rear Braking (CCRb)). In a CCRm scenario, the vehicle avoids a rear collision with another moving vehicle. Usually, in this scenario, the front vehicle is moving at an inferior velocity than the ego vehicle. In the CCRb method, the car under braking begins to decelerate, generally at 2 m/s^2 or more, and the ego vehicle under test needs to avoid the rear-end collision using the brakes or evasive maneuvers. If it realizes a brake, it's necessary to mention that the deceleration value of the ego car needs to be superior to the front vehicle to avoid collision. It's recommended that the initial gap between the ego and braking vehicle is at least 12m [48]. Whoever this is subject to change according to the type of test and behavior under scrutiny. Finally, there will be scenarios where the front object, vehicle or not, is entirely immobile (usually representing a car accident or an object); these cases are the CCRs scenario, where the vehicle avoids relocating stationary cars.

So far, we have analyzed mainly the case of Software-in-the-Loop. However, Hardware-in-the-Loop is also a crucial part of the AEB system, primarily to test sensors and actuator's behavior in dangerous situations and to demonstrate the dangers the passenger may suffer under AEB. For that type of experience, the test dummy is one of the most famous systems of tests in the automobile industry. This full-scale anthropomorphic test device is fundamental during any HiL test. Especially in all first stages of human behavior tests (AEB simulations), particularly out-of-position tests, dummies are used as their primary source of information. Tests proved that with the initial speed of 30 km/h and the peak braking deceleration of the vehicle set to 0.7g, 0.8g, and 0.9g, the head displacement in the third test case was, logically, significantly greater (almost the double) than in the first scenario [49]. It is challenging to avoid forward displacement when volunteers have no subjective support for their arms during the braking phase. The most important factor affecting the removal is that with the increase of the peak braking, the out-of-position displacement will increase. Conversely, increasing the initial speed under the same braking deceleration has little effect, as shown in Figure 2.10.

The testing results from the actual road test are more accurate than those from using dummies. However, it is time-consuming, expensive, dangerous, and susceptible to unpredictable scenarios that can ruin the original objective of the test. Other unpredictability or difficulties associated with the road environment can contaminate the initial test case hypothesis. There are several alternatives to real-world test cases, like implementing a virtual-reality simulation platform for AV testing under the laboratory environment, called a test bench. This test

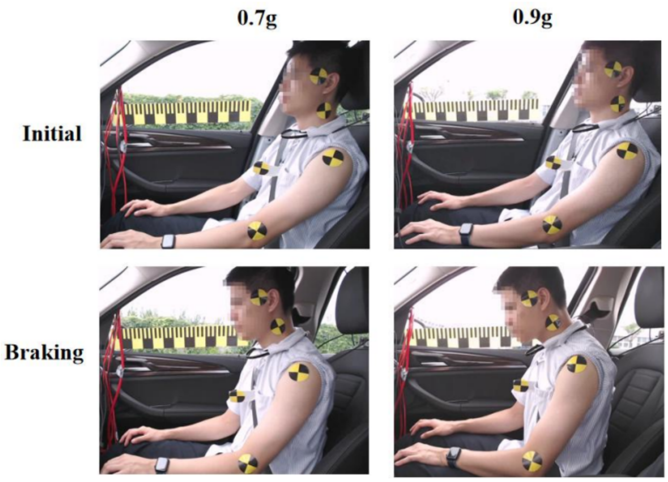

Figure 2.10: Body displacement during a brake action

platform can be accommodated with external factors (road friction coefficient, road unevenness, and road gradient) to simulate various natural road environments. Furthermore, it is also possible to deploy black boxes on sensors, which only interact with the virtual scenes, thereby maximizing the realism of test cases and consequently significantly increasing the fidelity of the test in both HiL and SiL [50]. This approach is an alternative to exposing hardware directly to real-world test scenarios and to design test cases independently of the software that was made. It allows a more controlled procedure for small tests to independent hardware segments while creating an intermediate stage for software cooperation with some hardware testing. Furthermore, there is a tendency to create Driver-in-Loop (DiL), the last step before the real-world tests, which makes platforms based on virtual scenarios and driving simulators (Figure 2.11) like the ones used for aviation tests. Creating this scenario in a loop pretends to test the driver's behavior and the interaction with the vehicle interface, both hardware and software performance. The software can create road environments with obstacles and simulate actual vehicle driving conditions based on a complete vehicle dynamic [51]. Then, according to user performance and opinion, the interface can be optimized to satisfy vehicle dynamics and aesthetics and give an overall natural experience to the passenger. In contrast with the previous study, where the passager was passive and analyzed his reaction and behavior using DiL, the passager had access to equipment like Logitech G29, which acts like an independent wheel and pedals that resist all driver maneuvers. This instrument makes him more of an active agent, enabling the driver to adjust the vehicle's trajectory in real time by judging the surrounding environment. The driver can interact with the virtual scene and generate a more realistic test case. These are a more focused ADAS test type because the AV system is to be completely independent without requiring human intervention. Still, even AVs have passengers who can try to interfere with the system, maliciously or not intentions. However, once we reach this level of full automation, the DiL will serve as the last barrier or opportunity before pursuing real-world testing.

Are already some DiLs capable of working with scenarios with road friction in the two front wheels. This small improvement provokes a slight yaw rotation of the ego vehicle, where the AV constantly adjusts the longitudinal deceleration and yaw moment. These scenarios are

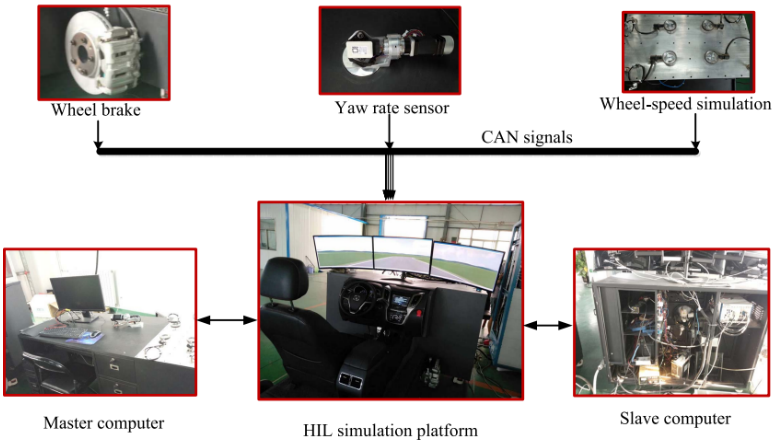

Figure 2.11: Structure of Driver in Loop

helpful for very rough roads or heavy environment conditions where there is a significant discrepancy between the friction on the left and right wheel. This method allows the HiL to test with a real wheel brake, yam rate sensor, and wheel speed sensor. The DiL consists of steering wheel hardware, wheel brake cylinders, standard sensors on board such as the yaw rate sensor, wheel-speed hardware simulator, a NI PXI computer, and a master computer [52]. These tests permit the V&V of both HiL combined with software for lateral stability and collision avoidance and risk assessment tests, allowing a more versatile way to realize road tests. Testing everything, a prior human trial, in a simple and small room, reducing cost, time, and human risk, and adapting quickly to other situations.

Chapter 3

Study of the test bench for Highway Path Planner with Autonomous Emergency Braking

3.1 AEB Algorithm Construction

The automobile industry estimates that human-driven cars will no longer exist, with estimates pointing to 2060 or 2080. Human drivers will most likely be forbidden on highways for safety and efficiency, as the same happens today relative to riding horses or bicycles on high-speed roads. Another possibility is the existence of multiple lanes for autonomous vehicles and few lanes for drivers, aiming to accomplish the efficiency necessary to optimize AVs and also maintaining some alternatives for more nostalgic drivers or emergencies. In most accidents, serious rear-end collisions (which create the most severe injuries), the driver in the following vehicle rarely takes complete braking. Sometimes, it does not even press the brake pedal, which can occur for the most diverse reasons already addressed in the previous chapter. However, this mainly proves the driver's need for more danger awareness. These human inabilities create a necessity to develop more accurate risk assessments, timely warnings, and communication of every type, but how can we correctly define the overall performance of the AEB system? The core rule focuses on three vital factors: efficiency, accuracy, and passenger comfort [53]. Passenger comfort is generally defined as a max value of deceleration or jerk (how abruptly and fast the forces over the human body change). In some urgent situations, ensuring safety and comfort while preventing a car crash is challenging as, in some cases, to prevent the accident, the deceleration and jerk exceed the comfortable value to ensure precision and accuracy, making the passenger comfort a very subjective factor and some times, even negligent one.

The test cases realized in this approach will base themselves on samples from a combinatorial approach for some vehicle attributes (commonly in values ranging in velocity, start position, and road lane). However, to increase diversity from prior works, some diversity will be added by allowing some vehicles to change lanes and to stop (simulate braking or crashing of other cars). Nevertheless, all the scenarios constructed obey the standard AEB ontology based on EuroNCAP scenarios [54], with some simplification implemented, especially considering the significant offset of the variable being null (except in lane changing or stopping). Other NCAP specifications and standards contribute to defining vehicle dynamics, road characteristics, and other physical and metric measurements.

A typical AEB working procedure mainly contains two stages: warning the driver system and partial or full braking force when there is a possibility of collision. A full braking force is applied if the collision risk rises over a threshold and passes a determined safety barrier. The standard peak braking coefficient could reach high levels between 0.6g and 1.0g. Every value under that threshold is considered desired but achievable with significantly compromising

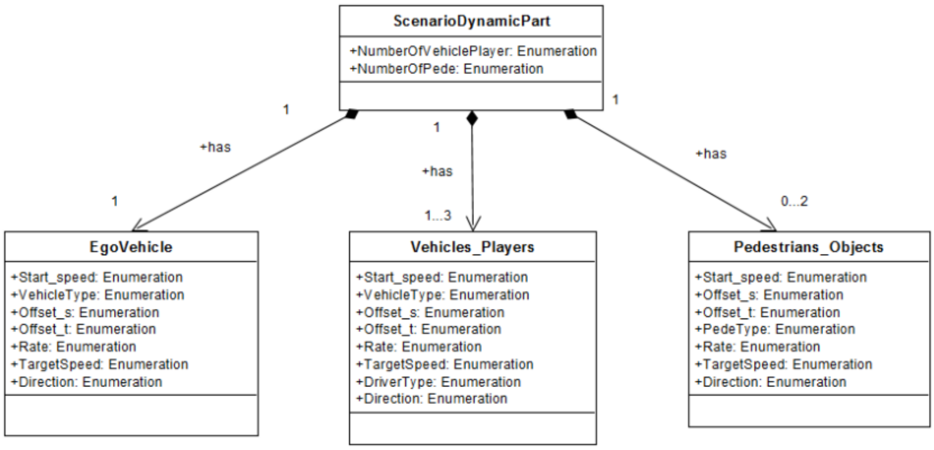

Figure 3.1: Constructed AEB ontology based on EuroNCAP scenarios using UML

path planning. The risk assessment combines Inevitable Collision State (ICS) and TTC. ICS is a state defined by the following permission, "no matter what the future trajectory of the system is, considering both vehicle and obstacle dynamics and motion restrictions (maximum deceleration or jerk), a collision will occur." In other words, state A will provoke a closing on state B if it keeps the same status for a time, and there is no alternative state C, projected by Path Planning of the system, that is collision-free. When Path Planning cannot find a route without collision, it is necessary to find a Minimum Collision Avoidance Distance, which represents the minimum distance to the obstacles under the best possible input trajectory [55]. This time and distance threshold can be achieved using the TTC index, with a determined maximum deceleration under the current relative distance and velocity, which is typically used to evaluate when the braking process needs to be activated.

However, road conditions are more complicated in reality, and many factors, such as tire properties and road friction properties, can influence the brake distance. The same happens to variables that can condition these two, like weather. For example, rain will directly affect road friction and TTC. However, we can simplify this project by mitigating these conditions by adding a margin of error to the TTC calculations. The most crucial factor in TTC is maximum applied deceleration, a variable that defines the entire AEB system. If the value is higher than expected, it will increase the jerk and the risk of injury and discomfort from the passenger (more abrupt brakes). On the other hand, if it is too low, it can not be enough to prevent a crash because it will require more time and space to brake, and in almost all car crashes on highway scenarios, time and distance are metrics that will define a loose of life or not. This variable can be affected by road conditions, wind friction, tire properties, road inclination, gravity, and more, making it almost impossible to have a standard dynamic behavior for vehicles that abrange all scenarios [56]. The inability to correctly have maximum deceleration that satisfies all scenarios is paramount for dividing TTC into various thresholds or sub-TTC. The popular expression "better safe than sorry" is the almighty rule when discussing the mechanism of preventing the loss of life. Therefore, adding an error gap to the deceleration value in every braking phase is very natural. However, all these safety measures added to guarantee the passenger's safety will irrefutably harm the self-driving system's performance, adding a particular type of "hesitation" of the AV while performing the best course of action and consequently making the vehicle slower and computational cost higher. There is a better explanation of this ratio and performance when discussing the integration of the AEB subsystem in the AV system.

Other alternative procedures or indexes exit, for example, Braking Threat Number (BTN) and Steering Threat Number (STN), where TTC mainly indicates time according to velocity, distance, and acceleration, the BTN and STN measure the capability to avoid collision by braking and steering respectively [57]. However, any evasive maneuver or braking procedure is activated after the TTC considers that there is a risk of collisions (Figure 3.2), where they consider if it is still possible to steer the vehicle out of danger, using the path planning system to evaluate alternative trajectories. If not possible, steer out of danger, or if it has already reached the time to start breaking to avoid collision regardless of existing alternative paths, then the vehicle will brake. In other words, when a vehicle reaches a safe distance of any of the three braking stages, the vehicle will brake with the deceleration level determined for that specific stage. However, it always allows the path planning system to determine an alternative trajectory if that is available. This strategy means there is a hierarchy where the braking system is always activated when reaching a danger threshold. However, it will always try to find an alternative trajectory in any of the three stages instead of fully immobilizing the vehicle in the middle of the highway. However, suppose there is no alternative path, or the vehicle cannot apply it on time to avoid a collision (even a partial one). In this case, the vehicle will permanently immobilize itself before colliding with the front vehicle. However, it is essential to mention that the Path Planning system automatically discards high-risk paths with stiff steering, sideways whiplash, high neck movement, and collision with road barriers or walls. The smooth maneuvers are guaranteed because the planning will only create trajectories using the fifth-order polynomial function, enabling the generation of minimum jerk trajectory [58] where $s(t)$ is for the longitudinal or lateral distance. It's also possible determine the velocity $\dot{s}(t)$, the acceleration $\ddot{s}(t)$, and the jerk $\dddot{s}(t)$.

$$s(t) = a_6 t^5 + a_5 t^4 + a_4 t^3 + a_3 t^2 + a_2 t + a_1 \quad (3.1)$$

Should the controller choose the lesser of two evils when a traffic accident is imminent? There are three major types of collisions: avoidable, ordinary unavoidable, and pedestrian inevitable [59]. In this project, in all the cases tested, the collision was avoidable if the vehicle took the proper actions to prevent a crash in cases where, no matter the actions, the collision is unavoidable (cases where there is no sufficient time or space to do any course of action) are automatically discarded. These situations are a harsh reality, but in some cases, no action can be taken to avoid a collision, only trying to reduce the damage to the minimum possible. The deceleration will always be necessary, independent of the scenario, with a good and capable AEB that can ensure the capacity to reduce the damage. However, as stated, there are cases where the pedestrian can be invisible to sensors before being detected, and then the distance can be less than the minimum safe braking distance, and injuries are inevitable, although the supreme rule to mitigate maximum damage possible is applied.

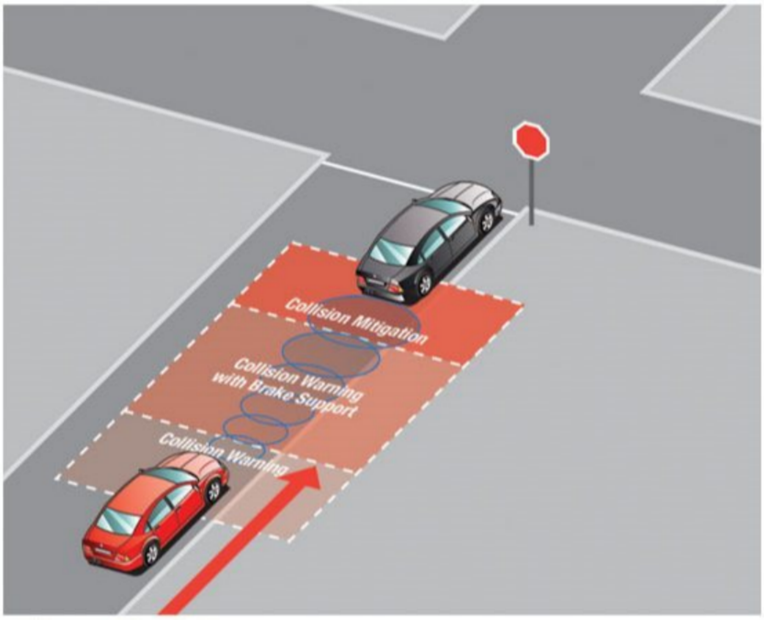

Figure 3.2: Division of risk TTC in phases

The braking process adopted for this AV is a multi-stage AEB divided into three stages that will be activated sequentially as a cascade process as the danger of collision increases. The first stage is the warning phase, which is responsible for evaluating risk assessment and starting the warning light to inform the passenger when a collision possibility exists. Secondly, we have the first partial braking force phase (t1). In this stage, the warning light changes to an intense color (for example, orange), and the second partial braking force phase begins, with a soft increase of the deceleration force. The application of this middle stage occurs for two reasons: an inferior velocity allows some time and distance for Path Planning to find an alternative path and the MPC to adapt to the new circumstances. Second, this preventive deceleration in this phase will reduce the deceleration threshold between the previous and the next and final stage (full brake), reducing jerk and increasing passager comfort. The soft changes between deceleration values will inevitably reduce the body and neck displacement of the passenger during this emergency maneuver, thus reducing discomfort and the risk of injury, namely whiplash neck, concussions, or even seat belt injuries. Finally, the third stage is the complete braking phase (t3), an abrupt deceleration force value, shifting automatically to the maximum deceleration pre-determined. The second phase can mitigate the aggressive shift, significantly reducing jerk value. Some studies appoint a TTC of 1.5s and a max deceleration of 0.9 g as optimum, in two-stage AEB with TTC = 1.0s at the first stage and TTC = 0.6s at the second stage being optimum. These are generic values used in various situations, varying from urban roads to crossroads. The deceleration values need to be higher because, in highway situations, they need to directly correspond to higher velocities involved, which makes ego velocity and MIOs (Most Important Objects) higher than in typical conditions. Hence, the deceleration force in every phase must correspond to this velocity increase. A multi-objective optimization problem using the NSGA-II algorithm uses a TTC of 1.737s, a deceleration of 7.428 m/s^2 [60]. The standard evaluation considers maximum decelerations of 3 m/s^2 and 9 $m/s2$, which correspond to either too smooth or aggressive shifts of deceleration force. Generally, in the three stages of AEB, the brake pressure of the first stage is at the fifty percentile (t1 = 2.5 $m/s2$), at the second stage, pressure is at the seventy-five percentile (t2 = 4.5 m/s^2), and the third stage full brake is at the ninety-five percentile (t3 = 5.5 m/s^2), we can see this idea represented in the Figure 3.3. The AEB system present in this work is a variation of this last-mentioned method. However, instead of using this fixed deceleration value in each phase, it will use the same process but with values to represent the scenarios inserted accurately.

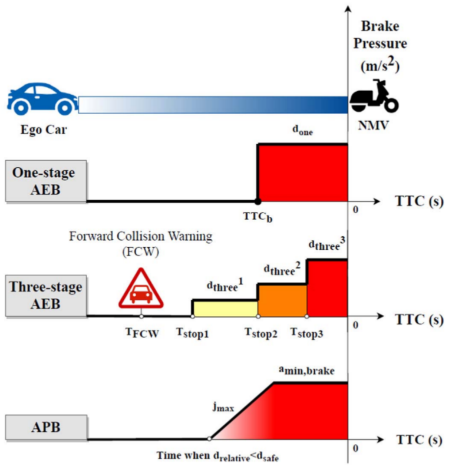

Figure 3.3: Braking profiles of one-stage AEB, three-stage AEB and Constant models

3.2 Matlab Simulink and Vehicle Dynamics

It will present the basic information of algorithms, systems, blocks, and tools Matlab Simulink offers that helped develop this project. The primary tool used is the Vehicle Dynamics Block of Simulink Matlab. One crucial fixture is the necessity to change between coordinate standards, using three cartesian standards like SAE J670 and ISO 8855. The vehicle dynamics modeling follows the SAE J670 standard defining an Earth-fixed coordinate system in which axes are fixed in an inertial reference frame where the angular velocity and linear and angular acceleration are zero. Vehicle coordinate systems are located on the ego vehicle and placed on the ground, below the rear axle middle. The Path Planning and simulations follow the ISO 8855 standard, which defines the world coordinate system, in which the axes are according to the traditional and commonly used coordinate system, the difference between the axes Y and Z of the vehicle coordinate and world coordinate system are presented in the Figure 3.4. The world coordinate system is also used in the Frenet frames [61] to describe better the movement of a particular solid object moving in a tridimensional space.
Another essential tool in this project is a 3DOF model, widely used for simulation purposes because it allows for studying various vehicle behaviors such as velocity, acceleration, braking, and steering. The Vehicle Body 3DOF block[1] from the Vehicle Dynamics Blockset of

[1]https://www.mathworks.com/help/vdynblks/ref/vehiclebody3dof.html

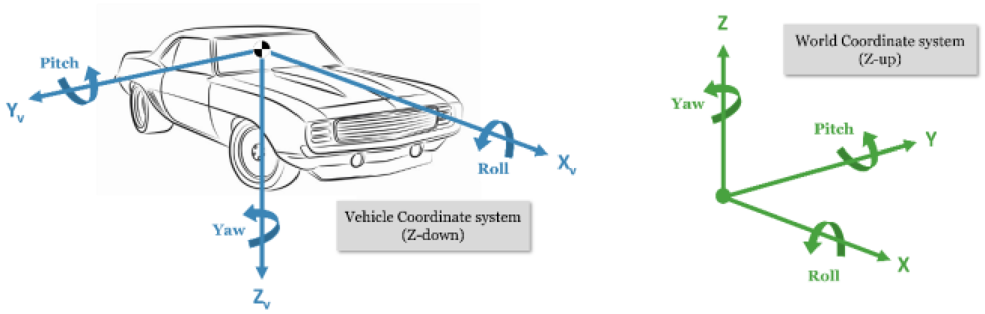

Figure 3.4: Axis systems for the vehicle coordinate system and the world coordinate system.

Matlab[2] was used to design and model the Ego vehicle. Solid object dynamics models are classified based on the number of degrees of freedom they present in their movement. In norm, road vehicles use 3DOF, which has movement freedom characterized by three important vectors. Horizontal displacement, where it can move forward or back. Lateral displacement, where it can move right or left, and yam rotation describes the object's rotation over the z-axis. Furthermore, if we try to describe the movement of a plane or a drone, we need to use 6DOF, where three types of movement are added to the previous. Roll describes the object's rotation over the x-axis. Pitch describes the object's rotation over the y-axis and vertical displacement, where it can move up or down. The 6DOF is mainly used for aeronautic projects. In contrast, the 3DOF model is widely used for vehicle simulation because it describes an object's motion over a planner surface. It includes all important variables associated with car driving, like distance, velocity, acceleration, braking, and steering. These allow us to determine the vehicle's trajectory and study the path described by the vehicle. Furthermore, simplifications allow us to represent the vehicle's dynamics using a two-wheeled vehicle model, known as a single-track or bicycle model (shown in Figure 3.5). Even though this model is simplified, it can accurately represent the vehicle's lateral dynamics, which are still important for this study as it can change trajectories to avoid collisions.

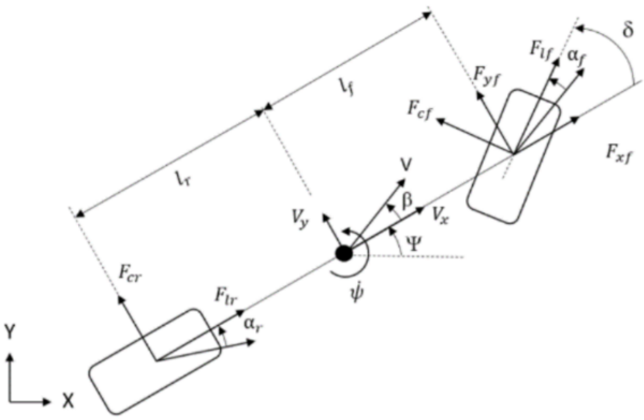

Figure 3.5: Bicycle Model

[2]https://www.mathworks.com/products/vehicle-dynamics.html

Autonomous Emergency Braking System

For the elaboration of this work, five crucial assumptions are needed:

- Assumption 1: The ego vehicle travels in a forward direction with a constant velocity, with a longitudinal velocity equal to the initial longitudinal velocity ($V0$) set at the beginning of the scenario. In contrast, the lateral velocity, longitudinal acceleration, angular velocity, and steering angle are all assumed to be zero at the start of the simulation.
- Assumption 2: The ego vehicle is modeled as a bicycle with four wheels, in which the left and right axles are aggregated into a single axle with two wheels (representing front and back wheels). The tractive force only comes from the front wheels. The vehicle's mass concentrates at the center of gravity (CG), located at the midpoint of the segment connecting the two wheels.
- Assumption 3: The vehicle has 3 Degrees of Freedom (DOF): two displacements in the plane (longitudinal and lateral) and one rotation around an axis perpendicular to that plane (yaw rotation).
- Assumption 4: This model neglected all suspension movement, road inclination, and aerodynamic influences with default or standard values for all parameters related to these characteristics.
- Assumption 5: The other vehicles in the plane are called target vehicles or Most Important Objects (MIOs). These vehicles travel at a constant speed, with some exceptions for some test categories, on a fixed reference path defined by a set of waypoints. Usually, MIOs have two waypoints, initial and final position, but more waypoints are added to represent a lane change.

As mentioned in planar motion, the coordinates to describe the vehicle motion are X, Y, and Ψ, where (X, Y) represent the inertial coordinates of the location of the center of gravity of the vehicle, while Ψ (yaw angle) indicates the orientation of the vehicle-fixed frame about the Earth-fixed z-axis. The vector V is the velocity at the CG of the vehicle and makes a slip angle (β) with the longitudinal axis of the car. With the car in an absolute inertial frame, the motion equations are:

$$\chi = Vcos(\Psi + \beta) \quad (3.2)$$

$$\Upsilon = Vsin(\Psi + \beta) \quad (3.3)$$

$$\Psi = \frac{Vcos(\beta)}{lf + lr} \tan(\delta) \quad (3.4)$$

Where χ and Υ are the longitudinal and lateral velocities, Ψ (yaw rate) is the vehicle angular velocity about the vehicle z-axis, δ is the steering angle, lf, and lr are the longitudinal distance from the CG to front and rear wheel, respectively.

For all the following equations in this section, we need both specific formulas to be applied to front-wheel f and another for rear-wheel r because both have different behaviors and functions of displacement in space.

All object displacement in space is subject to the laws of Newton, especially the second law, which establishes the correlation between force, mass, and acceleration. The bicycle model is no exception, wherein in a frame of time, it is possible to extract all adding moments (Mx, My, Mz) and forces (F_x, F_y, F_z). These moments and forces pulled around the vertical axis at the CG represent all the vehicle's lateral motion and rotational dynamics according to three coordinates and degrees of freedom. In other words, to extract the acceleration (lateral and longitudinal), we need to use Newton's equations for translational motion in χ and Υ. This will allow us to obtain the vehicle's longitudinal and lateral motion equations. On the other hand, the z-axis uses Euler's equations to describe the angular motion. All these equations aim to describe the object's displacement in the lateral motion. Still, it is also vital to have the longitudinal forces that will influence the velocity, acceleration, and braking. This last will be incredibly essential to TTC calculation from the AEB system.

$$m\chi = m\Upsilon\Psi + F_{xf} + F_{xr} + F_{x\ ext} \quad (3.5)$$

$$m\Upsilon = m\Upsilon\chi + F_{yf} + F_{yr} + F_{y\ ext} \quad (3.6)$$

$$I_{zz}\Psi = l_f F_{yf} - l_r F_{yr} + M_{z\ ext} \quad (3.7)$$

Where χ and Υ represent longitudinal and lateral accelerations, Ψ is angular acceleration and m is the vehicle's mass. F_{xf} and F_{xr} are the longitudinal forces applied to the front and rear wheels respectively. In contrast, F_{yf} and F_{yr} are the lateral forces of the front and rear wheels. $F_{x\ ext}$ and $F_{y\ ext}$ are the external forces applied to vehicle CG. I_{zz} is the yaw polar inertia. Finally, $M_{z\ ext}$ is the external moment of the vehicle CG about the vehicle z-axis.

Longitudinal and lateral forces applied to the front and rear wheels are calculated using longitudinal and lateral tire forces and the steering angle. We can see that the force of the rear wheels is logically simply the traction force of tires for longitudinal displacement F_l, while the lateral displacement of the rear wheel is equal to the cornering force of tires F_c. This is because only the front wheel can rotate when steering. On the other hand, the front wheel forces will be determined by the difference between the angles of traction force and cornering force.

$$F_{xf} = F_{lf}\cos(\delta) - F_{cf}\sin(\delta) \quad (3.8)$$

$$F_{yf} = F_{lf}\sin(\delta) - F_{cf}\cos(\delta) \quad (3.9)$$

$$F_{xr} = F_{lr} \quad (3.10)$$

$$F_{yr} = F_{cr} \quad (3.11)$$

Furthermore, traction directly results from the longitudinal force generated by tiers during acceleration (negative or positive). In contrast, the tire slip angles α and linear cornering stiffness C_y influence the tire's lateral force generated while cornering. It is also necessary to take into account the wheel friction coefficient μ and the normal force applied to the vehicle in the z-axis F_z over the nominal normal force applied to axles along the vehicle z-axis F_{znom}. These last two represent the forces that impact the tire that originate from the yam rotation.

$$F_{lf} = F_{xf\ input} \quad (3.12)$$

$$F_{lr} = F_{xr\ input} \quad (3.13)$$

$$F_c f = -C_{yf}\alpha_f\mu_f\frac{F_{zf}}{F_{znom}} \quad (3.14)$$

$$F_{cr} = -C_{yr}\alpha_r\mu_r\frac{F_{zr}}{F_{znom}} \quad (3.15)$$

To maintain pitch and roll equilibrium, the normal forces obtained by the following equations are divided by the nominal normal load to vary the effective friction parameters during vehicle weight and load transfer. All forces or moments that act over the vehicle CG about the vehicle axis need to be considered, including external forces.

$$(l_f + l_r)F_{zf} = l_r mg - (\chi - \Upsilon\Psi)mh + hF_{xext} + l_r F_{z\ ext} - M_{y\ ext} \quad (3.16)$$

$$(l_f + l_r)F_{zr} = l_f mg - (\chi - \Upsilon\Psi)mh + hF_{xext} + l_r F_{z\ ext} - M_{y\ ext} \quad (3.17)$$

Where h is the height of vehicle CG above the axle plane, $F_z\ ext$ is the external force applied to vehicle CG along the vehicle z-axis, $M_{y\ ext}$ is the external moment of the vehicle CG about the vehicle y-axis.

Finally, the slip angle of the tires (front and rear) represents the angle between the wheel velocity and the direction of the wheel itself.

$$\alpha_f = \arctan \frac{\Upsilon + l_f \Psi}{\chi} - \delta \quad (3.18)$$

$$\alpha_r = \arctan \frac{\Upsilon - l_r \Psi}{\chi} \quad (3.19)$$

Considering the Bicycle Model 3.5, all these equations represent the calculation executed by the Vehicle Body 3DOF block to simulate the actual effect of the inputs on vehicle dynamics, providing the natural reactions to the action directed by the AV. Figure 3.6 represents the block structure in Simulink of the dynamics vehicles where it is possible to detect five main components:

1. Inputs - the inputs are the steering angle and acceleration provided by the Controller Block and fed to the Vehicle Body 3DOF block.

2. Lower Level Controller - this block receives the acceleration value. Then, it converts it into the throttle 3.44 force required to track the desired acceleration, incorporating a lag in tracking the desired acceleration to simulate the imperfect tracking of an actual vehicle.

3. Vehicle Body 3DOF block - Simulink block that recreates the vehicle's behavior according to the inputs it receives from the Controller and the abovementioned equations presented in this section.

4. Coordinate Change - convert from vehicle coordinates used by the 3DOF block, according to the standard SAE J670, to world coordinates, according to ISO 8855 standard, to be used by the Controller Block and Path Planning.

5. Outputs - four outputs give us information about the Ego vehicle: the position of the car in the world, the velocity, the yam angle, and its rate. The other two inputs are Longitudinal velocity, used in Controller and Metric blocks, and Lateral velocity for the Metrics block.

It's important to mention that the Vehicle Dynamics and Controller blocks are closely related to longitudinal velocity. This is because the controller will apply the acceleration needed to the vehicle to accomplish the necessary velocity to accomplish the path determined by the Path Planning. The controller's function is to use the acceleration to cancel the difference between the longitudinal velocity provided by the Vehicle dynamics and the velocity necessary to accomplish the path defined. Both longitudinal and lateral velocities are needed for the Metric Assessment block to determine the TTC, measure the collision risk of the ego vehicle, and display other safety variables to the driver.
Furthermore, some standard and fixed values must be defined in the Vehicle Body 3DOF block. They refer to variables in the equations above that are vital to realizing the simulation, like the vehicle's mass, initial velocity, or initial yam rate. This values are shown in Table 3.1

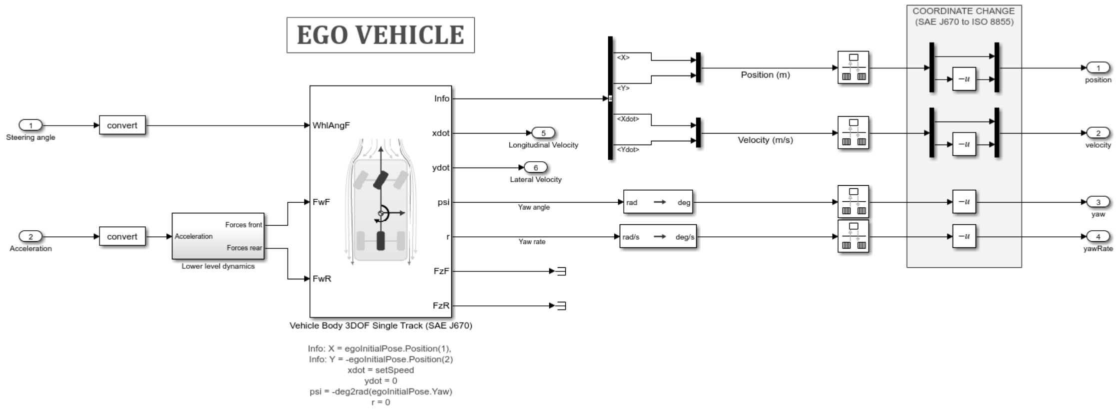

Figure 3.6: Vehicle Dynamics Block

Parameter	Value	Units
I_{zz}	2875	Kgm^2
V_0	0	$\frac{m}{s}$
C_{yf}	19000	$\frac{N}{rad}$
C_{yr}	33000	$\frac{N}{rad}$
Ψ_0	0	$\frac{rad}{s}$
$F_z nom$	5000	N
m	1575	Kg
l_f	1,2	m
l_r	1,6	m
h	0,35	m
α_f	0,1	rad
α_r	0,1	rad

Table 3.1: Ego vehicle variavels

3.3 Highway Lane Change with AEB test bench

The test bench created in the Matlab Simulink must represent a vehicle capable of driving on a highway without collision or going off the road. As shown in Figure 3.7, this test is constituted of six major subsystems:

1. Scenario and Enveirmont - Reads map data from the base workspace and outputs map information about lanes, actors, and path references.

2. Path Planning - Complex system that determines the best references of a path that the vehicle should act on by resolving optimization problems according to three ADAS.

3. Path Following Controller with AEB - Compute optimal control actions while satisfying all requisites stabilized by the Path Planning subsystem and applying the deceleration force (braking) when safety rules are breached.

4. Vehicle Dynamics - Represents the behavior of a rigid object (car) when provided with some stimulus (acceleration and steering).

5. Metrics Assessment - Provide information for a driver dashboard and analyze the collision risk calculations for the AEB system.

6. Visualization - Function that creates a Matlab plot using the inputs from the scenario, environment, and planner systems

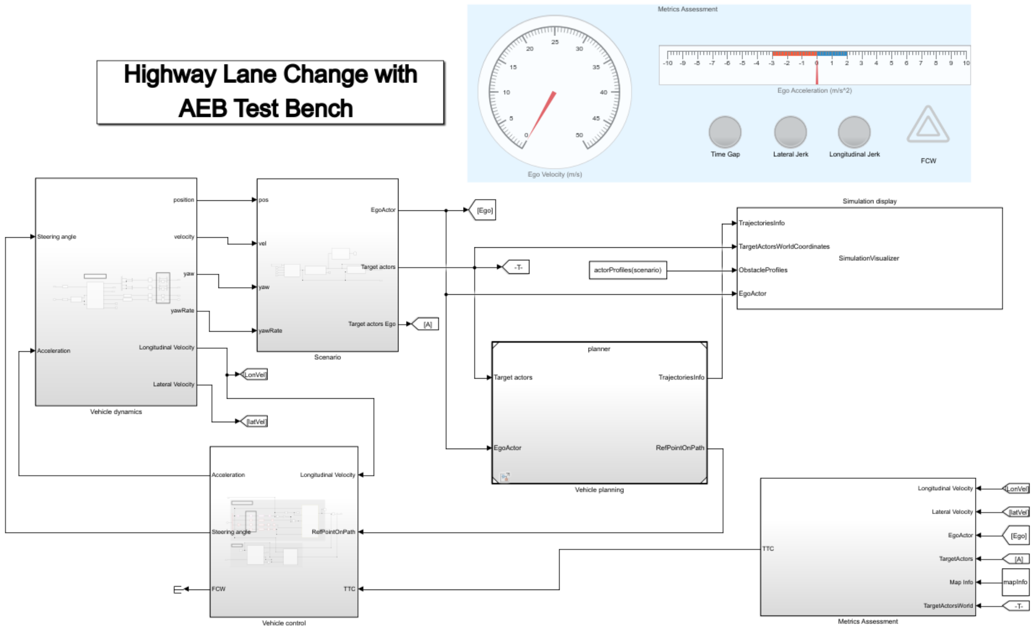

Figure 3.7: Highway Lane Change with AEB test bench

The Scenario and Environment block is divided into four components. The first is the Ego Vehicle block. This block aims to create a Matlab Bus (a data structure used in Simulink), creating a variable representing the primary vehicle. This variable is used in almost all subsystems of the test bench, being the core of the entire system, and it is created using data provided by the Vehicle Dynamic block, whose outputs represent the vehicle's behavior according to Controller actions. Another important block is the collision detection system that stops the simulations if the car goes out of the road map or collides with other target actors (vehicles, barriers...). In other words, it evaluates if the three-dimensional space representing the Ego vehicle penetrates the three-dimensional space of another actor in the scenario, stopping the simulation. Furthermore, the two most essential components are the Scenario Reader block (using ISO 8855), which reads a driving scenario from the workspace. These scenarios are pre-loaded, and their creation will be mentioned in the next chapter. As shown in Figure 3.8, this block will receive the Ego Vehicle Bus to perform a closed-loop simulation of ego in the scenario. It will provide ground truth about Actors and lane boundaries in

Ego vehicle coordinates. Finally, the last block is the Vehicle to World block, which converts Target vehicle positions from vehicle to world coordinates. These coordinates will permit the path planning phase to detect the position of other vehicles in the scenario. This block will create the Ego vehicle as a variable and define the positions of the Target Actor in world coordinates, where these two variables will feed directly to the Path Panning Block.

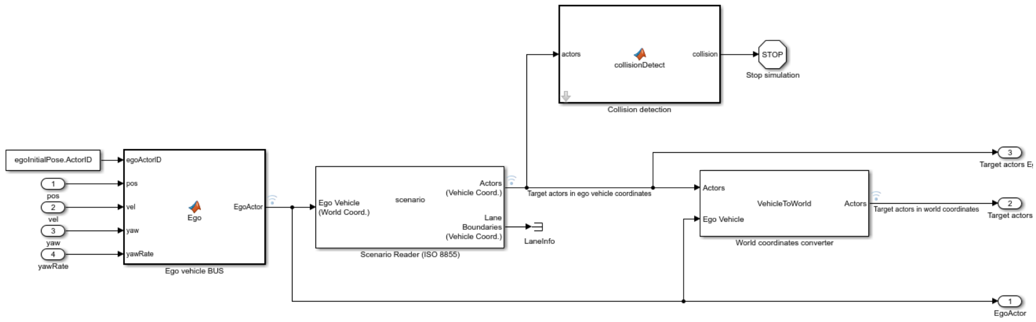

Figure 3.8: Scenario and Environment Block

3.4 Path Planning

Trajectory planning uses planners to generate a smooth, collision-free trajectory that is recalculated every second of the scenario, considering the environment and road objects.
Every vehicle in the scenario is uploaded from the workspace. It has information on multiple conjunct waypoints, with a minimum of two, with the first being at the starting point (according to a CG of the vehicle) and the final point being located at the road end, with the simulation ending when CG reaches that point. In these cases, with only two waypoints, the car will travel forward in a straight line with a constant velocity. However, if we want to add some curvature to the highway (not a case of studies) or if the vehicle changes lanes during the simulation (one or more times), we must add several waypoints to represent the trajectories we require the car to perform.
Before explaining the Path Planning subsystem, we need to explain the concept of Fernet coordinates. The Frenet coordinate system is a way to represent the position on a road more intuitively than the traditional cartesian coordinates (mathematically more straightforward representation). A six-element row vector represents the global states of a vehicle on cartesian coordinates:

$$globalState = [X, Y, \theta, K, V, A] \quad (3.20)$$

Where (X, Y) are vehicle position in meters, θ is the orientation angle in radians, K is the curvature in m^{-1}, V is the vehicle velocity in m/s, and A is the vehicle acceleration in m/s^2.

Frenet coordinates use instead variables s and l to describe the vehicle's position, where s represents longitudinal displacement (the distance along the road, arc length), and l represents the lateral displacement (the side-to-side position on the road relative to a reference path). This system aims to describe the kinematic properties of an object moving along a continuous and differentiable curve in a three-dimensional Euclidean space. More specifically, it is defined by the derivatives of the tangent, normal, and binormal vectors of the displacement of the moving object. In this scenario, we present displacement over a plan, so the binomial vector is disregarded. Thus, the Frenet states are a six-element row vector represented by position, velocity, and acceleration relative to a reference path:

$$frenetState = [s, \frac{ds}{dl}, \frac{d^2s}{dl^2}, l, \frac{dl}{ds}, \frac{d^2l}{ds^2}] \quad (3.21)$$

Though understanding the motion planner of Lane Change Maneuver on a high level, we need to divide the main operation into three core functions: Dynamic Capsule List, Reference Path Fernet, and Trajectory Generator Frenet [3]. Dynamic Capsule List uses the functionality of Simulink, which generates dangerous areas or ovals in turn of a target object. These capsules are created every instant over a determined predicted future trajectory and represent a hazardous zone of a potential collision in each instant of time. This function allows us to check for collisions and discard all paths invading this capsule area in optimization. If the vehicle terminal states when choice a path collides or intercepts the capsule area automatically, this path is considered to have a low value as an optimal solution. Conversely, Reference Path Frenet [4] converts the coordinates from global coordinates to Fernet coordinates or from Fernet coordinates to global coordinates. Finally, when given a conjunct of terminal states, the Fernet Trajectory Generator will generate multiple candidate trajectories that the ego vehicle can apply to achieve the terminal state that it pretends to reach. All conducted trajectory generation and motion predictions happen in Fernet Space.

The first thing we do is execute a coordinated transform from Global to Fernet for the Ego Vehicle and surrounding Most Important Objects (MIOs), which are received from Scenario and Environment Block. With the surrounding objects, traffic conditions, and map information, the Terminal State Sampler will define terminal states according to three major ADAS: Lead Car Following, Cruise Control, and Lane Change. The preferred lane or maneuver to apply will depend on where it finds itself in the scenario. The first step is to identify all safe MIOs based on TTC, usually considered to save all vehicles traveling behind the Ego Vehicle at inferior velocity, all vehicles too far away on the road, or vehicles traveling side by side at the same or lower velocity. However, as mentioned, the car can use three ADAS in three significantly different situations. First, if the current Ego lane (lane where the Ego vehicle is traveling) is safe and the unsafe targets are detected or not in the left or right adjacent lane, the preferred lane would be the current Ego Lane. In this case, the desired performance will be either Cruise Control or Lead Car Following mode, depending on whether there is a safe target in front of the Ego car. If there are no MIOs in the Ego lane, it will perform determined terminal states according to Cruise Control Mode. In this case, the terminal state generated will be the car's position in the Ego lane according to a set velocity in a close space of time.

[3]https://www.mathworks.com/help/nav/ref/trajectorygeneratorfrenet.html
[4]https://www.mathworks.com/help/nav/ref/referencepathfrenet.html

Autonomous Emergency Braking System

Suppose there is a safe target in the Ego lane. In that case, it will use the Lead Car Following mode, which means the terminal state will keep the Ego vehicle in the same lane. However, if necessary, it may lightly reduce the velocity to accommodate the initial velocity set when the scenario was created and the velocity gap between the two vehicles. This implies that Ego terminal state prediction will always be positioned slightly behind the lead vehicle. Finally, in case of an unsafe MIO in the Ego lane and one of the adjacent lanes is safe (right or left), the Terminal Generator Sampler will generate the terminal states according to the Lane Change mode. These terminal fernet states are defined according to the direction of the lane change and the curvature of the trajectory. All this process occurs inside the Terminal State Sampler, which occurs inside the Fernet Space.

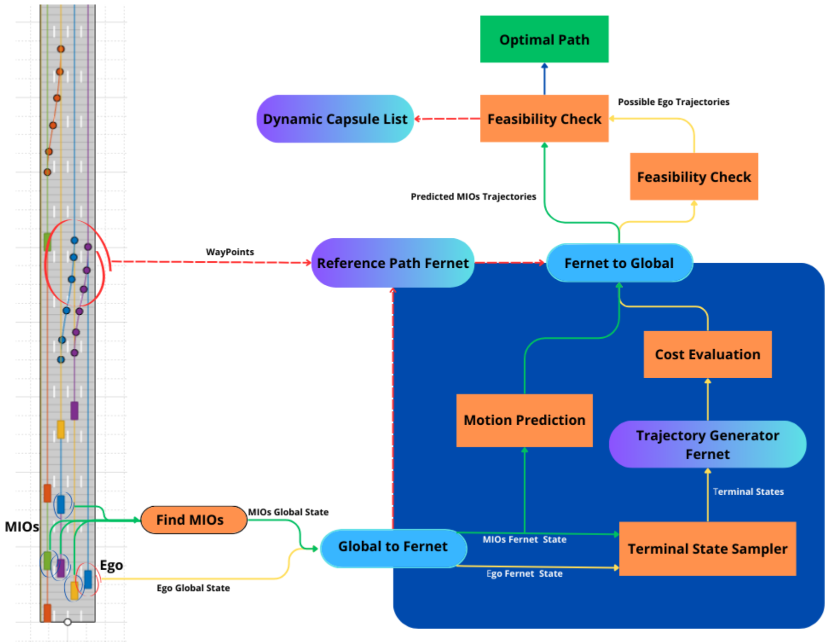

Figure 3.9: Schematic of Motion Planner

The terminal states defined by these three ADAS are then sent to the Fernet Trajectory Generator, creating multiple trajectories to achieve these terminal states. Then, as in all typical AI algorithms, these trajectories that Ego vehicles retrieved are now sent to Cost Functions and a Feasibility Function. These two functions will allow us to end up with a feasible trajectory with the lowest cost. Simultaneously, it also predicts target trajectories for all the MIOs, which are then intertwined with Ego vehicle possible trajectories to check for collisions and discard all that end in a crash. Ultimately, we will obtain all possible collision-free Ego trajectories sorted by cost and feasibility, where the first option in the list is the best or optimal trajectory that the Ego vehicle should apply for that specific situation where it finds itself. All description mentioned above is represented in a schematic of the motion planner for lane change maneuvers, shown in Figure 3.9.

This schematic is directly translated for the block. For example, the first two blocks transform the Ego Vehicle and MIOs into Fernet coordinates, like in the schematic. Next, we can divide this block into three subsystems: the Terminal State Sampler, MIOs Motion Prediction, and Ego Motion Planner. The Terminal State Sampler defines terminal states for trajectory generation based on surrounding traffic and map information. The first block in this subsystem is a function that receives Ego e MIOs Fernet States and map information to update the Ego vehicle position information, having as the function to determine the current lane and to find a preferred lane. Then, it finds the terminal states for the three aforementioned driving maneuver modes and will concatenate all terminal states to be sent to the next phase, as shown in Figure 3.10. It is also important to mention the essential variable, "PlannerParams," which is a bus that holds the planning parameters necessary for the subsystem. There are five required data types for the planning process:

1. Planning Parameters contain variables relative to time management, like time horizon, time resolution, replan rate, and preferred lane.

2. Safety Parameters with TTC, safety gaps, and front extension.

3. Cost Weights for lateral deviation, time, and velocity 3.26.

4. Feasibility Parameters consider maximum acceleration, curvature, yam rate, and minimum velocity.

5. Behavior Mode to enable Cruise Control mode (CC) 3.22, Lead Car Following mode (LCF) 3.24 and Lane Change mode (LC) 3.25.

The other variable is the bus "MapInfo," which contains details about the highway road, like the number of lanes, lane width, lane center, and data about global plan points (waypoints). Also, the preferred lane is where the Ego can travel safely or without risk of collision. This is done using a fundamental metric for AEB, the TTC, in which the lane with the lowest TTC is preferred. One problem that AEB will resolve is when there is no collision-free lane, and the only alternative is to keep traveling in the current lane, but this option, in time, will lead to a collision.

The Terminal Sates Generator is divided into three modes, according to 3 ADAS. The first is Cruise Control mode because there are no front vehicles in their longitudinal displacement. This means that the longitudinal position has no restriction (Nan), the Ego longitudinal velocity will be determined by the Ego velocity set at the start of scenario testing ($\dot{s}_{des}$), and logically, for this mode the acceleration will be zero. The Ego vehicle pretends to remain in the same lane, so the terminal state lateral deviation ($l_{expLane}$) will be the lateral offset of the vehicle regarding the Ego lane center (which is estimation according to its lane centers from map information), while lateral velocity and lateral acceleration will be zero. Finally, the last variable used is the time horizon (t_{hor}), which determines the number of steps (N-seconds) in the future to predict the terminal state.

$$cruiseControlState = [Nan, \dot{s}_{des}, 0, l_{expLane}, 0, 0, t_{hor}] \quad (3.22)$$

The second is the Following Lead Car mode, similar to the previous one. However, there is a safe target in front of the Ego Vehicle, so velocity must be regulated to avoid collision with the front target. This means that we need to predict the displacement of all target vehicles close to Ego in the Ego lane in a time horizon. When detected, the nearest vehicle to ego, or lead vehicle, can extract that vehicle's longitudinal and lateral distance (in Fernet) and velocity. The acceleration will be zero because all target cars travel at a constant velocity:

$$closetLeadVehicleState = [s_{Lead}, \dot{s}_{Lead}, 0, l_{Lead}, \dot{l}_{Lead}, 0] \quad (3.23)$$

The terminal state of this mode is calculated based on the notion that the Ego vehicle will mimic the lead vehicle, receiving his lateral distance, lateral velocity, and longitudinal velocity. Still, longitudinal distance needs to subtract a safety distance factor so the terminal state doesn't end up close or inside the danger capsule of the lead vehicle.

$$followLeadVehicleState = [(s_{Lead} - s_{safety}), \dot{s}_{Lead}, 0, l_{Lead}, \dot{l}_{Lead}, 0, t_{hor}] \quad (3.24)$$

Finally, the Lane Change mode corresponds to the terminal states that need to be generated for other adjacent lanes because there is an unsafe target in the Ego lane right in front of the primary vehicle. Any object in front of the ego that breaches the distance or velocity safety gap is automatically considered unsafe by traveling too slowly or too close to the Ego. The process starts similarly to the previous modes by determining the current Ego lane, but now, all adjacent left or right (if exit) must also be evaluated. Suppose these lanes are valid or unrestricted of targets. In that case, a similar function of Cruise Control is applied, but in that lane, when the only difference is that the longitudinal velocity will be equal to the correct velocity instead of the Ego velocity set for the scenario.

$$cruiseControlState = [Nan, \dot{s}_{cur}, 0, l_{expLane}, 0, 0, t_{hor}] \quad (3.25)$$

Every one of these modes will output a set of terminal states in a bus called "TerminalStates", and all these terminal states will be concatenated in a bus called "TerminalStatesConcatenation", which is composed of the number of combinations, the combinations (all the states) and the correspondent driving mode (CC, LFV, and LC). The combinations and driving modes can only represent, at a maximum, values for 60 terminal states.
The next phase or subsystem, Motion Planner, evaluates all terminal states generated and attributes a cost value to each of them, then sorts the states by cost and feeds into a function that produces trajectories for the ego vehicle. This function will generate multiple candidate Ego trajectories to reach the terminal states. With the trajectories defined, they filtered through kinematic feasibility functions, where trajectories with an excessive curvature or a yam rate that was too high or too low will be automatically removed. Pathways and terminal states with no viable trajectory according to predefined standards or behaviors are discarded. Now, with feasible trajectories, they are again reduced by removing all the trajectories that result in collisions against predicted trajectories of MIOs. The third and last subsystem, the MIOs Motion Prediction block, predicts the trajectories of target objects according to their trajectory waypoints.

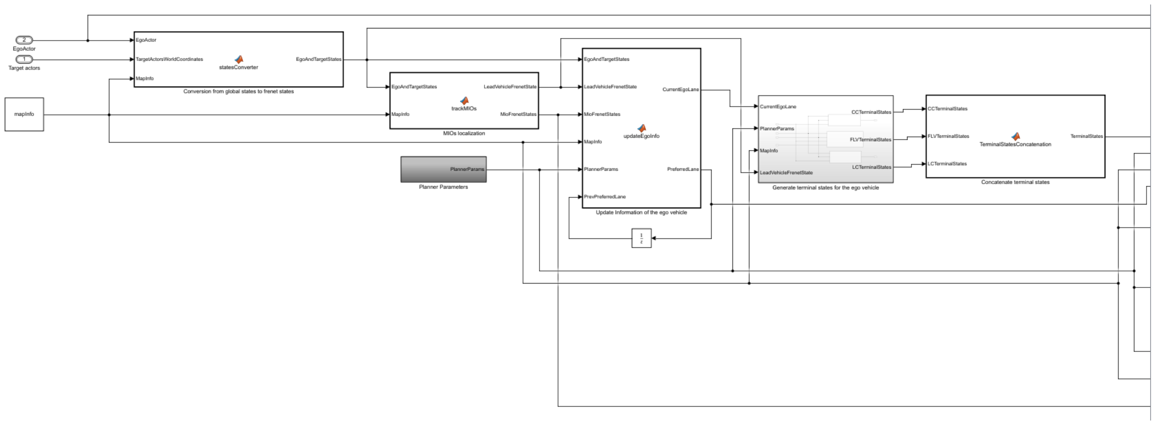

Figure 3.10: Path Planning Block: Terminal State Sampler

But how is the cost concretely measured? The cost of a terminal state is determined by three significant factors: the cost associated with lateral deviation, the time cost, and the velocity cost. All attributes have the same importance because we pretend to have a planner to generate a terminal state close to a center lane with a low-velocity variation (acceleration) and the slowest trajectory possible.

$$C_{TOTAL} = C_{latDev} + C_{time} + C_{speed} \quad (3.26)$$

The lateral deviation cost C_{latDev} penalizes all terminal states that deviate too much from the center of a lane (Ego lane or adjacent lanes). This is possible using the minimum argument of the lateral deviation between the current Ego lateral deviation $L_{egoLane}$ and the lateral deviation of any terminal states $L_{terminalState}$. The scheme will prioritize modes that will make the Ego vehicle stay in the Ego Lane and only change if there is no safer option. On the other hand, independently of the mode activated, it will prioritize all the terminal states close to the lane center. This aims to give the vehicle a more fluid and natural movement, and prioritizing driving in the center of the lane is an easy way to guarantee road security and vehicle management.

$$C_{latDev} = W_{\triangle L} * \triangle L \quad (3.27)$$

$$\triangle L = argmin(|L_{egoLane} - L_{terminalState}|) \quad (3.28)$$

All the other weights will follow the same premise, which, in case of time, will prioritize all the terminal states with a longer time to reach that terminal state. This means that a slower-to-achieve terminal state typically has a smoother curvature. This leads to a more friendly driving experience and increases the driver's comfort. Therefore, this equation is the only one where the weight will present a negative valor.

$$C_{time} = -W_{\triangle t} * \triangle t \quad (3.29)$$

Finally, the velocity costs prioritize all terminal states that aim to preserve the current Ego velocity. With a lower velocity variation, we obtain a softer variation of acceleration and a lower jerk value, increasing the driver's comfort. This is received by the differential of terminal state velocity $v_{terminalState}$ and the Ego velocity v_{ego} defined by the driving scenario.

$$C_{velocity} = -W_{\triangle v} * \triangle v \quad (3.30)$$

$$\triangle v = v_{terminalState} - v_{ego} \quad (3.31)$$

Then, with a cost associated with each terminal state, they will be sorted in ascending order, with the lowest cost being the first choice as a terminal state to adopt. Then, equations of fifth-order polynomials 3.33 and 3.34 are used to generate trajectories with smooth and continuous curvature to retrieve a conjunct waypoint that will represent a more realistic and dynamic representation of the vehicle's motion, having in consideration its velocity over the time horizon.

The updates of all global trajectories occur at every replan cycle based on the information received from other blocks. The default replan cycle is one second, created by a pulse generator. As such, there is a necessity for every generated trajectory bus for Ego to have a boolean to indicate if the trajectory is newly created. Each trajectory has its bus that contains mainly the trajectory waypoints and respective times where the maximum number of waypoints given to a determination trajectory is measured based on the relation between time horizon t_{hor} and time resolution t_{res}, plus the actual waypoint where the Ego vehicle finds itself.

$$maxTrajPoints = \frac{t_{hor}}{t_{res}} + 1 \quad (3.32)$$

Where t_{hor} is the time horizon to predict the ego vehicle's expected behavior, N-seconds in the future. t_{res} is time resolution of the sampled trajectories in seconds[5]. Plus the waypoint referent to the current CG of Ego vehicle. Both are defined in Planner Parameters.

Furthermore, we will present a deep dive into the generation of possible trajectory process using the equation 3.1, which, when converted to fernet coordinates, produces the equations for longitudinal displacement over time $s(t)$ and lateral displacement over time $l(t)$:

$$l(t) = a_6 t^5 + a_5 t^4 + a_4 t^3 + a_3 t^2 + a_2 t + a_1 \quad (3.33)$$

[5]https://www.mathworks.com/help/mpc/ug/choosing-sample-time-and-horizons.html#bujyygn

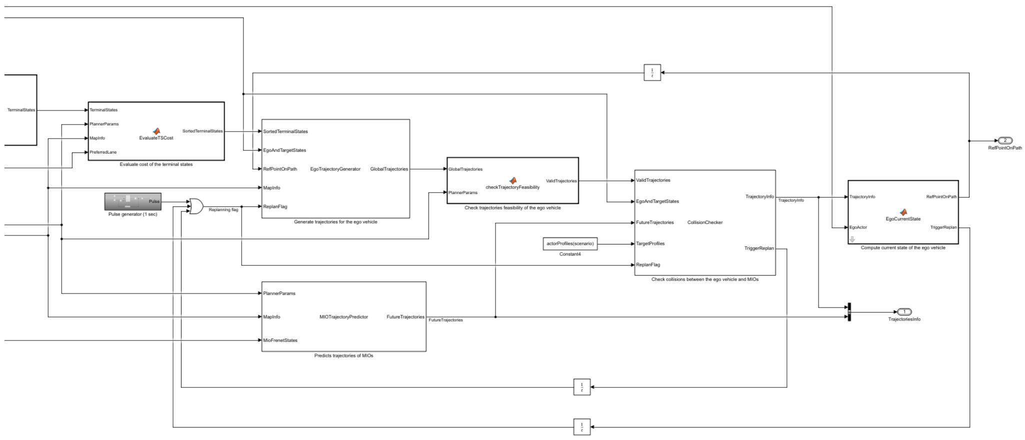

Figure 3.11: Path Planning Block: MIOs Motion Prediction and Ego Motion Planner

$$s(t) = b_6 t^5 + b_5 t^4 + b_4 t^3 + b_3 t^2 + b_2 t + b_1 \quad (3.34)$$

As shown, both equations have similar logic, and because they describe an object traveled distance over time, using these formulas, we can quickly determine the velocity $\dot{s}(t)$, acceleration $\ddot{s}(t)$, and jerk $\dddot{s}(t)$ by derivating multiple times.

$$\dot{s}(t) = 5a_6 t^4 + 4a_5 t^3 + 3a_4 t^2 + 2a_3 t + a_2 \quad (3.35)$$

$$\ddot{s}(t) = 20a_6 t^3 + 12a_5 t^2 + 6a_4 t + 2a_3 \quad (3.36)$$

$$\dddot{s}(t) = 60a_6 t^2 + 24a_5 t + 6a_4 \quad (3.37)$$

Futhermore the start boundary conditions at $(t = 0)$ are $s_{start} = a_1, \dot{s}_{start} = a_2, \ddot{s}_{start} = 2a_3$ and $\dddot{s}_{start} = 6a_4$. Therefore, replacing the equation 3.1, the formulation of longitudinal or lateral movement for a jerk-optimal trajectory generation is defined as:

$$s(t) = \frac{1}{6}\dddot{s}_{start} t^3 + \frac{1}{2}\ddot{s}_{start} t^2 + \dot{s}_{start} t + s_{start} \quad (3.38)$$

The same is applied to velocity, acceleration, and jerk:

$$\dot{s}(t) = \frac{1}{2}\dddot{s}_{start} t^2 + \ddot{s}_{start} t + \dot{s}_{start} \quad (3.39)$$

Autonomous Emergency Braking System

Automobile (Sedan)	Measures
Length	4.7m
Width	1.8m
Height	1.4m
Wheelbase	2.8m
Front Overhang	0.9m
Rear Overhang	1.0m

Table 3.2: Sedan dimensions in Scenario Designer

$$\ddot{s}(t) = \dddot{s}_{start}t + \ddot{s}_{start} \quad (3.40)$$

$$\dddot{s}(t) = \dddot{s}_{start} \quad (3.41)$$

A trajectory with the best values for velocity, acceleration, and jerk (which, as mentioned, is used to measure the body displacement of drivers and their comfort), according to these equations, collision-free and feasible, will be defined as the optimal trajectory to be adopted by the Ego vehicle.

Finally, a glance at the collision-free trajectory process. This operation has a higher computational cost because it requires the prediction of MIOs' trajectories in the future, assuming that all target vehicles move forward and with constant velocity during the simulation time. The detection of the collision process uses Dynamic Capsule List [6] object, which controls two lists of capsule-based collision objects, one for ego and one for the targets. Each collision object contains an identification, three elements row, and the object geometry. To access the vehicle characterize 3.2 as variables it was used the bus Actor Profiles. With vehicle dimensions, it can generate capsule-based collision objects to target vehicles. Then, it checks for collisions between ego and target vehicles during and between replan cycles, with the first valid collision-free trajectory chosen as the optimal trajectory. Due to its computational cost, this Matlab tool is the last to be executed in the Path Planning.

The Path Planning will result in two essential outputs: the variable RefPointOnPath, which will be used on the Lane Change Controller and contains all data necessary for the MPC to pass from the current state to the next state (waypoint information), with values for velocity, curvature, yam, and others that are updated every time cycle and allow a constant adjustment from the MPC. The variable Trajectories Info is sent to the Visualization subsystem and contains information on the target vehicle's future and ego trajectories.

3.5 Path Planning with AEB Controller

Adaptive model predictive control (MPC) aims to compute optimal control actions while satisfying all requisites established by the Path Planning subsystem. This will allow us to fulfill

[6] https://www.mathworks.com/help/nav/ref/dynamiccapsulelist.html

the primary goal of autonomously steering a car whose lateral vehicle dynamics change with time due to the varying longitudinal velocity. The MPC predicts future behavior using dynamic models based on linear-time-invariant (LTI) to accumulate the high rate of variables changing over time [62].
The collision-free optimal trajectory from the Path Planning subsystem needs a Controller to apply that reference trajectory. Furthermore, to receive a reference point on the path, the longitudinal velocity from the Vehicle Dynamics subsystem and the TTC from the Metrics Assessment subsystem are also needed. On the other hand, the outputs of this subsystem will be used in the Vehicle Dynamics, thus establishing a closed-loop system.
This subsystem is divided into two parallel components. The primary controller block's primary mission is to adjust the vehicle to the changes needed for the Path Planning to achieve the following state (waypoint). In contrast, the secondary block aims to address the collision scenarios and the AEB system.

3.5.1 Path Planning Controller

The primary Controller is an upper-level, path-following controller block that aims to keep the vehicle in a marked highway lane while maintaining a user-set velocity. The data received from the planner consists of several parameters to achieve the next state. However, we need four main parameters: reference velocity, reference curvature, reference yam angle, and lateral offset deviation. The controller must know the ego vehicle's lateral deviation and relative yam angle to the virtual lane. It is also necessary to convert the coordinates from ISO 8855 to SAE J670 standard, as shown in Figure 3.12.

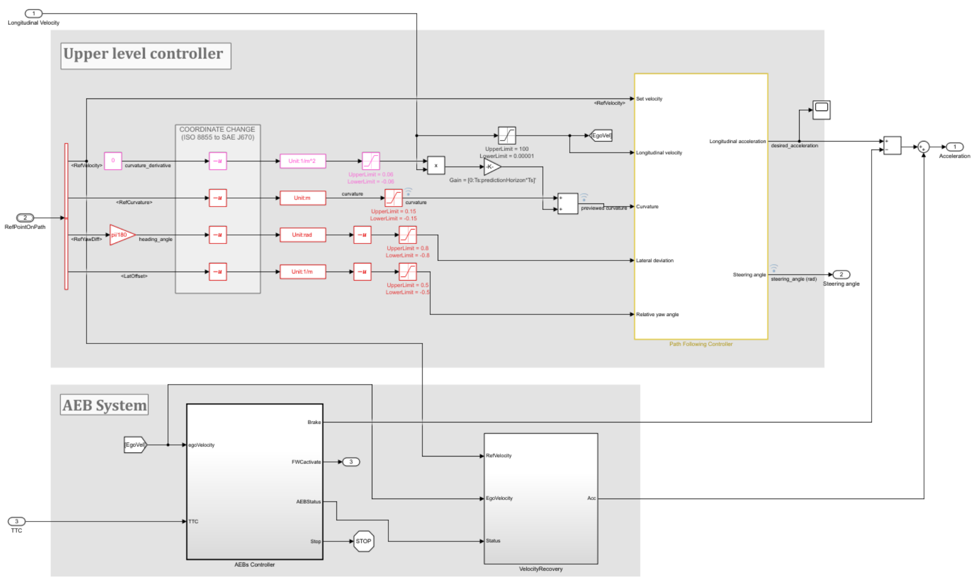

Figure 3.12: Path Following Controller block

The Path Following Control [7] from Model Predictive Control Toolbox [8] combines the lateral and length longitudinal control using the Adaptive Model Predictive Control. This Simulink

[7]https://www.mathworks.com/help/mpc/ref/pathfollowingcontrolsystem.html
[8]https://www.mathworks.com/products/model-predictive-control.html

out-of-the-box block will generate the longitudinal acceleration and steering angle. This controller is chosen over other control methods, such as the PID controller. The primary reason is that applying and tuning the PID controller for large systems like this becomes challenging and sometimes ineffective. With an MPC, it is possible to reduce tuning efforts, support multivariable (predominant case in this project), better handle constraints, and have a preview capability, which means slowing down the vehicle if a target is detected. A traditional MPC controller is less efficient for handling the varying dynamics over time because it uses a constant internal planned model. As such, it is necessary to have an adaptive MPC for this project. The Path Following Controller follows a center line of a lane, and when the Ego vehicle moves away from that center line, the lateral deviation and relative yam angle change. To combat this process, the Lane Keeping Assist (LKA)[9], and its primary goal is to drive both lateral deviation and relative yam angle close to zero, traveling along an imaginary that represents the center of the lane by adjusting the front steering angle of the Ego car. Combined with the road's curvature, this will define the lateral control part of the Path Following Controller block. In contrast, the longitudinal control tracks the velocity between the Ego set velocity by the user at the beginning of the simulation and the velocity of the lead vehicle (if one exists), maintaining a safe distance by adjusting the longitudinal acceleration of the Ego car. These are called Adaptative Cruise Control (ACC)[10]. To perform a lane change, the two parts must work simultaneously: lateral and longitudinal controls. The Adaptative MPC's ability to accommodate operating conditions over time will allow a reduced error in accuracy with a fixed plant model structure that changes over a finite prediction horizon. The default ego vehicle Model Predictive Controller [11] uses plant, disturbance, and noise models for prediction and state estimation, being defined as follows:

$$x(k+1) = Ax(k) + B_u u(k) + B_v v(k) + B_d d(k) \quad (3.42)$$

$$y(k) = Cx(k) + D_v v(k) + D_d d(k) \quad (3.43)$$

Where x is a state, k is the time index, u is the manipulated outputs (outputs that the MPC needs to adjust), v is the measured disturbance input, d is the unmeasured disturbance inputs (white noise), y is the output of the system. A is the state matrix, B_u, B_v, B_d, are the input to state matrices of u, v d respectively. C is the state to output matrix, D_v, and D_d are the feedthrough matrices of v and d respectively.

In other words, u corresponds to the vehicle's front wheel steering angle and acceleration (the output controlled by the MPC), while v indicates the previewed curvature.

As mentioned, an Adaptive MPC also requires an extra input, which contains an updated plant model of the system and the nominal operating point, at which its model applies to

[9]https://www.mathworks.com/help/mpc/ref/lanekeepingassistsystem.html
[10]https://www.mathworks.com/help/mpc/ref/adaptivecruisecontrolsystem.html
[11]https://www.mathworks.com/help/mpc/gs/mpc-modeling.html

obtain the LTI approximation. The updated plant model combines two state-space models: one for ACC and one for LKA. The ACC system controls the velocity of the Ego vehicle using the throttle, which is defined as follows:

$$\dddot{x} = \frac{1}{\tau s + 1} \dddot{x}_{des} \quad (3.44)$$

Where $\dddot{x}$ is the throttle, $\dddot{x}_{des}$ is the desired acceleration, and τ is the desired acceleration time constant ($\tau = 0.5s$ for analysis and simulation).

This translation of the desired acceleration to throttle is done between the adaptative MPC and the vehicle dynamics, where the throttle will be fed directly to the 3DOF bicycle block 3.6. Following this, the predictive state-space model for Adaptive Cruise Control (ACC) is defined as follows:

$$A_1 = \begin{bmatrix} -\frac{1}{\tau} & 0 \\ 1 & 0 \end{bmatrix}, B_1 = \begin{bmatrix} \frac{1}{\tau} \\ 0 \end{bmatrix}, C_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}, D_1 = 0 \quad (3.45)$$

In addition, the predictive state-space model for Lane Keeping Assist (LKA) is defined as follows:

$$A_2 = \begin{bmatrix} \frac{-2(C_f+C_r)}{mV_x} & \frac{-V_x-2(C_f l_f - C_r l_r)}{mV_x} \\ \frac{-2(C_f l_f - C_r l_r)}{I_{zz}V_x} & \frac{-2(C_f (l_f)^2 - C_r (l_r)^2)}{I_{zz}V_x} \end{bmatrix}, \quad (3.46)$$

$$B_2 = \begin{bmatrix} \frac{2C_f}{m} \\ \frac{2C_f l_f}{I_{zz}} \end{bmatrix}, \quad (3.47)$$

$$C_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (3.48)$$

$$D_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.49)$$

All these variables are taken from the Vehicle Dynamic Block (3 DOF bicycle block 3.6), where m is the vehicle total mass, I_{zz} is the yaw moment of inertia, l_f is the longitudinal distance from Ego vehicle CG to front tires, l_r is the Longitudinal distance from Ego vehicle CG to rear tires, c_f is the front tires cornering stiffness, c_r is the rear tires cornering stiffness and V_x is the car longitudinal velocity.

As the Path Following Control system combines these two previous systems, its model results in the following predictive state-space model:

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \quad (3.50)$$

$$B = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix}, \quad (3.51)$$

$$C = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix}, \quad (3.52)$$

$$D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}, \quad (3.53)$$

A close look at the Path Following Controller block shows three tabs: Parameters, Controller, and Block. The Parameters tab provides the bicycle model parameters associated with the Ego vehicle. The Controller tab has actuator limits such as minimum and maximum steering angles and longitudinal acceleration and also includes an MPC setting to tune the controller performance.

3.5.2 AEB system

The AEB controller needs two significant variables: Ego longitudinal velocity from the Vehicle Dynamics subsystem and TTC. This last parameter is calculated in the Metrics Assessment subsystem and, accordingly, with the environmental parameters, will perform a deceleration maneuver until the car stops, or when danger is not more imminent, it will return to the Path Following Controller. Using a stopping time calculation approach, the AEB Controller subsystem can implement the Forward Collision Warning (FCW) and all three braking stages, with access to information relative to velocity and acceleration, in this case, the deceleration, which corresponds to a negative acceleration necessary applying to stop the vehicle motion. The formula can determine the time required to immobilize a vehicle:

$$T_{stop} = \frac{V_{ego}}{\alpha_{brake}} \quad (3.54)$$

Where T_{stop} refers to the time until the vehicle becomes stationary, V_{ego} longitudinal velocity of the ego vehicle, and α_{brake} is the standard brake acceleration applied in the particular state.

In other words, it calculates the necessary time for a specific deceleration force to stop a vehicle traveling at a determined velocity entirely. This value is later compared to TTC off

the vehicle and the forward vehicle to classify the correct braking state where the car finds itself and acts accordingly.
The AEB subsystem is divided into three central stages, defined by their value of deceleration force. Each stage has a pre-defined value of $5.8m/s$, $7.3m/s$, and $9.8m/s$, respectively, with FCW defined at $4m/s$. These states are denominated as Partial Braking 1 (PB1), Partial Braking 2 (PB2), and Full Braking stage (FB). These deceleration values and the Ego longitudinal velocity make it possible to determine how far in advance it is necessary to apply the brakes to stop the vehicle entirely before the collision. Moreover, when calculating the time to these stages, a standard error margin in time calculation is necessary to give enough time to rectify minor errors in the process, which can occur due to various factors, like road friction, poor weather, computational error, and others. It is preferable to prevent a mistake by being too cautious and reckless, even if that inevitably will reduce the vehicle's performance. This will occur due to the early activation of the partial brake, which will have a cascade effect, as shown in the next chapter. Typically, the value applied is the time necessary to use the time cycle one more time (the simulation runs in seconds, so it adds one second). This inadvertently allows the entire system in the loop to have one more chance to check all parameters and go back through the process, especially the Path Planner. It is also important to mention that the FCW works independently from braking stages and is generally activated before braking. Still, it is also possible to appear when the vehicle realizes it needs to start the full brake immediately, and the FCW is activated simultaneously. This typically happens when a target vehicle suddenly cuts to the front of the Ego vehicle, changing the environment dynamics and the system's overall behavior. The calculation of FCW is based on a derivation of the main formula:

$$T_{FCW} = T_{reaction} + \frac{V_{ego}}{\alpha_{brake}} \quad (3.55)$$

In this case, a reaction time that corresponds to the predicted time of the execution of the driver is taken into consideration to add the time that the driver needs to react to the illumination of the warning signal $T_{reaction}$, which is defined as approximately 1.2 seconds. The value of deceleration applied must always be close or superior to the deceleration value of the Partial Brake 1 state. However, the FCW signal is activated before the PB1 but still activated to all braking states (PB1, PB2, FB), only being turned off when the car leaves a dangerous situation, or the vehicle successfully executes the braking maneuver to stop and the car finds itself full immobile.
The AEB systems will be divided into two subsystems. The first is used to calculate the times of every braking stage according to the rules and equation 3.54, which are influenced by the current longitudinal velocity of the Ego vehicle. The second subsystem aims to evaluate and analyze the state's situation based on TTC and the times-to-stop provided by the first subsystem. Next, by classifying the state and determining the corrective value of deceleration that needs to be applied at that time state, it will also define the AEB status (that ranges between zero, not activated, to three, corresponding to FB stage) and FCW activation. The overall idea of the project is defined in Figure 3.13. A parallel component evaluates if the car is entirely

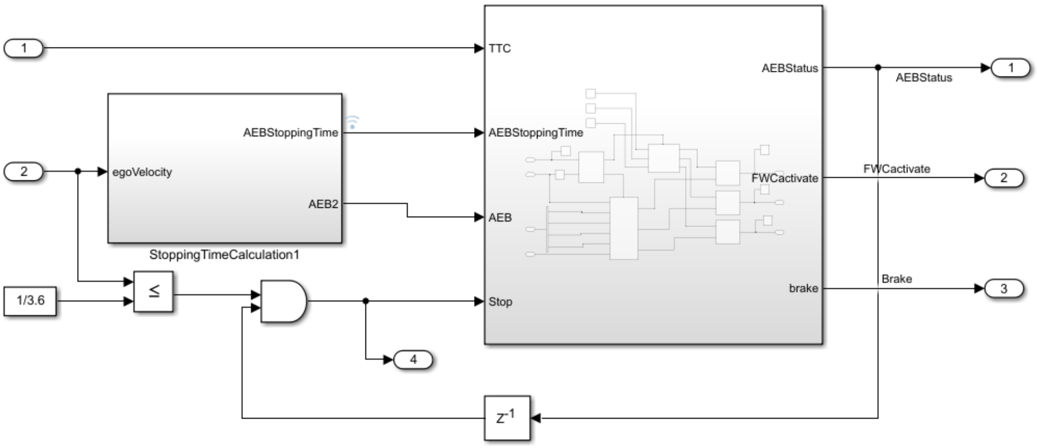

Figure 3.13: Subsystems contain on AEB Controller Block

stopped based on its AEB status and velocity. Furthermore, it is also essential to mention the variables "AEB" and "AEBStoppingTime", in which the first contains all the static values of deceleration of each braking state plus the margin error to be applied and time reaction variables. On the other hand, "AEBStoppingTime" has all time values obtained after using an equation of T_{stop}, containing all times for activating the respective braking states and FCW's ideal activation time. This Matlab Bus will be used to determine the state based on comparison with the value of TTC. Furthermore, the variable "stop" helps determine if the vehicle was stopped successfully, and if the Path Planner can't generate an alternative trajectory, the simulation will end. This is another criterion to stop the simulation; in addition to any vehicle reaching the end of the road, the simulation time runs out, or the Ego collides with a target car or leaves the road. Both Buses are illustrated in Figures 3.14 and 3.15, respectively.

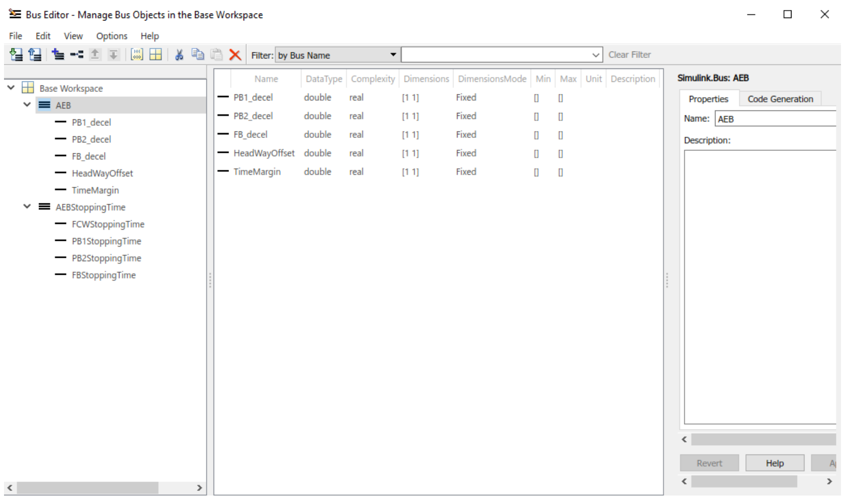

Figure 3.14: Structure of AEB Bus

It is essential to mention that the overall system has three consecutive stages of braking but with three different independent stages. The general idea is to pass for every stage gradually. However, this can not consistently be implemented because of vehicles cutting in or out of the Ego lane or another lead vehicle starts braking with a high degree of deceleration. As so

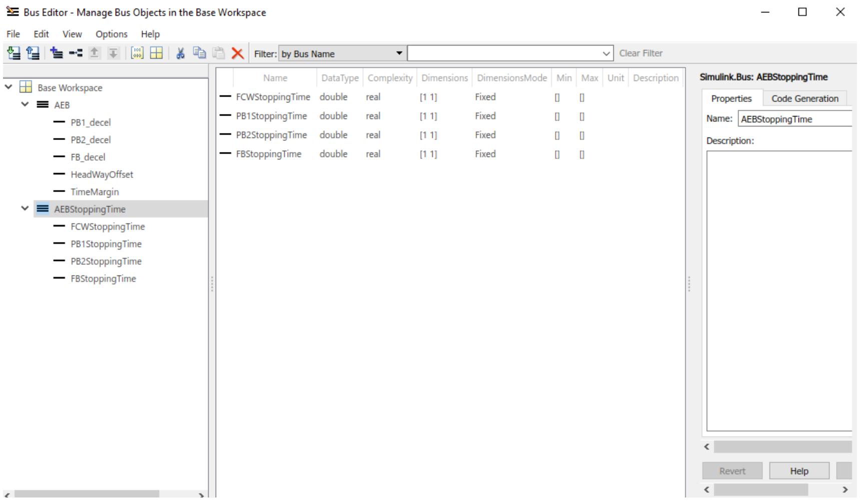

Figure 3.15: Structure of AEBStoppingTime Bus

is necessary in some more dangerous cases, the FCW is activated only when Partial Brake 1 is activated, or Partial Braking 1 is wholly ignored in circumstances of extreme variation. Environment states may demand that some stages of the braking process be ignored to rapidly apply the drastic deceleration forces, passing automatically to the last stage (FB), occurring in scenarios with target cars colling. This also depends on the Path Following Controller finding and executing alternative steering maneuvers to avoid collisions.

The last vital subject to be addressed is the Recovery Velocity block. This simple function is necessary for cases where the vehicle has performed a partial brake. Still, in the meantime, the Path Following Controller could find an alternative path to an adjacent lane, and the car needed to recover the lost longitudinal velocity during partial braking. These blocks will combat the inability of the MPC to add large values of acceleration (maximum of $2m/s^2$) if the longitudinal velocity gap is too large relatively to set velocity and if the Ego lane is currently clear of any unsafe target vehicle in front of Ego. This gradual acceleration boosts the vehicle's longitudinal velocity until it almost reaches its set velocity. It doesn't need to forcibly arrive at the specified velocity because when the velocity gap is smaller than the MPC maximum acceleration capacity, the MPC can more thoroughly manage the velocity gap than the recovery velocity. Still, if the new lane is unsafe (with an unsafe target vehicle present), this acceleration process will not occur, and MPC will keep its central control without receiving any reinforcement. Another case that pretends to mitigate is when the car realizes a complete brake and almost entirely stops. However, there is little time to wait to see if the road conditions become safe and if the Path Planning system can generate a trajectory to return to the driving process. Then, the recovery velocity will be applied again to return the vehicle to its new lane and reduce the time to return to its original set velocity.

3.6 Metrics Assements

The Metrics Assessment subsystem has two main goals: to allow the driver to monitor significant system metrics and to provide the calculation necessary for the AEB system, namely

TTC. This subsystem will need a lot of variables from the entire system. For example, it requires the lateral and longitudinal velocity from the Vehicle Dynamics subsystem, Ego actor dynamics, the Scenario and Environment subsystem's target actors, and a variable containing all the map information. All the metrics generated by this subsystem are displayed on a dashboard showing all these metrics while the simulations run.

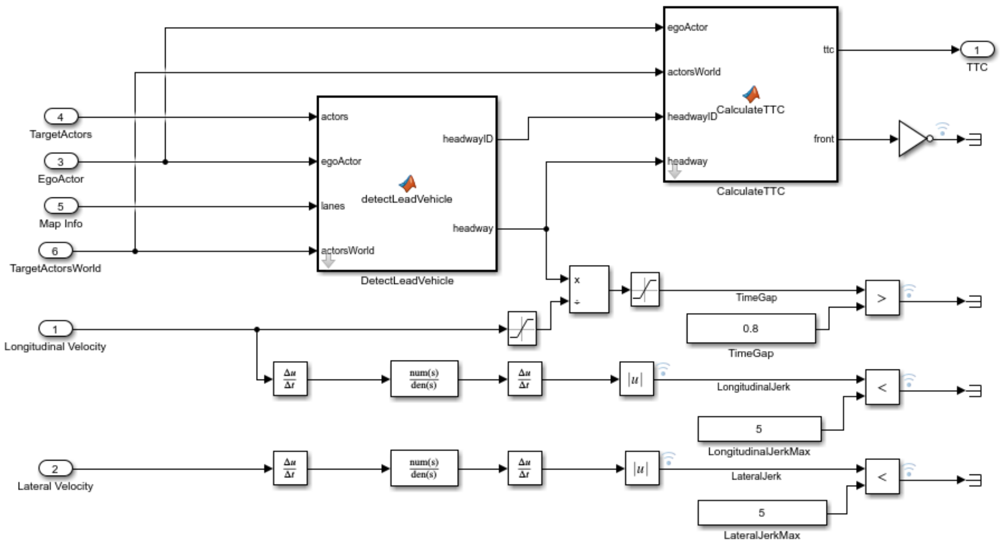

Figure 3.16: Metrics Assements block

The Detect Lead Vehicle block computes the distance between ego and lead vehicle, also known as headway, which calculates the time gap. The time gap is calculated using the distance to the lead vehicle $d_{headway}$ and the longitudinal velocity v_s of the ego vehicle.

$$t_{gap} = \frac{d_{headway}}{v_s} \quad (3.56)$$

The longitudinal jerk $\ddot{v}_s$ is calculated using the derivation of the longitudinal velocity, and the lateral jerk $\ddot{v}_l$ is computed using the derivation of the lateral velocity. Both metrics are compared to standard safe values, and if those values breach safety standards, the lamp in the dashboard corresponding to them will turn red, signaling that violation.

Longitudinal headway is the distance between the vehicle's rear axles reduced by the distance of the Ego rear axle to the front Ego_{rF} and reduced by the distance of the MIO rear axle to the rear Mio_{rF}. This phenomenon happens because the origin offset is the relative position [x,y] of the vehicle center concerning the rear axle. In contrast, the distance of the rear axle center to the vehicle center is equal to a negative origin offset $Ego_{offeset}$. The longitudinal headway is the distance from the front of the Ego to the rear of MIO, while the entire system tracks the vehicle position according to his CG. We must determine the front of the Ego and the rear of the lead vehicle:

$$Ego_{rF} = \frac{Ego_{len}}{2} + Ego_{offeset} \quad (3.57)$$

$$Mio_{rF} = \frac{Mio_{len}}{2} - Mio_{offeset} \quad (3.58)$$

$$headway = d - Ego_{rF} - Mio_{rF} \quad (3.59)$$

Where Ego_{len} and Mio_{len} are the length of the vehicle according to scenario data, and d is the distance between the vehicle's rear axles.
Furthermore, the most critical block in this system is the TTC calculator. This function will start by detecting the lead car if it exists, where, according to the information provided by the Path Planning subsystem, it will see the headway distance of any vehicle ahead and in the Ego lane, more critically, will return information about the lead vehicle identity, which will use the access the longitudinal velocity of the target leading vehicle. With all data reunited about Ego and lead vehicle, it is possible to calculate TTC using the following equation:

$$TTC = \frac{d_{rel}}{v_{rel}} \quad (3.60)$$

Where TTC in seconds is the time necessary with the current velocity and distance of both vehicles until a collision occurs, d_{rel}, which corresponds to the relative distance, including the lateral and longitudinal distance between the ego vehicle and the Most Important Object (MIO), and V_{rel} which corresponds to relative velocity including lateral and longitudinal velocity between the Ego vehicle and MIO.
All the cars in the simulation drive at a constant velocity with no acceleration and always in forward motion, which is usually applied in highway situations. However, there are some exceptions in the test cases scenarios 5 and 6 with accident collisions, where abrupt changes in speed between the colliding cars imply a high negative acceleration value for these vehicles due to their collision. The same is applied in cases of CCRb when some vehicles suddenly applied an abrupt or gradual value of declaration, or even in CCRs cases where some of the cars find themselves stationary.
To calculate the magnitude of the distance and velocity vectors between both vehicles (relative distance and velocity), we take advantage of the function from Matlab called cart2pol. This function uses polar coordinates to generate the magnitude vectors and their respective angles. The angle of the relative velocity is the relative yaw angle, which is the angle from the center line of a lane about the longitudinal velocity, as shown in Figure 3.17.

$$v_x = v * \cos(\theta_{yam}) \quad (3.61)$$

$$\theta_{yam} = \theta_{vel} - \theta_{dist} \quad (3.62)$$

where v_x is the longitudinal velocity, θ_{yam} is the relative yaw angle, θ_{vel} is the velocity angle, and θ_{dist} is the relative distance angle.

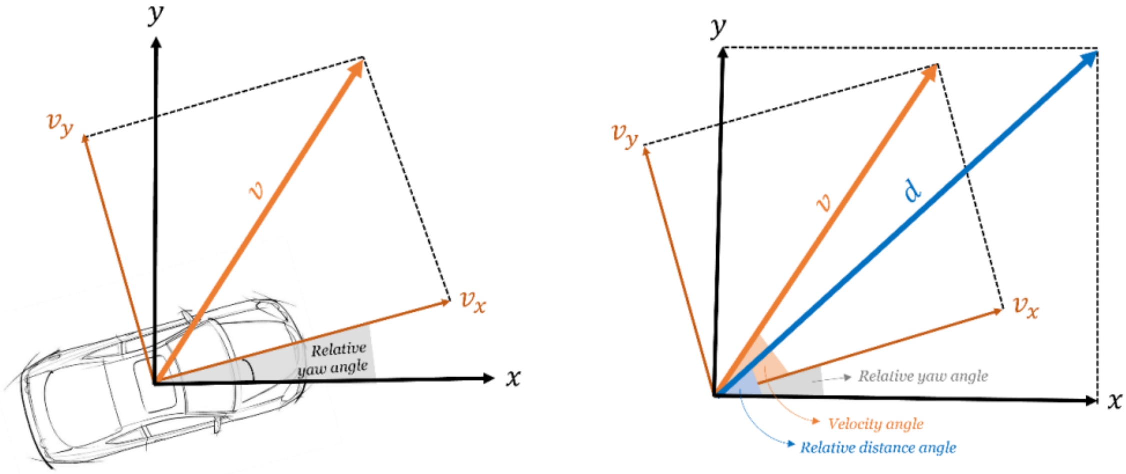

Figure 3.17: Relative velocity angle and the relative distance angle

This allows us to extract the relative velocity and distance between the Ego and the lead vehicle. Then, we can feed it to the function to calculate the TTC 3.60, which will be provided directly to the AEB system inside the Controller block.
All this calculation must occur at every instance of time, reevaluating the TTC and T_{stop} in each state transaction. This means evaluating all three stages every time cycle of the simulation is necessary to maintain a flux of accurate data about the environment and the dangerous situation that Ego faces.

3.7 Vizualization

The Visualization subsystem is a Matlab function that creates a Matlab plot using the inputs from the scenario, environment, and planner subsystems. Mainly, it is used to change the view setting of the simulation. It allows the chase view to be enabled, which plots the tridimensional vehicles, the trajectories of EGO, and the capsule list of all cars close by Ego. It also gives a top view, which provides a better display of the surrounding environment, giving us a better understanding of what is happening and why. Both points of view chase the Ego displacement throughout the simulation. This will enable us to correct and determine situations and impressions and rethink different driving scenarios or problems our system needs to challenge itself to separate. This Matlab function is already prepared to simulate any scenario from the Driving Scenario Designer in his high-level setting, allowing an easy way to change some visualization settings to analyze a simulation from very different ways.

Chapter 4

Tests, Results, and Discussion

This chapter presents the software-in-the-loop simulation with a relative explanation of the designed tests, the methodology used during the test trials, and a report of system performance results over all the test cases. This is followed by an analysis of the main results between the two work benches, one with the AEB system and another without the AEB system.

4.1 Driving Scenario Designer

The Driving Scenario Designer [1] app and the Automated Driving Toolbox [2] from Matlab allows us to build the simulation environment using a world coordinate system. All the scenarios are exported as a Matlab function, where the code generated is equivalent to the scenarios created. The Driving Scenario Designer app generates a driving scenario [3] object that contains information about a scenario's roads, vehicles (target actors), lanes, and objects. Then, simple alterations allow the generation of significant variations of each scenario by changing parameters like velocity, waypoints, yam angles, and other scenario variables. These scenarios will run in a closed-loop Simulink software model in real time. The Scenario Reader [4], presented on Scenario and Environment Block, reads these scenario variables and exports them to the bus containing the map's information. All scenarios need to include an Ego vehicle with a reference path with two waypoints: the initial waypoint (vehicle's CG) and the final waypoint at the end of the road. All roads have a longitudinal extension of eight hundred meters, and each lane has a lateral length of 3.6 meters (Driving Scenario Designer and CCR default parameters), with a maximum time set at thirty seconds for all scenarios. Still, it automatically stops if there is a collision between Ego and other vehicles or if any car reaches the last or ending waypoint (at the end of the road). This project has three types of highways, where the lateral length varies in each scenario. The road is centered at position zero on the y-axis, the lanes at the right have the lane center positioned with negative coordinates, and the left has positive ones, as shown in Figure 4.1.

All driving scenarios operate according to one of these roads. The case of two lanes (A) is used to test cases when the Ego maneuverability options are limited, the traffic is less dense, and the number of adjacent lanes is reduced to a minimum of one. This serves as the primary test case due to its simplicity and limitations. In contrast, this makes it a good test scenario for the AEB system because it is easier to create scenarios where the Ego car is forced to activate the brakes. As in all cases, the road center is set at zero, and the center lines of each lane are set at 1.8 meters and -1.8 meters, corresponding to the left and right lanes, respectively. We can see

[1] https://www.mathworks.com/help/driving/ref/drivingscenariodesigner-app.html
[2] https://www.mathworks.com/products/automated-driving.html
[3] https://www.mathworks.com/products/automated-driving.html
[4] https://www.mathworks.com/help/driving/ref/scenarioreader.html

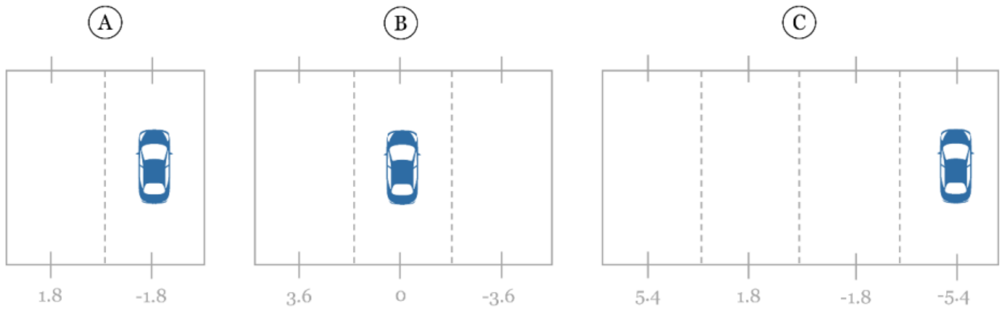

Figure 4.1: Three types of Highways lateral displacement

that the distance between every center lane equals 3.6 meters, as specified by the set lateral length. However, to test scenarios with more maneuverability options and very dense traffic, the three and four-lane roads were created to simulate these situations. In the scenario with three lanes (B), the center line of each lane, from left to right, is 3.6 meters, 0 meters, and -3.6 meters, respectively. In the case of four lanes (C), the maneuverability options and traffic density are taken to their maximum potential in these simulations, with the center lines of each lane, from left to right, being 5.4, 1.8, -1.8, and 5.4 meters, respectively.

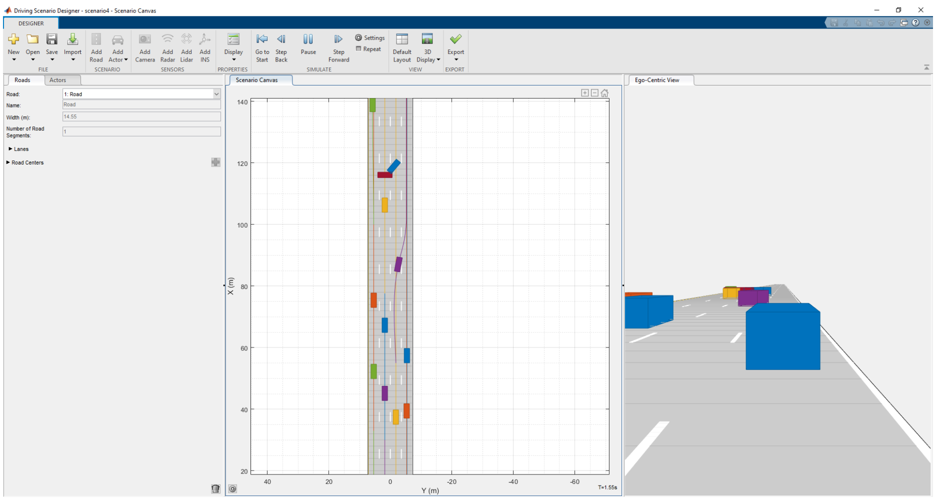

Figure 4.2: Creation of a scenario in Driving Scenario Designer app

4.2 The Seven Test Scenarios

The scenarios covered in tests 1 to 3 are the cases presented in the previous section as A, B, and C. In each scenario, minor adjustments and variations will occur due to lane changes of some target vehicles. This section aims to describe these same lane changes, what vehicle performs them, what velocity adjustments need to be done to avoid collision between target vehicles, and more. All non-ego vehicles must travel forward at a constant velocity, having no regard for the movements of other cars and unaware of surrounding dangers or can brake, making the creation of scenarios quite challenging. The scenarios covered between cases 4 and 6 are highway scenarios with four lanes, which address more specific situations. For

vehicle	velocity	starting position
ego	x	[50, -1.8, 0]
v1	23	[10, 1.8, 0]
v2	17	[70, -1.8, 0]
v3	22	[75, 1.8, 0]

Table 4.1: Scenario 1 target vehicles velocity and coordinates

example, cases 5 and 6 were developed for the CCRs situation, which aims to simulate a collision between two vehicles, thus suddenly creating two immovable obstacles blocking two lanes. On the other hand, scenario 4 is a variant of scenario 3 but a little more complex, with higher dynamism from the traffic and non-vehicles performing multiple lane changes. Some vehicles can have at least fifteen set waypoints, unlike standard two-way points. For every lane change enacted by the cars, an average of three to five waypoints were adopted with a yam rotation variation between zero and eleven exclusive and a longitudinal displacement between ten meters and twenty meters. All these value changes can vary according to the desired goal of the test scenario but maintain some order of naturality to it. This aims to give a high range of types of lane changes, with the majority having a smooth and natural curvature. Still, in some lane changes, especially the ones to avoid collision with immovable objects, the curvature is more aggressive and dangerous. Some of the lane changes of target vehicles are merely intended to prevent collisions between them. The dark blue square represents the Ego vehicle, and all the vehicle's positions are shown on the description tables of every scenario.

4.2.1 Scenario 1

In this example of two lanes, only vehicle 1 makes a lane change to the right between 300 and 330 meters. This occurs because the Ego vehicle has a speed greater than 23, making it one of the fastest cars, and to avoid a collision, it has to carry out an extra maneuver. For the case of CCRs, vehicle 4 was added, which stopped in the right lane at around 271m. In addition, vehicle 2 has to make two changes to avoid collision with the immobile vehicle 4, with the first turn being made to the left between 240m and 267m and then returning to the lane on the right between 340m and 356m. Finally, in the CCRb case, vehicle 2 in the right lane performs a braking process between meters 150 and 170. All notes of each vehicle's longitudinal velocity and position are shown in the table 4.1.

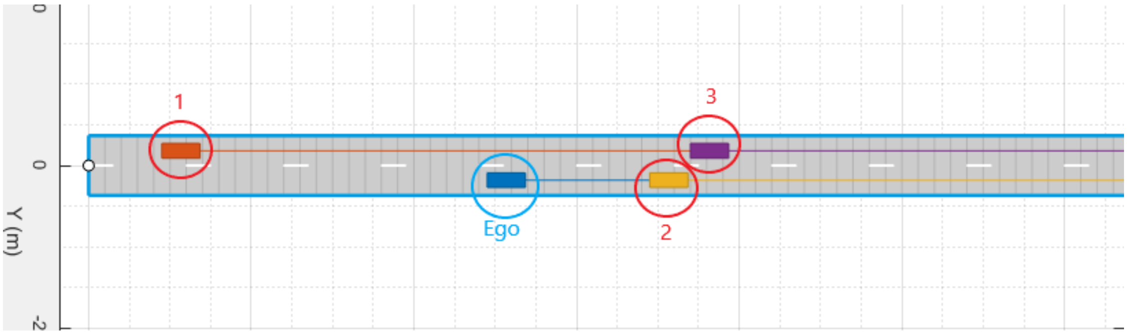

Figure 4.3: Design Scenario 1

vehicle	velocity	starting position
ego	x	[15, 0, 0]
v1	17	[7, -3.6, 0]
v2	17	[10, 3.6, 0]
v3	19	[31, -3.6, 0]
v4	15	[40, 0, 0]
v5	17	[50, 3.6, 0]
v6	15	[80, 0, 0]

Table 4.2: Scenario 2 target vehicles coordinates and velocity

4.2.2 Scenario 2

In this three-lane example, the vehicles executed four major lane change maneuvers. The first case is vehicle 1, which will change lanes from the left to the center between meters 480m and 511m. Then, vehicle 2 makes two lane changes, with the first occurring to the right and carried out between 110m and 144m, and the second, with left transition, to return to the leftmost lane between 204m and 234m. Vehicle 3 also makes two lane changes, initially turning left between 400m and 429m and finally returning to the rightmost lane at around 442m and 566m. The last element to change lanes is vehicle 5, which varied from the leftmost lane to the center of the road between 450m and 487m. For the CCRs case, two immovable vehicles are added: vehicle 7 in the right lane at 150m and Vehicle 8 in the right lane. Adjusting the predefined lanes to avoid collisions between these identical vehicles and obstacles was necessary. An example of this is vehicle 1, which starts in the leftmost lane and which, to avoid collision with vehicle 7, has to change lanes (between 99m and 130m) to the center of the highway road and then return to the right lane (between 222m and 246m), finally return to the central lane (between 525m and 548m). Vehicle 2 continues to make two lane changes, to the right and then left. However, the first occurs between 449m and 473m, while the second occurs between 497m and 516m. On the other hand, vehicle 3 passes to the left (between 100m and 129m) and finally returns to the right (between 160m and 185m). In the case of CCRb scenarios, vehicle one brakes (between 232m and 247m), vehicle three also brakes (between 240m and 254m), and finally, vehicle five also brakes (between 160m and 174m). Finally, vehicle 6 has to change lanes to the right (between 400m and 426m) to avoid a collision.

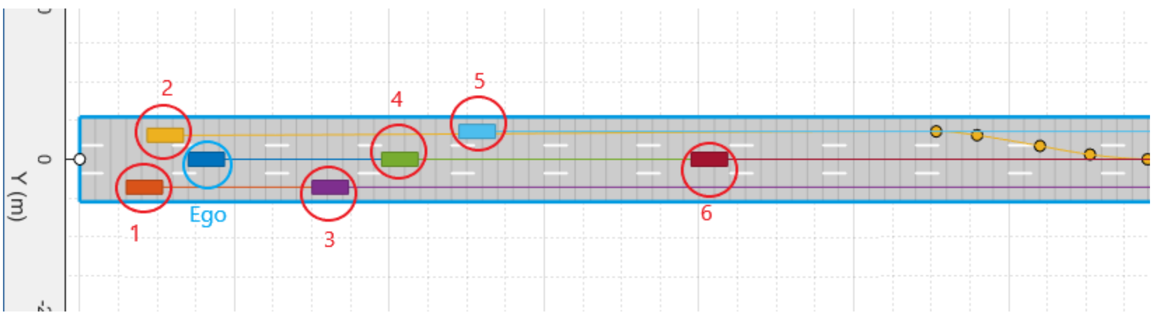

Figure 4.4: Design Scenario 2

vehicle	velocity	starting position
ego	x	[15, -5.4, 0]
v1	20	[1, 5.4, 0]
v2	17	[7, -1.8, 0]
v3	18	[13, 1.8, 0]
v4	21	[15, 5.4, 0]
v5	21	[30, 1.8, 0]
v6	24	[33, 5.4, 0]
v7	16	[50, -5.4, 0]
v8	18	[55, -1.8, 0]
v9	22	[100, 5.4, 0]

Table 4.3: Scenario 3 target vehicles coordinates and velocity

4.2.3 Scenario 3

This example consists of a four-lane highway. With lane changes, vehicle 1 turns to the right (between 260m and 288m), just like vehicle 4, which only makes the lane change between 340m and 370m. Vehicle 5 makes two lane changes, the first to the right (between 300m and 334m) and again to the right (between 400m and 425m), thus moving into the rightmost lane. Finally, vehicle 9 passes to the right between 520m and 549m. Regarding the CCRs category, vehicle 10 is added at the rightmost lane, where it is stationary 120m from the highway, and to avoid collision with it, vehicle 7 makes two lane changes, one to the left (between 92m and 116m) and the next to the right (between 128m and 160m), returned to the lane from which it started. In the CCRb tests, two cars are braking: vehicle 7 in the rightmost lane (between 210m and 223m) and vehicle 8 in the center-right lane (between 140m and 170m). To avoid collision between vehicles, car 2 has to make two lane changes to avoid an accident, the first to the left (110m and 136m) and then to the left (180m and 204m), returning to the initial lane.

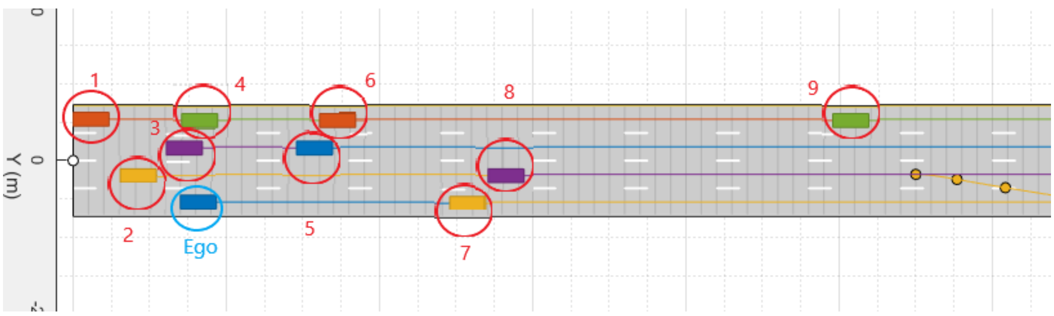

Figure 4.5: Design Scenario 3

4.2.4 Scenario 4

This example consists of one variable of scenario 3 with some changes in the initial positions of some vehicles' positions and longitudinal velocity, plus additional lane changes (this scenario has the highest number of lane changes and waypoints). Vehicle 1 changes lanes to the right (319m and 344m), and Vehicle 3 changes lanes to the left (260m and 292m). Vehicle 4 makes two turns, one to the left (between 220m and 256m), followed by one to the right (be-

vehicle	velocity	starting position
ego	x	[10, -5.4, 0]
v1	20	[1, 5.4, 0]
v2	17	[7, -1.8, 0]
v3	18	[13, 1.8, 0]
v4	22	[15, 5.4, 0]
v5	21	[30, 1.8, 0]
v6	24	[33, 5.4, 0]
v7	16	[50, 1.8, 0]
v8	16	[55, -1.8, 0]
v9	22	[100, 5.4, 0]

Table 4.4: Scenario 4 target vehicles coordinates and velocity

tween 283m and 307m), returning to the rightmost lane. On the other hand, vehicle 5 makes two turns, the first to the right (between 70m and 102m) and the second to the left (between 180m and 206m). Vehicle 6 makes only one turn to the right (between 120m and 153m), and finally, vehicle 8 also turns into the rightmost lane (between 71m and 101m). In the case of CCRs tests, a stationary obstacle (vehicle) is added in the central lane on the right, close to 210m. In addition, lane changes are added to avoid collisions, as is the case with vehicle 2, which has to turn into the rightmost lane between 179m and 204m to avoid collision with the immovable object. Finally, in the case of CCRb, it is vehicle 8 that makes an emergency braking after changing lanes close to 120m right in front of vehicle ego.

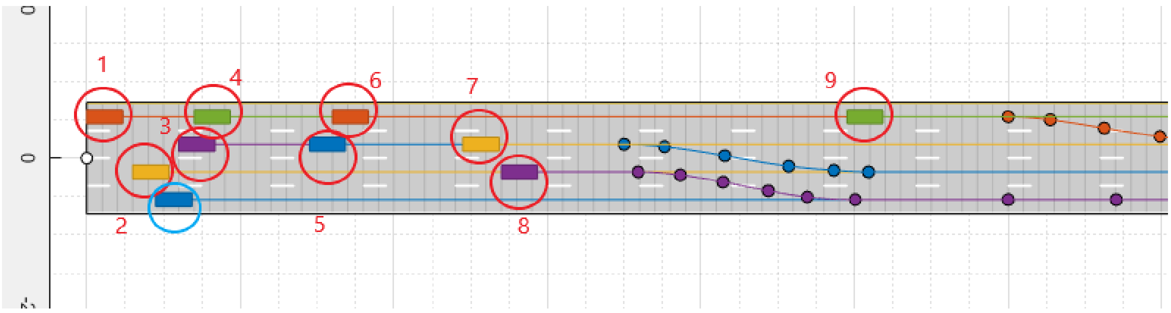

Figure 4.6: Design Scenario 4

4.2.5 Scenario 5

Scenario 5 is a variant of scenario 2 with the primary function of simulating a road accident. This accident occurred between vehicles 3 and 6 near 274m along the highway, where vehicle 3 lost control, ended up invading the center lane, and collided with vehicle 6, thus occupying two of the lanes, the central and the rightmost. Furthermore, vehicles 1 and 4 must brake to avoid collision with these cars, which are carried out between 240m and 258m. However, two cars managed to pass through the blockage caused by the accident. These are vehicle 5, which is in the leftmost lane and is not affected by accident, and vehicle 2, which changes lanes to the center (between 110m and 138m) and returns to the leftmost lane (between 200m and 234m) thus avoiding the blocked lanes. The primary goal of this test is to evaluate the behavior of the Ego vehicle when it comes across a collision. This scenario can only be adapted to one

vehicle	velocity	starting position
ego	x	[50, -1.8, 0]
v1	25	[35, -110, 0]
v2	16	[160, -1.8, 0]
v3	19	[70, 1.8, 0]

Table 4.5: Scenario 7 target vehicles velocity and coordinates

category, the CCRs, where the primary vehicle needs to react to two targets and suddenly becomes stationary.

4.2.6 Scenario 6

This scenario is a variant of scenario 3 with the primary function of simulating a road accident. In this case, the collision occurred between vehicles 5 and 8 close to 230m on the highway, blocking the two central lanes. Unlike the previous scenario, the cars behind traveling in the lane blocked by the colliding vehicles change lanes to avoid colliding with them. These being vehicles 2 and 3. Vehicle 1 turns to the right (260m and 288m), just like vehicle 9 passes to the right between 520m and 549m, and Vehicle 4 passes to the right between 340m and 370m. Finally, vehicle 2 turns right to avoid collision (between 200m and 227m). Vehicle 3 initially transitions to the left lane (between 208m and 226m), followed by a right turn returning to the initial lane (between 245m and 271m), that is, after passing through the area where the collided cars are located. The same logic of Scenario 5 is applied here, where this scenario can only fulfill the requirements of one category, the CCRs because the Ego vehicle needs to react to two targets and suddenly becomes stationary.

4.2.7 Scenario 7

This scenario is particular because it represents an entry ramp between a two-lane highway (like in Scenario 1) and a simple one-lane road. The secondary road starts at the coordinates [32.2, -111.8] and intersects the highway near the coordinates [195, 3.6] with the vehicle 2 leading the vehicle 1. However, vehicle 1 adjusts the lateral displacement according to the center line of the rightmost lane to enter the highway, and these curves occur between 193m and 220m. Furthermore, it also presents a higher longitudinal velocity than vehicle 2, so it needs to execute a lane change to the left to avoid collision (between 390m and 420m). For the category CCRs, vehicle 2 is stationary in the rightmost lane, close to 186m (before the entry ramp). In the category CCRb, vehicle 2 starts on position [76, -1.8], executes a slow brake process close to 166m, and ends up close to the entry ramp.

4.3 Test Cases

With each scenario, three central components vary significantly: scenario design (the number of driving lanes, traffic volume, and scenario complexity), the categories according to the European New Car Assessment Programme (EuroNCAP) standard, and the Ego vehicle set velocity. Between scenarios 1 to 4, the number of lanes increases from two lanes to four

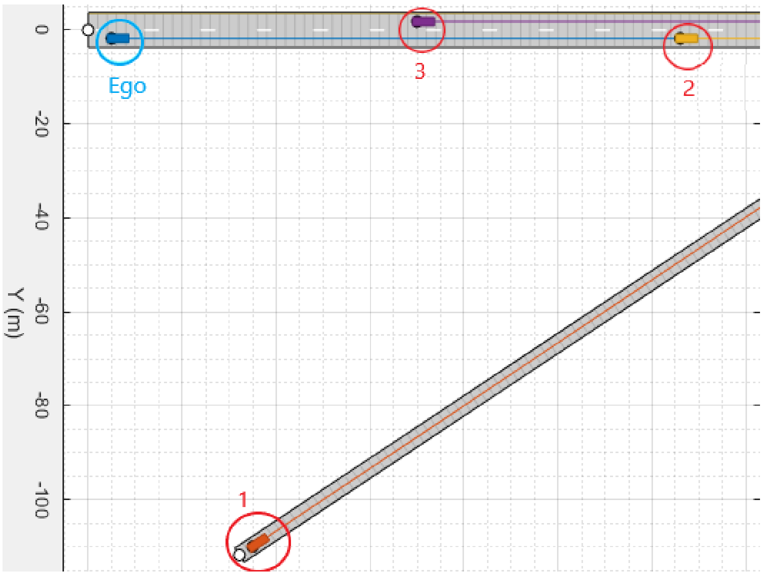

Figure 4.7: Design Scenario 7

lanes, with the number of cars also increasing from four (Ego vehicle inclusive) to nine and, in some categories, even reaching eleven (nine moving targets plus two stationary targets). Scenario 5 presents the lane drive and traffic volume of scenario 2, with the same happening between scenarios 6 and 3 and scenarios 7 and 1. Relatively to complexity, every scenario has a gradual increase in complexity and diversity, with every new map having more dynamic vehicles, making more lane changes, and being faster, with multiple possible paths and alternative routes for Ego vehicles and the Path Planner. The test cases for scenarios 5 and 6 allow us to test Ego behavior when a traffic accident occurs in front of the primary vehicle. Of course, there are some rules to these collisions. Even if they regard the problem of unpredictability of highway traffic, the Ego vehicle needs to have the minimum parameters to react to what happened. No cautious actions or evasive maneuvers can prevent a collision of Ego if the accident between two target cars occurs right in front of the Ego vehicle (a few meters in front). The collision can be mitigated by decelerating the vehicle (braking). However, collisions are unavoidable, so this type of test is not available in the spectrum of tests. Nonetheless, some tests off the record show a potential to drastically reduce the impact force, colliding with the front vehicle at a longitudinal velocity of $7m/s$ or less when applying a deceleration superior to $10m/s^2$. Finally, scenario 7 was created to test some dynamics in a prevalent real-life situation, especially in highway-traveled cities or major urban centers. With prolonged traffic in the lanes, it is close to the interchanges, intersections, and entry ramp of highways. However, in some cases, the velocity of accidents is ordinarily inferior to straight lines on the road, so they do not have a higher degree of representation in the test cases relative to the other scenario, only presenting two lanes and an entry ramp. The categories, according to EuroNCAP, as stated, are CCRm (rear collision with a moving vehicle), CCRs (rear collision with a stationary vehicle), and CCRb (rear collision with a braking vehicle). For each scenario from 1 to 4, all cases are tested for these three categories, where some target vehicle modifications need to be made to enforce this phenomenon. This is because, when trying a scenario for CCRs, we want to especially evaluate the vehicle response to a situation relative to the stationary front vehicle and not to other moving target vehicles. As such, only changes to the longitudinal velocity of some target vehicles must be made to force the car to handle these situations. Other changes are to avoid collision with the vehicle and how his speed changed, but this will be better explained when discussing the results. Scenarios 5 and 6 only are inserted in the CCRs category. Finally, the last component is the Ego vehicle initial or set longitudinal velocity, varying values 18m/s, 20m/s, 22m/s, 24m/s. Each test combines seven possible scenarios, each varying according to three categories and combining with four possible set longitudinal velocities for the Ego car. Adding some special cases that were made to test some exceptional cases, we created a set of seventy-five test cases.

4.4 Data from Test Visualization

The test case visualization is presented as shown in Figure 4.8. The screen is divided into two main windows: the chase view stands behind the Ego vehicle (third-person point of view), which follows all the vehicle movement, and a top view of the primary vehicle and all closed surroundings, including all target vehicles close. The top view perfectly interprets the surrounding environment, especially regarding cars behind the Ego vehicle. This is especially important to analyze the case where a target vehicle in the Ego lane travels behind the Ego vehicle at a superior velocity. This case is rare, but with the help of this point of view, it is possible to have a better understanding of the rear environment that can affect the Ego car. It also allows us to understand better the curvature taken by the vehicle from a two-axis perspective. However, the main point of view is the chasing camera, and this allows a natural sensation and perception of the car's steering motion and checks the collision capsules that influence the path-planning system. Every MIO has a group of capsules set in front of it. These are the capsules used in the collision evaluation process. When compared with the Ego vehicle's group of capsules, the trajectory is automatically invalid if they intercept at one given instant of time. From both points of view, we can see the trajectory generated by the Path Planning subsystem, where a dashed purple line represents all trajectories that violate the fifth polynomial curvature, the trajectories blocked by vehicles or that end up in a collision are represented by the continuous red line, the path generated with higher cost or not evaluated are shown as a straight continuous line. In the scenario shown, keeping the trajectory defined a few steps behind has an inferior cost to the other options, as the mechanism will be evaluated according to a sorted list of trajectories because the current trajectory is the best possible and satisfies all the parameters necessary, the other options are not taken in consideration. However, if that trajectory stops being possible (fails any parameters), it will automatically reevaluate all possible trajectories until it finds a new feasible one. However, because the current trajectory (continuous green line) satisfies all criteria, the cost of maintaining this trajectory will always be lower than the other options. In the scenario below 4.8, it is essential to mention that the Ego vehicle is decelerating to change the lane. Because of the proximity to the green car, the ego is forced to reduce velocity, even to change lanes, to confront the possibility of the lead vehicle suddenly starting to brake. However, because the velocity set by the ego vehicle is 24 m/s, it will change the lane for one where it can keep that parameter, so it switches to the left lane. Because the vehicle is still in the middle-right lane, it evaluates all possible trajectories to all adjacent lines, including the rightmost and center-left lanes. But the moment the vehicle's CG changes to the center-left lane, it will not start to generate possible trajectories for the leftmost and center-right lanes.

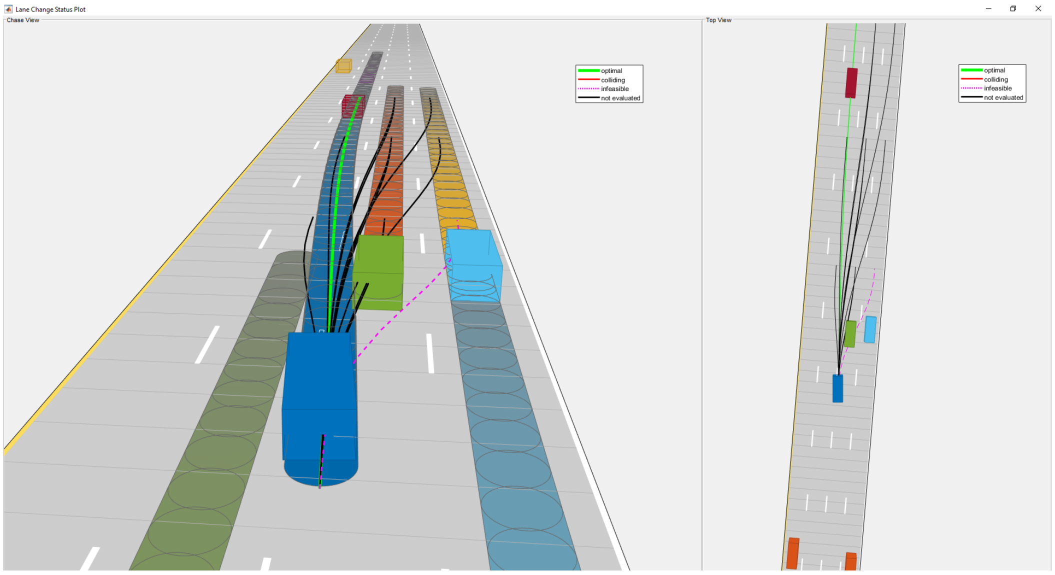

Figure 4.8: Data Vizualtion from test case CCRm of scenario 4 with Ego vehicle at $24m/s$ (CCRm4D)

One other example is the test case CCRb of scenario 7, where the vehicle is presented with a deadlock, where the front of Ego lane is blocked by vehicle 2, which is realizing a partial brake to allow vehicle 1 to enter the highway, and the adjacent lane is occupated by the non-safe target, car 3. With the current longitudinal of the Ego vehicle, all lane changes and maneuvers present the risk of collision when comparing the safety capsules. In this case, it will maintain the current lane because it has no suitable alternative trajectory. Due to the approximation between the Ego vehicle and the lead vehicle and the increase in collision risk, the AEB system will be activated, with all three deceleration stages, until it fully brakes approximately 4 meters from the leading car. This difference and revelation of this case is that, in this case, all alternative trajectories are evaluated by Path Planning because the optimal path is no longer acceptable. However, because all trajectories cannot be executed (due to collision or feasibility), the vehicle has no alternative but to stay in the current lane and let the AEB system realize its function. In all test cases, the activation of the AEB system can be graded in two central situations: the first is a partial brake during the driving process. However, it never reaches the state of a full stop (occurs specifically in the CCRm scenario). In this scenario, because all vehicles are moving at a constant longitudinal velocity, the Ego vehicle applying a partial brake allows a fraction of time to be gained during this velocity reduction. They are provoking the altering of the entire environment and the dynamics of the Ego relative to the target objects. In these cases, a simple slowdown at the beginning of the process will have a cascade effect, preventing the vehicle from finding itself in a collision situation. Simple caution actions are enough to avoid a future risk or dangerous situation. In contrast, the second situation happens when the Ego vehicle needs to apply a full brake during the driving process. This occurs when the car comes across deadlocks, which are situations where the lead vehicle is immobilized, in braking maneuvers or an aggressive process of deceleration (specifically in the CCRs and CCRm scenarios), and also all adjacent lanes are occupied by non-safe targets, which refers to cars that with current velocity and trajectory will collide with Ego vehicles if Ego decides to make any lane change maneuver. In some instances where the primary vehicle can apply all three braking stages gradually, it is possible to generate enough time for the highway environment to change or for the Path Planning to delineate a new feasible trajectory, but this is an exception to the rule. In most cases, the variation between the calculation of TTC in each sample time is much higher in the CCRS and CCRb scenarios than in the CCRm scenario. This higher decrease of TTC in specific scenarios means that the time that the Ego vehicles pass in every one of the three braking stages is inferior, with CCRs being the lowest, followed by CCRb cases and CCRm being the highest. This logic is intuitive because reaching a moving object will always take longer than getting an immobile object. Inverting this logic, we can understand why, in CCRm cases, we only have partial brakes, if necessary, any brake at all, while in CCRs and CCRb, in almost all cases, we end with a full brake.

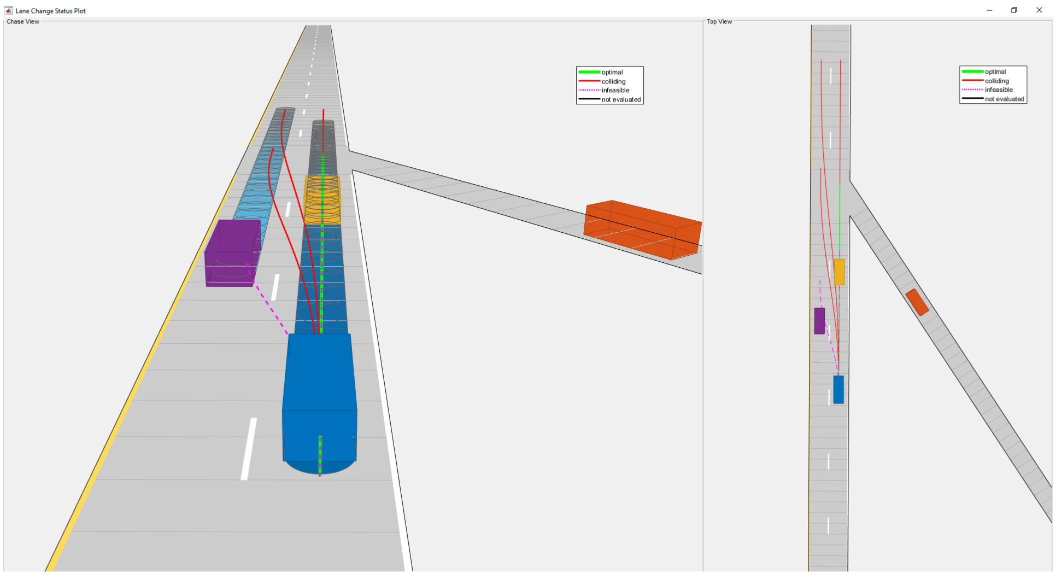

Figure 4.9: Data Vizualtion from test case CCRb of scenario 7 with Ego vehicle at $24m/s$ (CCRb7D)

4.5 Rating Criteria

As mentioned, seventy-five test scenarios were generated, combining seven scenarios with three categories and four sets of longitudinal velocity for the Ego vehicle. Every scenario was tested pre-development of the AEB, with some specific scenarios being modified to generate specific deadlock scenarios and braking situations. Seven of the twenty-one CCRm scenarios resulted in car collisions, eight of thirty-two CCRs (car collision scenarios included) resulted in car collisions, and seven of twenty CCRb resulted in vehicle collisions. Furthermore, two extra scenarios were added to represent a specific situation. These exceptional cases are variants of others built to test simple modifications in fascinating behaviors and scenario dynamics. The test occurred according to the hierarchy system, where a scenario can be

Category	Nº of Collisions	Nº of Test cases
CCRm	7	21
CCRs	8	32
CCRb	7	20
Especial	2	2

Table 4.6: Number of collisions for all test cases in each category

described by an abbreviation, presenting first the category, next the scenario, and ending with the Ego vehicle sets longitudinal velocity. For example, "CCRs6B" means we are talking of scenario 6 of the CCRs category with the Ego vehicle traveling a fixed velocity of 20m/s, where the "A," "B," "C," and "D" stand for the values of velocities of 18m/s, 20m/s, 22m/s, and 24m/s respectively. Finally, there is a case of a CCRs2B, which, with one alteration of the velocity of one vehicle, allowed the creation of a collision against a moving car during the test of the category CCRs. This enables the creation of a second CCRm2B, which presents the two stationary vehicles, cars 7 and 8 (the derivative of CCRs2B). When showing all cases according to their categories, I would like to point out that they will be brief, describing only some of the most relevant scenarios. However, some critical data will be presented, such as the time of collision of the test bench without AEB, the velocity changes made to some target vehicles, what type of stop occurred (if it was a complete stop with or without return to traveling in the highway), Ego performed lane changes, and some additional information extracted from the test case. In some scenarios, the vehicle realizes a complete brake, with all non-safe targets passing in the adjacent lanes. Sometimes, it returns to the traveling mode if the Ego can take a feasible trajectory to one adjacent lane clear of danger. This occurs due to the implementation of a second delay to allow the vehicle to reevaluate its path options before completely shutting the simulation. However, Ego's return to his traveling motion after a complete break and reaching an entire stationary state was not the aim of this project. However, to simplify the test cases, in most of them, whenever the car realizes a successful activation of the AEB system that completely stops the motion of the Ego vehicle, this will terminate the simulation.

4.6 Test Cases Results

All significant tests already designed for the standard test bench (without AEB) were maintained and used as the primary source of comparison and evaluation between the two systems. However, some improvements and adjustments were added to increase the dynamics of the environment on the highway. Some scenarios are designed to force the Ego vehicle to use the AEB process to avoid collisions. As a logic, these require some alteration from the standard scenarios, but they were restricted to manipulating the set velocity of some target vehicles. Furthermore, some immobile cars and objects are situated in critical moments according to the behavior pre-defined by already executed on the first round of tests without AEB, with a second round needed to adjust these vehicles' behavior to the scenario modification.

4.6.1 CCRm

First, we will analyze and compare all the results to the pre-defined scenario under the CCRm category. One particularity of this scenario is that all vehicles move along the highway with a set and constant longitudinal velocity. We also focus on all scenarios where the system without AEB will lead to a collision or a car accident. Comparing the performance of the AV system with and without AEB, the first premise that we can extract is that Ego vehicle behavior is similar between the two models. However, as expected, the vehicle's performance with AEB is slightly inferior due to the car having a more "cautious" approach to the highway environment. This phenomenon happens because the test bench with no AEB performs the purest optimal highway driving approach, dangerously discarding the proximity and assuming all other cars will perform their functions without significant trajectory variations like braking. In contrast, the test bench with AEB will start to act every time it reaches a dangerous proximity to the lead vehicle. This will increase the time the car requires to reach the end of the simulation, reducing his performance in the long term but increasing his road security and naturality of driving performance. In other words, without the AEB system, the Ego vehicle will prefer to maintain the set velocity established at the beginning of the simulation, with low values of deceleration, mainly produced by the ACC of the controller, with the primary function of preserving a small safe distance between the lead and Ego vehicles. However, as shown by the tests 4.7, when Ego is set with higher longitudinal velocity, simple deceleration measures applied by the ACC are not enough to prevent a collision, with a higher force of deceleration needed to avoid the accident. It is when the Ego presents a longitudinal velocity far superior to the lead car, and there are no available trajectories to change lanes. This occurs when there are unsafe targets (in proximity) on adjacent lanes to the Ego lane. As a logic, when the vehicle can't brake, there are no alternative maneuvers, and it's moving at a far superior velocity than the front vehicle, the collision is inevitable. When reaching an alarming longitudinal distance to the lead vehicle, the AEB will apply a partial brake, reducing the velocity to match the lead car (with the controller micro-managing that process) and wait for the circumstances on the highway to change, where the danger is reduced to a certain degree and new trajectories for changing lane being available. After avoiding the threat, it is delegated to the Controller or the recovery velocity subsystem to slowly approach the longitudinal velocity to the set velocity at the beginning of the simulation. As demonstrated by the previous example, the threshold of time that the vehicle waits for the highway environment to change and applying partial brakes will consequently cause the Ego to take longer to reach a certain point, in line with what was mentioned.

For scenario 1, there is a minor alteration or significant results. The general behavior is similar to and without AEB due to the simplicity of the environment. In the test case CCRm1A, both models conduct precisely the same way, with no dangerous situation; the vehicle doesn't need brakes, so the performance is the same. The first problem appears in the test case CCRm1B, where the car only needs to use a simple partial brake and automatically stop the collision. With the time provided by the partial brake, car 1 will pass and clear the left lane, which the Ego will use to make a lane change and overtake the lead vehicle 2. In contrast, in test cases CCRm1C and CCRm1D, Ego will apply partial brakes due to the proximity with

Identification	Collision	Time of Collision	Collision with AEB	Type of Brake
CCRm1A	X		X	X
CCRm1B	√	5.3	X	PB
CCRm1C	X		X	PB
CCRm1D	X		X	PB
CCRm2A	X		X	PB
CCRm2B	X		X	PB
CCRm2C	√	9.5	X	PB
CCRm2D	√	7.2	X	PB
CCRm3A	X		X	X
CCRm3B	X		X	PB
CCRm3C	X		X	PB
CCRm3D	√	5.9	X	PB
CCRm4A	X		X	X
CCRm4B	X		X	PB
CCRm4C	X		X	PB
CCRm4D	√	4.7	X	PB
CCRm7A	X		X	X
CCRm7B	X		X	X
CCRm7C	√	25.6	X	PB
CCRm7D	X		X	PB

Table 4.7: Collisions analyses for CCRm category

the lead vehicle, and when the left lane is without a non-safe target, Ego will change lanes. An important fact is that, without AEB, the system uses the assumption that the lead vehicle is constantly moving forward, so it has no regard for the possibility of the lead car abruptly stopping or even reducing velocity, which is not the case in this category but will happen with frequency on the other categories. This is why the necessity for full brakes increases drastically in the other test cases. If it is true that a vehicle can perform this test case without even braking, it is also true that the trajectory and risks taken are too much higher to be accepted. Presenting a priority on optimal behavior and trajectory at the expense of safety is not an acceptable tradeoff when human lives are at risk. This ethical problem will force the system to have a different set of priorities, where the optimal trajectory of Ego is now secondary to the ability of the vehicle to prevent or mitigate collisions. The scenario CCRm1B and CCRm1C need to have set longitudinal velocity chances to avoid a collision between car 1 and the rear of the Ego vehicle, which is not the scope of the tests. All these small changes in non-ego vehicles' longitudinal velocities are made to remove all types of collision that are not in the scope of this study. This study does not consider collisions with the Ego rear part or lateral, implying discard scenarios where non-ego vehicles realize a lane change that collides with the lateral of the Ego vehicle and all collisions between all non-ego vehicles.

In the CCRm2A, the behavior is similar, only activating a partial brake (stage one) when it is close to the first vehicle, then it changes to the far left lane. Next, it will calculated that eventually, it collides with vehicle 6, which is traveling at a lower velocity, forcing it to change to the center lane. In the center lane, we will be presented with a similar problem with vehicle 6, switching to the far right lane, where it will be until the end of the simulation. In these cases, it only needs to realize a small brake before the first lane change, but in test case CCRm1B, it needs to apply a small partial brake (stage one) three times before every single lane change. This occurs due to the increase in the overall longitudinal set velocity of the Ego vehicle, which now has less time and space between itself and every lead vehicle, so it needs to respect the safe distance in every single lane by applying constant small brakes to maintain a safe distance gap to the lead vehicle, but simultaneously, not reducing too much the distance gap to the following vehicle. In CCRm2C, because the Ego vehicle set velocity is superior to the previous one, it now requires a more drastic deceleration to avoid colliding with vehicle 4. It will change to the left lane and apply the recovery velocity (a constant acceleration boost to recover lost momentum). Still, when prepared to switch to the center lane, due to high velocity, it will apply the brakes again instead of changing lanes as in the previous test case, remaining in the leftmost lane, following the lead vehicle. Finally, in CCRm2D, the Ego will perform very similarly to the CCRm1B, making the exact lane change and operations. This is because the higher velocity of the Ego is counterbalanced with a more aggressive braking maneuver and a higher recovery velocity process. Once the vehicle passes the highly dense traffic, it can fully enjoy its superior set velocity. Still, after that, the vehicle behavior is identical to CCRm1B but with a higher velocity. In both CCRm2C and CCRm2D, the collision is avoided by applying a harsh partial brake (stage three), where both eventually ended up trapped behind vehicle 6, with car 5 occupying the left lane and car 3 occupying the right lane. Without a way to brake or an alternative lane to change, the test bench without AEB collides with the lead vehicle. These situations are avoided by the AEB, where in the CCRm2C test case, the initial brake inadvertently gains time to increase the distance gap between cars 5 and 3, then brakes once again and uses the ACC mode to follow vehicle 5. On the other hand, in the CCRm2D test case, because the Ego vehicle has a higher velocity recovery because it is set longitudinal value is superior, its force to change to the center lane because even with the distance gap gain by braking is not necessary to compensate the higher velocity require for the test case. This forces Ego to change to the middle lane, which has vehicle 6 with a low relative velocity, which also causes Ego to brake and change lanes to the rightmost lane. These successive partial brakes allowed the car to navigate the dense traffic without collision.

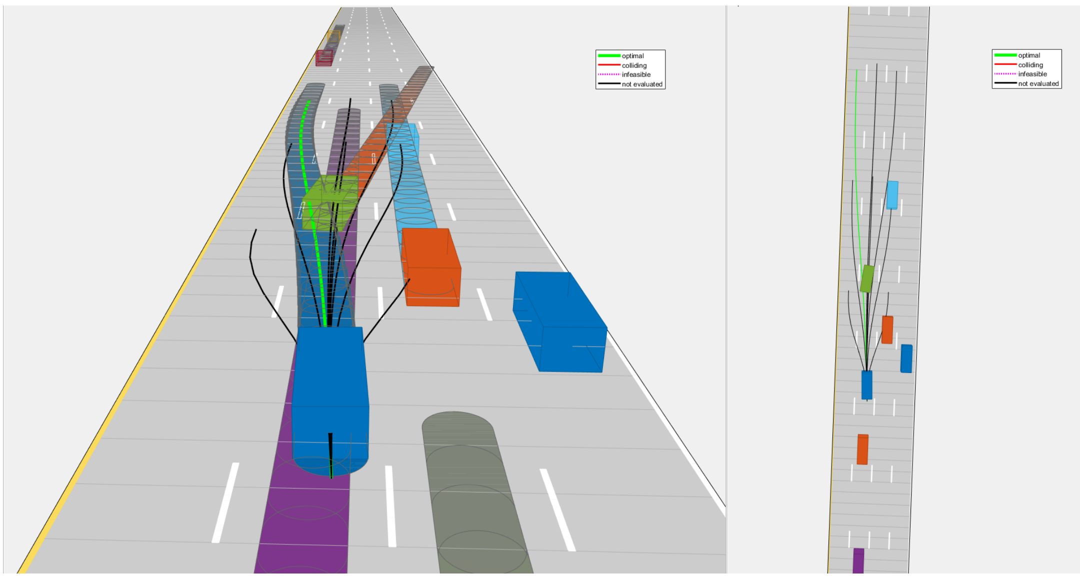

Figure 4.10: Lane Change in front of Ego vehicle CCRm3B

The CCRm3A is simple, with no variation between both models, no need for the AEB, and presenting an identical behavior. In contrast, in the CCRm3B test case, the vehicle starts in the rightmost lane and then changes to the left lane to avoid collision with lead vehicle 7. Next, it will follow car 8, which is moving at a slightly slower velocity, but only when Ego breaches the distance gap and vehicle 5 clears the left lane it opts to change to that lane. However, at this moment, car 4 also switches to the Ego lane, which is positioned right in front of the Ego vehicle. Due to this aggressive maneuver of the target vehicle, Ego decides to do a partial brake (reaching the lowest values of velocity and acceleration in this test) and position itself in the leftmost lane, where it will remain for the duration of the simulation. This maneuver is represented by Figure 4.10. Furthermore, all the essential data related to test cases in the course of the simulation can be seen in Figure 4.11, which shows the variation of the Ego longitudinal velocity, throttle (acceleration), yam angle, steering angle (radians), and AEB system activation. We can see in the graphics that the lowest values of deceleration and acceleration are directly related to the activation of the AEB, with the recovery velocity process being slowly applied after the braking maneuver. It's also possible to see all three left lane changes during the test case, with the following steering angle calibration to maintain the Ego vehicle in the lane center (LCA). The Ego vehicle never reaches its initial velocity due to the significant reduction of velocity used in consecutive left lane changes but slowly recovers the velocity until it gets an acetable value.

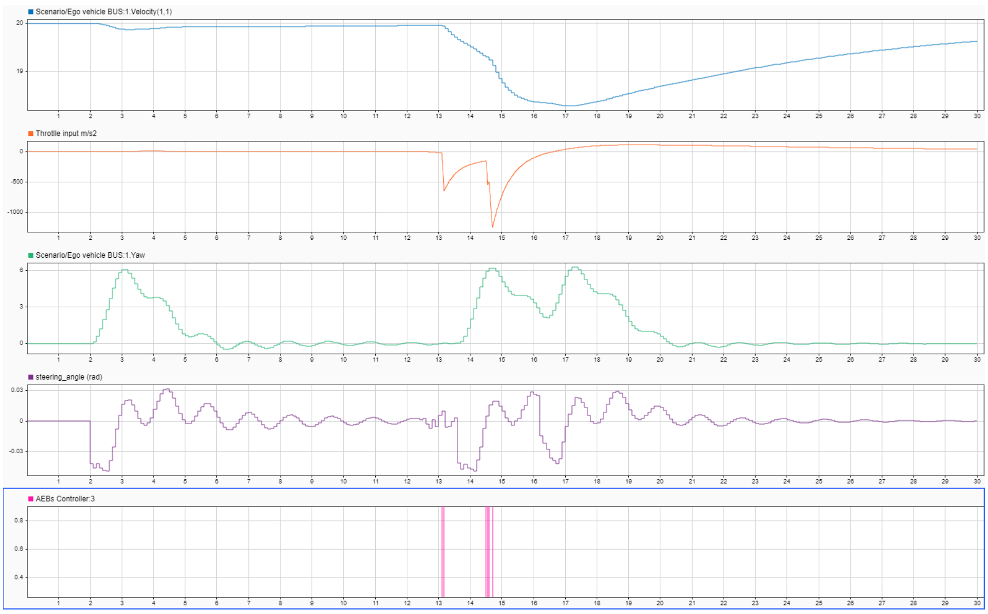

Figure 4.11: CCRm3B Data Inspector

The behavior in test case CCRm3C looks similar to three left lane changes. Still, because the velocity of the Ego vehicle is increased by $2m/s$, this is enough to create new scenarios with all new conditions. The car will start the braking process slightly early (due to increased velocity). Still, when it reaches a velocity more in line with the target vehicles in the surroundings, the behavior starts to be similar to the previous test case. As shown in Figure 4.12, we can see that the curves and shape of both velocity and acceleration are similar but occur very early in the simulation time scope, with only an increase of $2m/s^2$ between the two cases.

It still realizes three left lane changes, but now they are occurring very close to each other. When the Ego reaches the closing distance from vehicle 7, it changes to the middle-right lane in front of car 8, then changes to the middle-left and finds Vehicle 5 in the lead, and it is forced to change again to the leftmost lane on the highway. These maneuvers occur almost sequentially, with a low time gap between them, causing the Ego vehicle into a new, different scenario and behavior than its predecessor. In this scenario, car 4 is utterly indifferent to the outcome of any action. At the same time, in the CCRm3B, car 4 is pivotal to realizing the last lane changes because it needs to wait to pass by Ego, making this a notorious example of the "cascade" maneuver.

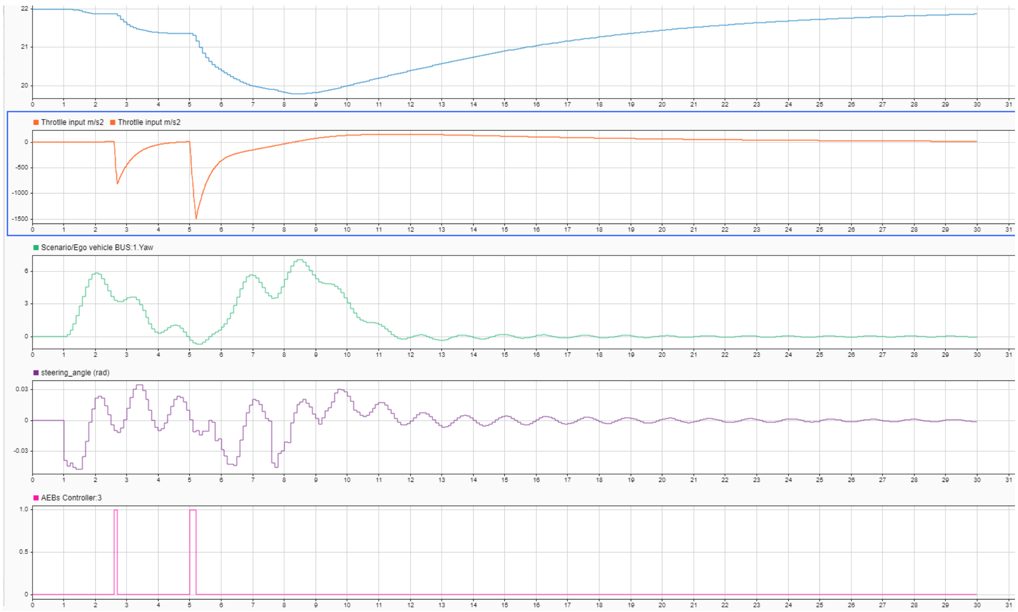

Figure 4.12: CCRm3C Data Inspector

Finally, for scenario 3, we have the CCRm3D test case, which has much in common with the previous test case. The slight difference is that after making the first left change, the Ego vehicle finds itself in a deadlock, with vehicle 8 blocking the Ego lane and vehicles 5 and 7 blocking the left and right lane, respectively. In a model without the AEB, the vehicle collides with the lead car. However, in this scenario, because of the AEB, the vehicle will perform a braking maneuver that will gain some time to clear the left lane, which the Ego will take advantage of to execute two consecutive left changes maneuvers until reaching the rightmost lane, which clear of any non-safe target. These simple and preventive brakes allow the vehicle to perform in highly dense traffic environments without colliding or significantly harming overall performance.
In the test case CCRm4A, there is no high variation and dynamism due to the Ego vehicle's low velocity, only making a left lane change when it realizes that car 8 has a lower velocity than it (breach velocity gap). This is another test case where AEB is not required, with both test benches performing the optimal trajectory. A similar situation will happen in test case CCRm4B. The AEB will perform two partial brakes to keep a cautionary distance gap from the lead vehicle 8, but otherwise, it still accomplishes the same objective without varying too much. Test case CCRm4C presents the same behavior, but in addition, the Ego vehicle needs

to brake with sightly superior force after changing to the middle-right lane, where it needs to wait for vehicle 5, which recently entered that lane, to move out of the lane, by switching to the rightmost lane and clearing the way for the Ego vehicle, which will prevail in that lane until the end. Finally, the case CCRm4D, where Ego finds itself in a deadlock after changing to the middle-right lane, with lead vehicle 5 blocking the Ego lane and vehicles 8 and 3 occupying the right and left lanes, respectively. The test bench without AEB collides with the lead vehicle. In contrast, in the model with AEB, when the Ego vehicle detects car 5 switching to the middle-right lane, dead ahead of Ego, forces it to apply an aggressive braking maneuver (stage 3) to avoid collision with car 5. This reduction in velocity is enough to simultaneously gain time to allow the lead vehicle 5 to change to the rightmost lane and open the lane for the Ego vehicle. Next, the Ego vehicle is forced to switch to the middle-left lane due to car 4 performing two consecutive left lane changes, ending right in front of the Ego. This maneuver is less dangerous than the one performed by vehicle 5, but it makes Ego change lanes again. It also requires a partial brake for the Ego to maintain a preventive safe distance from car 4 until the Ego changes to the middle-left lane, which remains until the end of the simulation. To finish all scenarios created for this category, we now analyze the four test cases of scenario 7. Starting with CCRm7A, in this case, due to the lowest velocity of the Ego, only the ACC mode is required because it doesn't end up closely reaching any non-safe target. In test CCRm7B, Ego is closer to vehicles 1 and 2 but never gets in a position requiring a partial brake. This changes in the case of CCRm7C, where first, the Ego vehicle needs to brake and switch lanes after vehicle 1 enters the highway. This occurs because the moment car 1 enters the highway is when the Ego starts to recognize it as a non-safe target, so it needs to change to the left lane, preferring following vehicle 3. Nonetheless, when all information about the new car is calculated, it determines that the velocity gap to car 1 is much higher than vehicle 3, making the return to the right lane a preferable choice. Pass forward sometime in the simulation, and Ego is trapped in a deadlock, with lead vehicle 2 having a far inferior velocity than Ego and car 3 blocking the left lane. This forces Ego to apply the brakes two times to adjust to the velocity of the lead vehicle. The model without AEB can't reduce the Ego velocity enough and collides with the lead vehicle 2. However, the AEB model can apply a braking maneuver to reach an equal velocity with the lead vehicle, not needing to brake fully but also capable of avoiding a collision. When car 3 finally clears the left lane, the Ego vehicle starts a lane change to the left, following the movement of car 3, which has a higher velocity than vehicle 2. The last case is CCRm7D, where it is needed to reduce the set velocity of car 1 to avoid a lateral collision with the Ego vehicle when entering the highway. This makes car 1 enter behind the ego vehicle, which puts some pressure on the ego to maintain a high velocity. The ego vehicle only applies a partial brake when getting close to car 2 and keeps a safe distance due to the velocity gap between vehicles. It then performs a left lane change and has the leftmost lane completely clear of any targets until the end of the simulation.

4.6.2 CCRs

This category has a drastic increase in the necessity to realize braking maneuvers. One particular situation is prevalent in this type of situation: the deadlock. When an immobile object

Identification	Collision	Time of Collision	Collision with AEB	Type of Brake
CCRs1A	X		X	X
CCRs1B	$\surd$	10.6	X	FB
CCRs1C	X		X	PB
CCRs1D	X		X	PB
CCRs2A	X		X	PB
CCRs2B	X		X	PB
CCRs2B	$\surd$	22.9	X	PB
CCRm2B	$\surd$	11.2	X	PB
CCRs2C	X		X	PB
CCRs2D	X		X	PB
CCRs3A	X		X	PB
CCRs3B	X		X	PB
CCRs3C	$\surd$	4.4	X	FB
CCRs3C	$\surd$	4.4	X	FB
CCRs3D	X		X	PB
CCRs4A	X		X	X
CCRs4B	$\surd$	9.6	X	FB & R
CCRs4C	X		X	FB & R
CCRs4D	X		X	PB
CCRs5A	X		X	FB
CCRs5B	$\surd$	12.7	X	FB
CCRs5C	X		X	PB
CCRs5D	X		X	PB
CCRs6A	$\surd$	11.8	X	FB
CCRs6A	X		X	X
CCRs6B	$\surd$	10.5	X	FB
CCRs6B	X		X	PB
CCRs6C	X		X	PB
CCRs6D	X		X	PB
CCRs7A	X		X	PB
CCRs7B	X		X	PB
CCRs7C	X		X	PB
CCRs7D	$\surd$	7.4	X	PB

Table 4.8: Collisions analyses for CCRs category

is present in the Ego lane, and the adjacent lanes are occupied by non-safe target vehicles, this will drastically reduce the freedom of moments and trajectories possibly generated by the Path Planning system. When the car is trapped in this impasse, there are no alternatives other than to execute a complete stop. We can see in table 4.8 that from nine collisions from the model without AEB, we can see that six of them will require the vehicle to perform a Full Brake until the Ego vehicle becomes entirely stationary to avoid colliding with the front car, while only in three cases a simple Partial Brake is enough to prevent the collision.

This test also includes scenarios 5 and 6, which refer to collisions of non-ego vehicles in the middle of the road. They are characterized as CCRs because they are already entirely stationary when the Ego collides with them. However, they still block two lanes, making it more challenging than the other scenarios in this category. They only present some objects blocking one road lane or two objects blocking two completely different lanes but with a considerable distance between them. In contrast, these two scenarios will block two simultaneous lanes, one adjacent to the other, making these scenarios a fascinating but hard-to-manage test case.

First, we start by analyzing the more straightforward cases in CCRs1A. In this test, Ego's vehicle tries to change lanes when detecting the proximity of car 1 in the adjacent lane, so it waits for the vehicle to pass him and clear the left lane, which Ego changes due to the proximity of the lead vehicle. However, when car 2 reaches the stationary vehicle occupying the right lane, it also changes to the left lane, positioning itself right in front of the Ego. However, because the velocities and distance are significantly low, this does not require using an AEB. So, both models have the same behavior. In contrast, test case CCRs1B could not be more different, where the model without the AEB collides with car 2 due to its inability to stop. With extra velocity, the Ego vehicle changes to the left lane. Still, because car 1 approaches Ego's rear at a superior velocity, Ego decides to return to the left lane and follow behind car 2. Nonetheless, the moment it chooses to do that is when car 2 changes to left to avoid collision with stationary vehicle 4. Now, it's the exact moment the dangerous object is visible to the Ego vehicle, and the model with AEB activates the most aggressive braking maneuver available to avoid the collision. In this scenario, the Ego changes left due to the velocity gap to car 2. It performs another right change due to the fast reduction of the distance gap to car 1, threatening an eventual rear collision with Ego. When Ego returns to the left, it abruptly presents with an immobile object as soon as car 2 changes lanes. It has no alternative than applying the full brake because the stationary target is only visible when the lead vehicle changes lanes. The CCRs1C and CCRs1D are similar due to the proximity of the Ego to the car2 at the start of the simulation, needing to brake to keep a safe distance gap immediately. Due to the differential velocity gap between the two vehicles, Ego performs a left change. Still, in this scenario, because the velocity of Ego and car 1 are similar, they never get too close to being considered dangerous. This is also possible due to the recovery velocity, which allows a more gradual recovery of velocity after the brake, avoiding any problem with car 2, the soon to be the following vehicle. This enables the maintenance of a constant and safe distance to lead car 1 and the rear car 2 after this change to the left lane.

In test CCRs2A, the ego vehicle starts braking at a certain point to reduce its velocity to match the velocity of vehicle 4, which is moving at a lower velocity. The Ego has no room for maneuvering with non-safe targets occupying all adjacent lanes, with changing lanes not being an alternative; this forces the Ego to reduce velocity until it reaches the lead vehicle velocity. In time, the leftmost lane is cleared of non-safe targets, and the Ego takes the opportunity to change to the left, where it remains in cruise mode until the end of the simulation. In CCRs2B, instead of following the lead vehicle, it will instead change to the left lane, taking advantage of its superior velocity, then will remain on the leftmost lane until starting to get close to lead vehicle 5, then will perform a double right lane change, until reaching the rightmost lane. This happens because that lane is entirely free of cars, and in the middle lane, where it's found, vehicle 6 is moving at a plodding speed. In contrast, the rightmost lane is entirely free of any car, so the path planning opts to go directly to the rightmost lane instead of maintaining itself in the middle and changing to the right after reaching car 6. With these double right lane changes, it can get a stable position to efficiently use the cruise control mode and drive without further complications. This test only does the braking maneuver in the early stage before performing the left lane change to maintain a safe distance between Ego and the lead vehicle. In the same scenario, CCRs2B, but with the velocity of vehicles 6 and 3 set at $17m/s$ and $20m/s$, we can create a scenario where the test bench without AEB collides with the immobile target. This occurs because the Ego's inability to apply the brakes in the early stages makes it execute the two first lanes change faster than the model with AEB. When Ego reaches the middle lane, vehicle 3 still occupies the rightmost lane, so Path Planning opts to switch to the left instead of the right. A few seconds later, Ego will find itself in a deadlock, with a stationary car in front and vehicle 6 in the adjacent lane, Ego collides. That fateful decision does not happen in the test bench with AEB because, with an early break, the Ego allows the simulation to evolve in a couple of seconds to a much more favorable situation, changing the entire dynamics of the highway. This shows that in very dense traffic environments, preventive action can have significant benefits in the long term. To prove the point that simple variations of $2m/s$ have in the highly-dense traffic environment, another exciting variation of this scenario (CCRm2B) was created by making vehicle 6 have a reduction of velocity to $14m/s$ and car 3 an increase to $20m/s$. In this case, the model without AEB can't perform any right lane change, keeping itself in the leftmost lane behind car 5, which will cause the Ego to collide with the rear of the lead vehicle. The model with AEB tries to perform a right lane change, but due to the distance gap to non-safe targets in the adjacent lane, it will switch to the leftmost lane instead, braking enough to adjust its velocity to match the lead vehicle 5. When an opportunity surges in the middle lane, Ego performs the double lane change to the right, ending in the rightmost lane, which is free of cars. In the test case CCRs2C, the vehicle completes the exact left lane change followed by two consecutive right lane changes like the Ego performance in the test case CCRs2B. The significant difference is that because the Ego vehicle travels at a much superior velocity, the braking maneuvers must be more aggressive. This is expected by a car with much superior velocity in an environment containing vehicles with inferior velocity. In the last CCRs2D case, the Ego vehicle, after performing the change to the leftmost lane, keeps in that lane for a couple of seconds until the adjacent lane (middle) is clear of non-safe targets, then performs a lane change to the right, and finally performs a lane change to access the leftmost lane which is clearly of vehicles until the ending of the simulation. This scenario has adjustments in the velocity of front cars due to the existence of two immobile objects at the leftmost and rightmost lanes, making many non-ego vehicles collide with each other. These adjustments aim to avoid unacceptable circumstances for these test cases.

Scenario 3 also presents some interesting test cases to study. First, in CCRs3A, when Ego reaches a close distance to the lead vehicle, it brakes to keep a safe distance, changes to the left lane, and remains in cruise control until the end of the simulation. This lane change also implies that the car inadvertently avoids the stationary vehicle. In the case of CCRs3B, the vehicle has the same behavior as the previous test case, but because it has a superior velocity, it tries to perform various lane changes. When vehicle 7 changes left to avoid collision with the stationary target, it cuts right in front of Ego, who tries to change to the adjacent left lane to avoid collision because it has a far superior velocity to the target car. However, car 7 quickly returns to the rightmost lane. Analyzing the new situation, Ego decides to maintain the current lane and cancels the process of changing lanes due to the adjacent lane's high traffic density. Further ahead, the Ego vehicle tries to move to the rightmost lane because of its lesser density of vehicles. Still, due to a rear car closing by, it is forced to return to the middle-right lane, which remains until the end of the simulation. In this simulation, Ego ends up canceling two lane changes due to the incredible dynamism of the environment with the very rapid shift in target vehicle displacement. The test case CCRs3C is divided into two test cases. The first case starts as the previous tests, with a brake followed by a left lane change, where it remains until a close distance gap with lead vehicle 8, and with the adjacent right lane with vehicle 7, Ego performs a lane change to the middle-left lane, where it will remains until the end of the simulation. All these maneuvers mentioned above of Ego vehicles can be seen in Figure 4.13. However, the other variant could not be more different. Having car 2 velocity increase to $23m/s$, the first lane will change to the middle-left lane is impossible because this target vehicle is blocking that change, which forces Ego to continue in the current lane, which will make Ego directly facing the stationary object, after vehicle 7 change to the left lane. This situation causes the model without AEB to collide with the vehicle while the model with AEB performs an entire brake maneuver to avoid collision. Finally, CCRs3D will act like the first variant of CCRs3C because the first brake process in this scenario will be harsher, leaving Ego at a similar velocity to the previous test. However, when the lanes start becoming clear of traffic, the recovery velocity process is activated, and it starts to gain velocity, reaching the end of the simulation much faster than its predecessor.

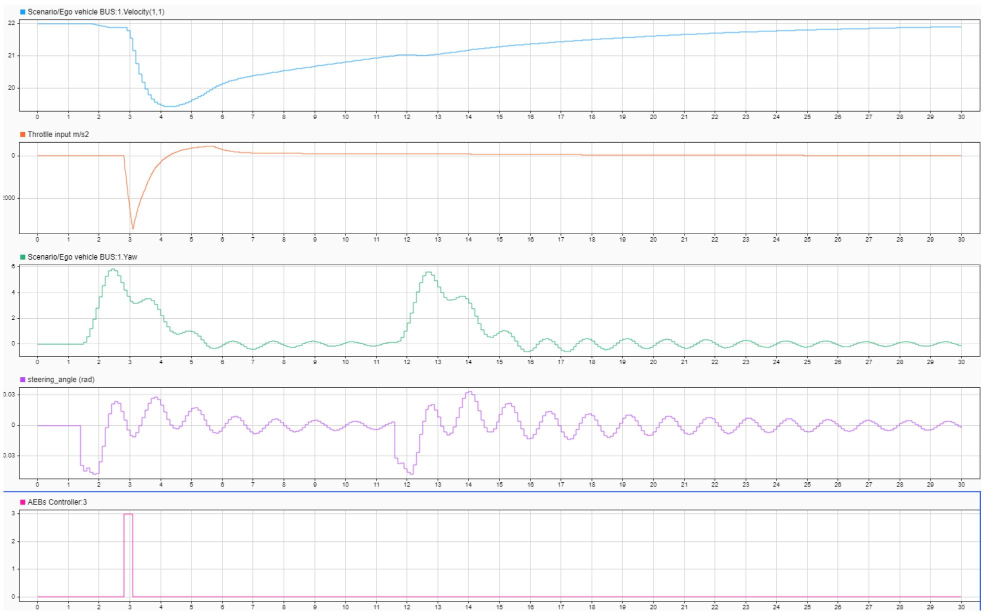

Figure 4.13: CCRs3C first variant Data Inspector

The next group of tests starts with CCRs4A. Ego starts in the rightmost lane, following the lead vehicle for a couple of seconds, but due to the eventual proximity to the lead vehicle, Ego starts a lane change to the left. This is when something exciting happens: the current target vehicle in that adjacent lane, car 5, suddenly executes a lane change to the right, making the front vehicle (car 10) the lead vehicle in that lane. The problem is that the new lead vehicle is stationary, forcing the Ego to cancel the lane change and maintain the current lane, following the lead vehicle 8. However, when Ego passes through the side of the immobile target, it will perform a change lane to the left, entirely free of cars. In this scenario, it wasn't necessary to activate the AEB system, so both models act the same, but things could not be more different for the other cases. In test CCRs4B, the Ego vehicle changes to the left lane due to the lead vehicle 8, but when it does that, the lead vehicle is the stationary car. The Ego vehicle is forced to brake with all two adjacent lanes occupied with non-safe targets. One detail is that the distance to the lead vehicle is still considerable, so Ego has time to decrease the velocity slowly. This allows the two non-safe targets to clear the rightmost lane, which Ego tries to use to overtake the car. However, there is still a vehicle coming for the rear, which forces the primary vehicle to brake again to reduce the velocity, allowing the target car to pass, and after that, it has the lane free to use. After applying a complete brake, the vehicle starts to recover velocity, changes to the middle-right lane (lane with fewer vehicles), and remains there until the end of the simulation. Test CCRs4C has two very distinct behaviors depending on the model. The model with AEB, due to brakes to maintain a safe distance with lead car 8 when changing to the left lane, finds itself in a similar situation to the previous test case. The only difference is that due to the extra velocity of Ego, this one prefers to change to the left lane, where it remains until the end of the simulation, where Ego takes advantage of the longitudinal momentum to insert itself into the adjacent lane. When using a model without AEB, since the vehicle can not brake, it can reach the stationary target much faster, then finds a situation when it can simply turn to the left lane ahead of vehicle 7 (in contrast with the model with AEB that goes behind car 7). This seems like an optimal solution at first glance. The problem is that when we analyze the overall behavior of the vehicle, we can see that in the trajectory to overtake the immobile vehicle, the ego vehicle has to pass at a very close range from the other vehicle. The distance between the vehicles when Ego is passing the car is so close that to the human eye, it is possible to think they collide, as shown in Figure 4.14. They are so close to each other that this behavior is unacceptable in the real world, with too high a risk for human lives. Finally, in CCRs4D, Ego performs similarly to the optimal trajectory. It does a lane change to the left, and when it faces the stationary vehicle, it executes two lane changes to the left, positioning itself in the leftmost lane. The realization of these lane changes occurs to avoid the stationary vehicle first and then the lead vehicle 7 in the middle-left lane. Because car 7 has an inferior and is right in front of Ego, it forces the primary vehicle to change again to the left to avoid a collision. Then, it remains there until the end of the simulation. The Ego will perform three brakes before every lane change to maintain the safe distance gap between vehicles.

Now for the extraordinary cases simulating a car crash between two non-ego vehicles in the middle of the simulation. The first case, CCRs5A, starts with a change to the left lane due to the breach of the safety distance with the lead vehicle 4. Ego follows the new lead vehicle 5, but in time, it will also violate the safety distance due to the velocity gap between them. However, the accident between the two cars happens when the Ego contemplates changing to the adjacent lane. With a fast steering maneuver, Ego cancels the maneuvers and returns to the current lane (leftmost) as fast as possible. As soon as it passes by the accident zone, the Ego vehicle changes to the middle lane to avoid vehicle 5 (the leading vehicle that passes

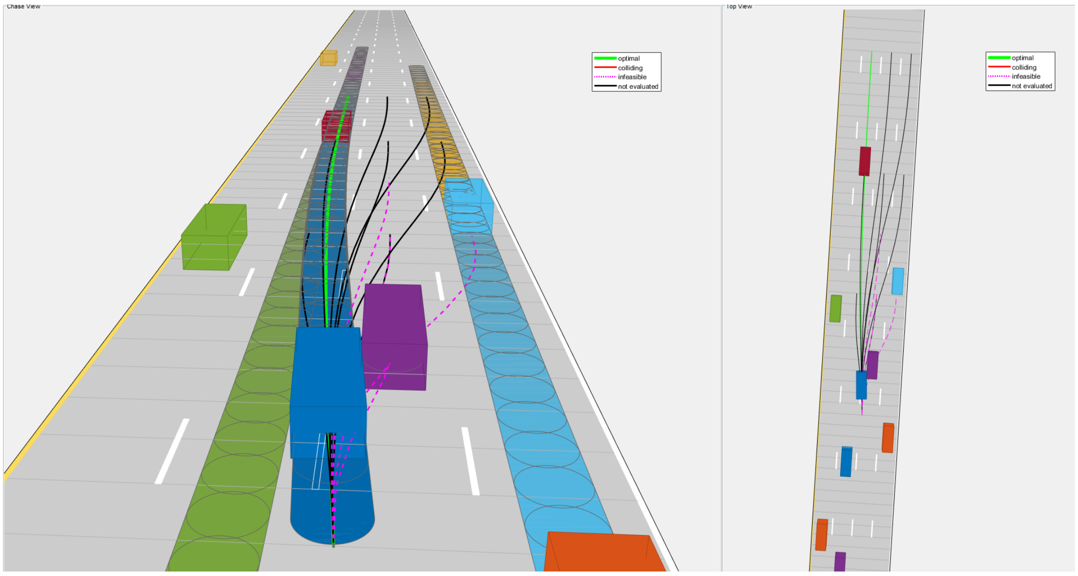

Figure 4.14: Model without AEB on CCRs4C

in the left lane adjacent to the accident zone), where Ego will remain until the end of the simulation. However, in the model with AEB, the behavior is different because of the Ego braking procedure due to the distance gap infringement, which prevents the risk of collision by reducing its velocity to match the lead vehicle. Still, when threatened by the proximity of the rear collision with vehicle 2, Ego is forced to change to the right lane, which makes following one of the cars that will collide. The Ego must apply a complete brake maneuver to avoid colliding with the crash cars (entirely stationary and blocking the right and middle lanes). This is the first scenario where the presence of AEB will lead the vehicle to a less desirable outcome. Prioritizing safe distance and reducing velocity, the Ego will find itself in a precarious position where it is forced to brake fully. In contrast, without AEB, it only needed a more aggressive maneuver and was entirely out of danger. The positive side is that the Ego vehicle didn't collide with any car in any of the two situations. Scenario CCRs5B resembles CCRs2A, with Ego performing a brake to keep a safe distance, changing to the leftmost lane, and executing two lane changes to the right, ending up in the rightmost lanes. This decision happens because the leading vehicle in that lane has a superior velocity than all vehicles in all three lanes. The problem with this decision is that car 3 is in a crash collision shortly. When that collision happens, the Ego has no alternative than full brake until it fully immobilizes the viature. In the model without AEB, the Ego collides with vehicle 3. Test cases CCRs5C and CCRs5D because they have a much superior velocity than all other vehicles. They execute both the same trajectory. First, they brake to maintain a safe distance and change to the left lane, where they will brake again to reduce the velocity and maintain proximity with lead car 5, which happens because vehicle 6, with a low velocity, is blocking the middle lane. After passing the accident zone, Ego changes to the middle lane, which finds lanes entirely free of vehicles.

The next group of test cases is more complex than the previous. Starting with CCRs6A, the Ego vehicle executes a lane change to the left to avoid collision with the front vehicle 7. In that lane, the Ego will follow the lead of vehicle 8, which will collide with car 5. The problem is that when this accident happens, all the adjacent lanes of Ego are occupied by non-safe targets, including vehicle 7 in the right lane and vehicle 3 in the left lane. Because Ego is trapped in the current lane, with no alternative trajectory to avoid the cars that will collide, the primary car is forced to apply a full brake to immobilize the vehicle before the collision. In other versions of this test case, with increased velocity for vehicles 7 and 2, the velocity was passed to $17m/s$ and $18m/s$, respectively. This simple alteration meant that car 2 would disable Ego's ability to turn left by blocking the adjacent left lane, and now Ego follows vehicle 7, which would increase velocity. Not making that left turn implies that Ego maintains in the leftmost lane, avoiding completely being in the lanes where the car crash will happen. After passing from the accident, the Ego turns left to the middle-left lane, which will remain until the end. There is no need for any actuation of the AEB because the deceleration provided by the ACC of the Controller is enough to reduce the velocity to a safe velocity gap between the Ego and the lead car. In test CCRs6B, we can divide this into two sub-tests, with both performing precisely as the previous test. In the standard test, because the adjacent right lane is occupied by car 7, the Ego will change to the left lane after a couple of seconds and is confronted with the collision. Even with the left adjoining lane clear because the accident occurred in two lanes in the middle, changing to any of those lanes will have the same outcome. This is because even if Ego changes to the left lane, it still finds the other car that took part in the collision completely stationary. The only viable alternative is braking until the Ego vehicle stops completely. As shown in Figure 4.15, we can see the Ego vehicle applying the brakes a few seconds before it entirely stops in front of the accident zone, more important we can see the overall performance of the Ego during the entire test in Figure 4.16.

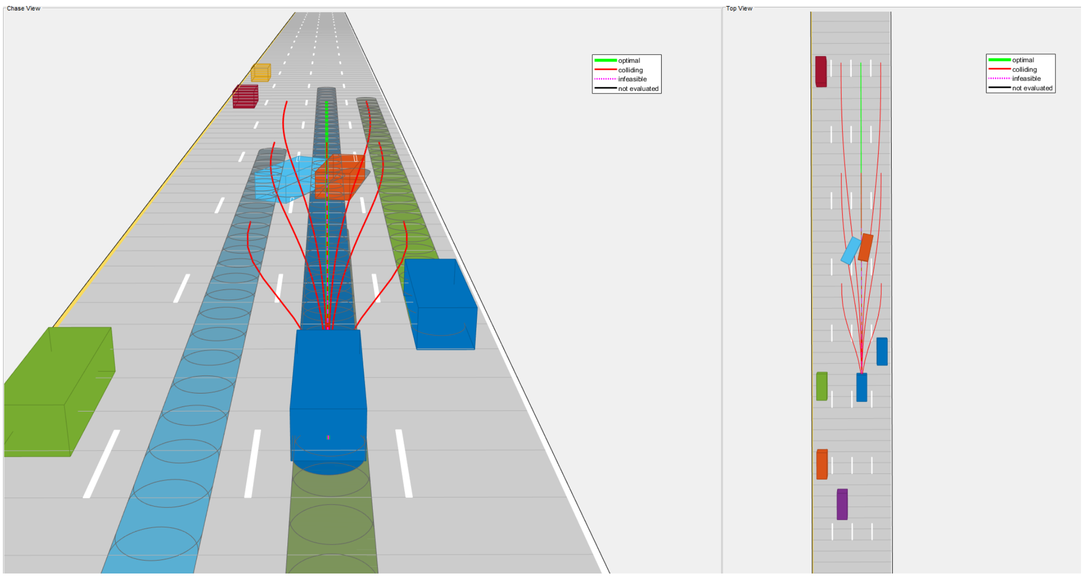

Figure 4.15: First sub-test CCRs6B, applying the bakes

If we increase the velocity of car 7 to $20m/s$, it is enough to retain Ego in the rightmost lane until passing the car crashing area, and after that, Ego changes to the middle-right lane, where it remains until the end. In the case of CCRs6C, the primary vehicle can overtake the lead vehicle because the Ego velocity is significantly higher than that of vehicle 7. This means that the Ego vehicle will perform a lane change to the left, and then, due to the low velocity of

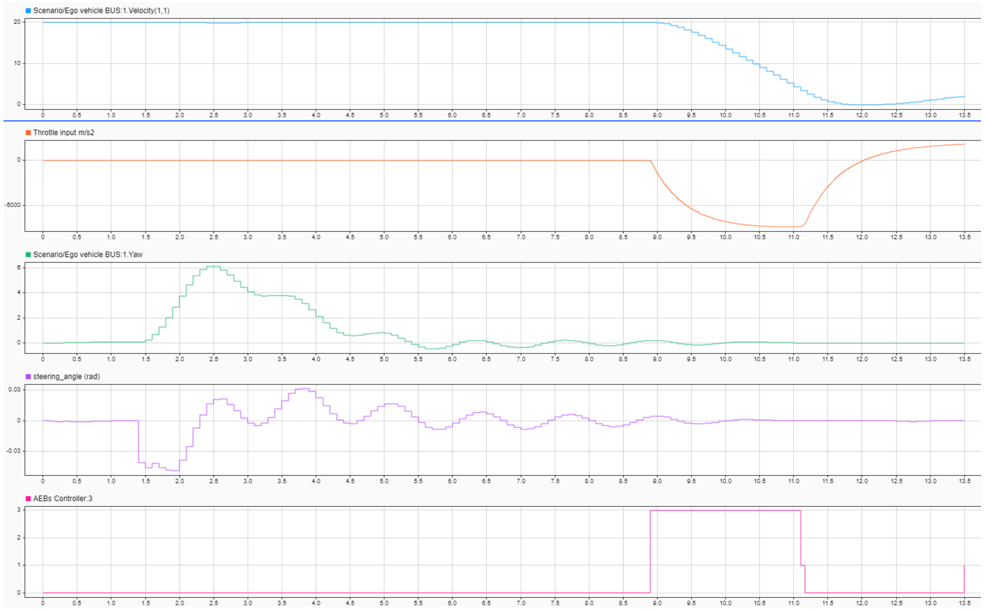

Figure 4.16: First sub-test CCRs6B Data Inspector

car 8, Ego will turn right and return to the rightmost lane, taking the lead over vehicle 7. With this maneuver, Ego can avoid the middle lanes where the collision will occur, and because it takes the lead over car 7, it now has no more vehicles in front. Ego maintains the current lane until the of the simulation. In the test case CCRs6D, Ego makes three consecutively changes to the left lane (braking before each one of them to keep a safe distance) and because it has a much superior velocity, is always trying to pass all vehicles, including all vehicles that will participate in the car crashing. When performing the third lane change, Ego vehicle lateral displacement is enough to avoid a collision with vehicle 5 (which has collided), using the middle between the leftmost lane and the middle-left lane. When using the leftmost lane, the Ego has to consider both rear vehicles, 4 and 1, forcing the Ego to adjust to their movement rapidly. As soon as it clears the car accident area, the Ego performs two consecutive lane changes to the right, where it will remain (middle-right lane) until the end of the simulation.

To end this category, only four test cases represent entry roads into the highway. In the case of CCRs7A, the Ego vehicle starts in cruise mode until it comes close to the entry road, where car 2 can be found entirely stationary. To avoid a collision, the Ego vehicle starts to apply a small brake and change to the left lane. Where it remains until the end of the simulation, this test is straightforward. In the case of CCRs7B, the vehicle will present precisely the same behavior because it brakes before changing lanes to the left, where it will not find any car. The CCRs7C test can look similar to the previous one, but this has one fascinating detail. When the Ego starts braking before making the left turn, this simple action allows car 3 to gain some distance from the Ego, which means that when the Ego changes to the left lane, there is some safety distance between it and the lead car. This decision will make the Ego of the test bench with AEB execute a safer and more cautious trajectory. At the same time, in the model without AEB, the ego runs a collision-free trajectory with a low distance gap that is considered risky to be acceptable in ordinary situations. This is very important, especially considering the following test case, CCRs7D. In this test, Ego reaches his maximum velocity

Identification	Collision	Time of Collision	Collision with AEB	Type of Brake
CCRb1A	√	5.9	X	FB
CCRb1B	X		X	PB
CCRb1C	X		X	PB
CCRb1D	X		X	PB
CCRb2A	√	8.7	X	FB
CCRb2B	X		X	FB
CCRb2C	X		X	FB
CCRb2D	X		X	FB
CCRb3A	X		X	FB & R
CCRb3B	√	6.8	X	FB
CCRb3C	√	6.1	X	FB
CCRb3D	X		X	PB
CCRb4A	√	7.6	X	FB
CCRb4B	X		X	PB
CCRb4C	X		X	PB
CCRb4D	√	4.9	X	FB
CCRb7A	X		X	X
CCRb7B	X		X	X
CCRb7C	X		X	PB
CCRb7D	√	7.1	X	FB

Table 4.9: Collisions analyses for CCRb category

in these simulations, and when approaching the stationary, the left lane is still occupied by vehicle 3. With no possibility of changing lanes, the model without AEB will inevitably collide with the fixed target. However, in the model with AEB, when it starts braking due to a risky distance gap, Ego allows the highway environment to change, and vehicle 3 will consequently clear the left lane, which the primary vehicle takes advantage of, switching to the left lane and avoid any collision. By reducing the velocity, the Ego will position itself where, as soon as the left lane is clear, it creates an opportunity for the Ego to change lanes and avoid the necessity to perform a complete brake maneuver. At the end of the simulation, Ego turns to the right lane due to the inferior velocity of the lead car 3 in the left lane, where it will be until the end.

4.6.3 CCRb

Finally, the last category of test cases involves all target vehicle braking maneuver tests. An interesting fact about this category is that the total stops accomplished for every test case are far superior to the previous category, as shown in Figure 4.9. The main reason is that the braking maneuver by target vehicles creates a false sense of security. In the other category, movement is constant even when the car is stationary (in that case, it is zero). However, in this case, the target longitudinal velocity is dynamic; it still obeys a constant deceleration pattern but is entirely different from all other targets that move at a constant velocity. This drastic shift in velocity creates an altogether new problem for the future prediction motion of MIOs because to achieve reduced computational cost, the system assumes that all targets will maintain the current velocity for the near future. This premise is true in most cases, with a lower velocity variation, but it is not valid when the lead vehicle starts reducing velocity drastically (braking). But we will analyze any test case individually.

In CCRb1A, when a vehicle reaches a particular distance gap from the lead vehicle 2, it tries to change to the left lane. Unfortunately, that lane is occupied by vehicle 1, which is approaching at high speed, making that left turn impossible. As such, the Ego needs to remain in the right lane, which implies handling a lead vehicle starting to brake. This prompts Ego also to brake to avoid a collision, which the model with AEB does with success, but the model without AEB can't do. In CCRb1B, the Ego velocity is fast enough to mitigate the velocity of vehicle 1. This means that Ego starts to brake according to the velocity of the lead vehicle, changes to the left lane, and remains until the end of the simulation. A similar thing happens in test CCRb3A, but in this case, after switching to the left lane, Ego determines that the velocity gap relative to the new lead vehicle (car 3) is not recommended, so it needs to change lanes to the right lane. It is important to note that after the first brake in cases like that, the recovery velocity applied in the Ego vehicle after it changes lanes is vital to maintain a safe distance relative to the rear car 1. It is as essential to keep a front-safe as maintaining a rear-safe distance. The test CCRb1D is a combination of CCRb1B and CCRb1C because, when the starts braking before the left turn, the deceleration is higher than the previous test, which provokes a higher reduction in the velocity (making it like CCRb1B). When positioned in the left lane, Ego slowly recovers velocity and follows the lead car 3. However, as soon as it regains the full set longitudinal velocity, this makes car 3 a non-safe target, which forces Ego to change to the right lane (CCRb1C). In these three last test cases, the Ego only applies the AEB system at the beginning of the simulation to keep a safe distance from vehicle 2.

In more complex test cases like CCRb2A, Ego performs a simple brake to maintain a safe distance from lead car 4 immediately, turning to the leftmost lane and following now vehicle 5. The problem happens when the new lead vehicle 5 suddenly starts performing an emergency brake, and with car 4 occupying the only adjacent lane available, the Ego vehicle also brakes to avoid a collision with the lead vehicle. This action is impossible for the model without AEB, making the Ego collide with the lead vehicle while the target car still performs the braking maneuver. In the case of CCRb2B, the initial performance is similar, with a starting brake maneuver followed by a lane change to the leftmost lane, where it remains with a similar impasse as the previous test case (CCRb2A). The Ego is forced to full brake to avoid a collision, showing an inherent problem when playing safe driving over optimal driving. Because Ego is forced to brake in the early stages of the simulation to maintain a safe distance, sometimes it allows the vehicle to avoid future problems or dangerous situations (which was already shown in some tests). Still, it also can mean that the Ego will lose opportunity windows to avoid situations where it doesn't need to break. In this case, the rule is less evident than in the subsequent instances because the model without AEB needs to perform a very aggressive takeover over vehicle 5 to avoid collisions. This means a very aggressive shift of the steering angle and yam rotation with Ego passing too close by other cars to make this model a good or even an acceptable solution, nevertheless possible to be executed. This stops being a problem with the increase of Ego velocity in the following cases. In CCRb2C, due to the increase in velocity, the Ego vehicle decides to go right instead of left after a small break, even with vehicle 3 (in the adjacent right lane) being closer to Ego than car 5. Vehicle 3 has a higher longitudinal velocity, making its velocity gap to Ego far superior to its counterpart's, making it a better option as a lead vehicle. The problem in this test is when Ego starts following the new lead car 3, this target car starts making a braking maneuver, and with a vehicle in the adjacent left lane, the Ego brakes to avoid collision. However, in the model without AEB, this problem does not happen because the model can't execute any braking action. This means the lane changes happen much faster than the counterpart, making it reach the lead car 3 much quicker and start a lane change to the left before even vehicle 3 brakes. Figures 4.17 and 4.17 show that the environments that Ego finds itself at a determined time, using the model with AEB or without AEB, are drastically different because the small brakings maneuver will have a cascade effect in the distance traveled over time. We can see that at the exact 10.1 seconds, the Ego finds two very distinct situations, which shows the traveling opportunity lost by the Ego vehicle by being too cautious. When the non-safe target starts braking, the Ego vehicle finishes its lane change to the left, with aggressive curvatures and shallow distance gaps to non-safe targets. Still, its bold and proactive actions are rewarded. On the other hand, in the exact instant where the model without AEB is changing to the right lane, the model with AEB still follows the lead car 3 at a safe distance. This again shows the idea of opportunity lost when making a run in a test scenario without braking fully. Both tests can end the simulation without collision (the primary goal of this study). Still, without AEB, the model throws proactive action and can bypass all the scenario challenges without braking. Is a more proactive solution preferred to a safer execution when human lives are at risk? No, but it shows interesting questions and debates about further system optimizations and the significant subjectivity of this type of test in this environment before trying real-world tests. An interesting thing happens in the test CCRb2D, which will behave more like the test case CCRb2B. Because Ego has a higher velocity than the test case CCRb2C, it can no longer considered an alternative to change to the right lane because the speed of car 5 is too low to make that maneuver possible, so Ego only has one alternative, and that is to turn left. Both models will present an overall identical performance. In the model with AEB, the braking and recovery velocity processes will be higher than the previous test, and the model without AEB will show a more aggressive and proactive trajectory to compensate for the increase of Ego set velocity.

In the case of CCRb3A, the vehicle starts in the rightmost lane due to proximity to lead vehicle 7. The ego changes to the middle right lane, traveling behind the lead vehicle 8, which after some time starts braking maneuver. The model with AEB recognizing the danger immediately starts braking, trying to keep a safe distance. With vehicle 2 approaching Ego, traveling on the rightmost lane, Ego waits until that lane is clear of any non-safe target to continue his driving process. It changes to the right lane, and after passing the dangerous zone, it makes two consecutive left turns to the middle-left lane (lane without vehicles), where it remains until the end. In contrast, because the model without AEB can't brake, the only choice is to perform a very tight curvature to avoid the immobile object and pass in front of car 7. This trajectory is very aggressive, with Ego passing between a moving object and a stationary one, with a dangerous maneuver. Nevertheless, the model can perform execute the test without collision. After that, the vehicle completes a double lane change to the left, where it remains until the end. In CCRb3B, the Ego turns left to avoid vehicle 7 and keeps driving with vehicle

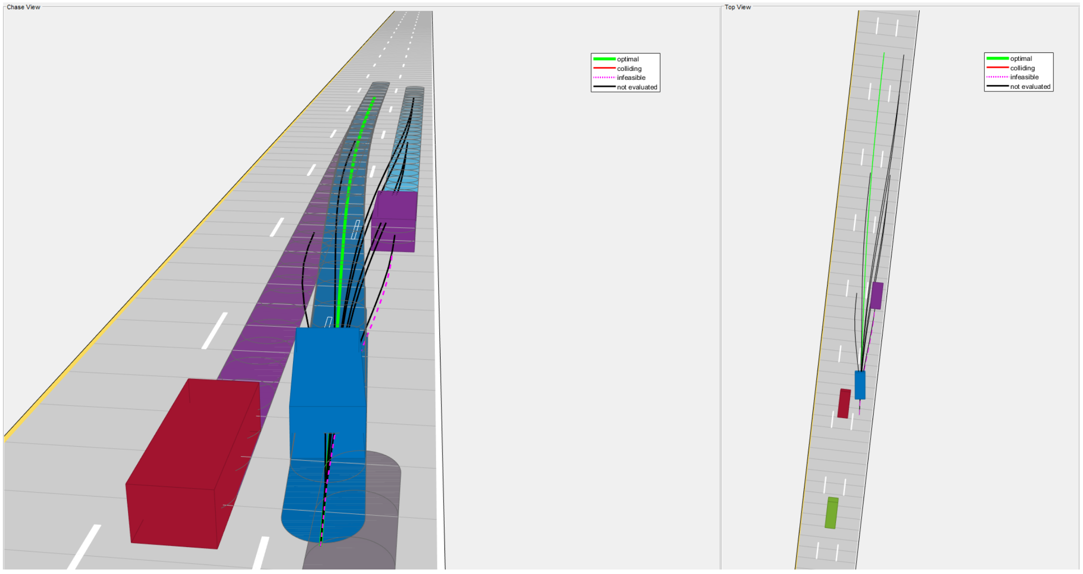

Figure 4.17: CCRb2C without, Ego vehicle at second 10.1

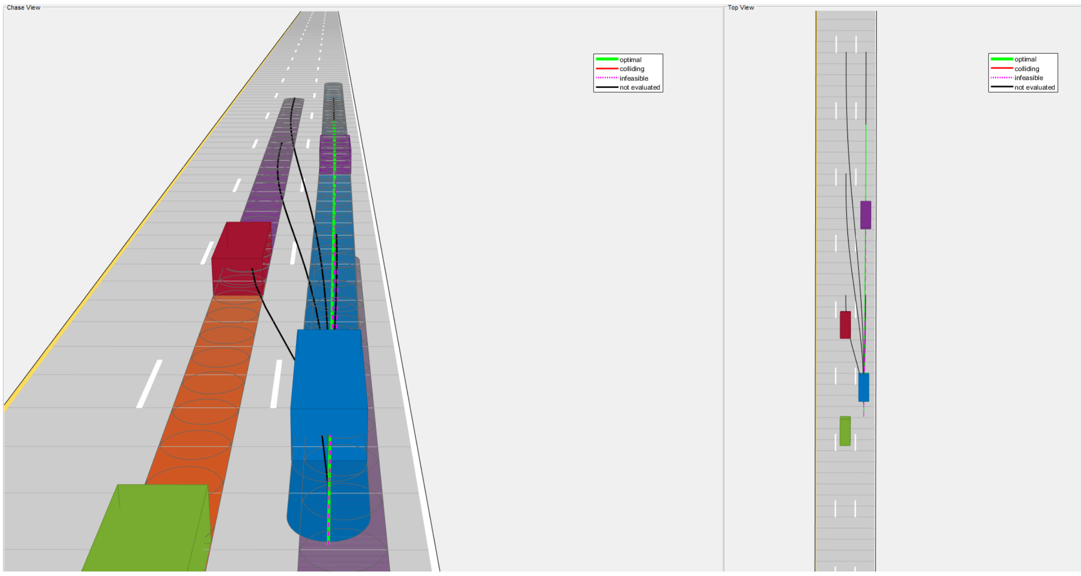

Figure 4.18: CCRb2C with AEB, Ego vehicle at second 10.1

8 in the lead, but when this target vehicle starts to brake, the Ego finds itself in a deadlock. With vehicle 7 in the adjacent right lane and vehicle 3 in the adjacent left lane, the Ego has to apply a full brake to immobilize the car before it crashes. In contrast, in the model without AEB, the Ego can't find any solution. With no alternative trajectory to escape that impasse, the Ego collides with the lead vehicle while braking. Both models will act precisely the same in test case CCRb3C as they performed in case CCRb3B, with the difference being that Ego goes to the adjacent left lane before starting braking. The execution of the second left lane change is now blocked not only by vehicle 3 but also by the vehicle in front of car 5 (both in the adjacent left lane). Everything else is precisely equal to the previous test case. In the test CCRb3D, Ego will choose to follow the lead vehicle 7, which had an increase in velocity until it started to brake. At that moment, Ego performs a left lane change, which will prevail until the end of the simulation. Due to the superior velocity of Ego and the fact that two of three

lanes will be blocked by braking cars, there is only one open lane; this means that after the Ego makes the left lane to avoid car 7, the highway is mainly free of vehicles. This makes Ego use the cruise control mode most of the simulation time.

The first case, CCRb4A, is very straightforward, with the Ego vehicle in the rightmost lane when car 8 changes to the right lane and ends up in front of the Ego. Abruptly, car 8 starts braking, and the only adjacent lanes are occupied by car 5. With all that, the Ego has no alternative but to brake to avoid collision. As a standard in these tests, the model without AEB can't brake, so it collides with the rear of the braking lead vehicle. In CCRb4B, the low velocity when vehicle 8 cuts in front of the Ego vehicles makes the Ego need to change to the adjacent lane. One problem is the significant velocity gap between car 8 and Ego, making the primary vehicle apply a small brake maneuver. This becomes very useful because car 5 simultaneously changes to the middle-right lane, where the Ego vehicle pretends to enter. With that simple brake, Ego already has a safe distance from a new lead vehicle (car 5) when entering the new lane. Moments after, vehicle 5 turned to the rightmost lane, and Ego had a clear lane until the end of the simulation. The test case CCRb4C is similar to its predecessor. The only difference is that when the Ego is in the middle-right lane following car 5, due to the velocity gap between the two vehicles, the Ego tries to change to the rightmost lane. The problem is that during this lane change process, the Lead vehicle also starts making a right lane change. This causes the Path Planner to abort the trajectory of lane change, return to the previous lane (middle-right), and let vehicle 5 take the rightmost lane. Finally, in test CCRb4D, the Ego longitudinal velocity is so high that when car 8 cuts in front of the Ego in the rightmost lane, the Ego has only a fraction of a second to react, braking and planning to change lanes. The problem is that the Ego needs to apply a higher deceleration value, losing momentum to make faster lane changes. At that precise moment, car 5 turns to the middle-right lane and blocks any trajectory from Ego to change to the only adjacent lane available. Trapped by all sides, when vehicle 8 starts braking, Ego must also start braking to maintain a safe distance.

Finally, the last scenario is relatively straightforward. In case CCRb7A, Ego activates the brakes to regulate the distance gap between Ego and car 2 when this one brakes close to the entryway. The Ego changes to the left lane, where it remains without any non-safe targets. Test cases CCRb7B and CCRb7C are very similar due to the low traffic density and lanes available, acting the same, With Ego following car 2 until this lead vehicle starts braking. Ego is forced to brake and change to the left lane, where they remain until the end of the simulation. The only things that change are the Ego longitudinal velocity and degree of deceleration that increase from one test case to the other. This also implies that when turning, in case CCRb7C, the Ego passes closer to the vehicle than in the other test case. This shows the influence of reaction time on the distance to brake. If we increase the velocity, the Ego starts applying the braking maneuver at a longer distance from the lead vehicle. The last case is CCRb7D, where Ego travels at maximum velocity. It will reach vehicle 2 much faster than in previous scenarios. This means that when car 2 starts to brake, vehicle 3 still occupies the adjacent left lane, blocking any trajectory or possibility for a lane change. With no solution, Ego brakes simultaneously with the lead vehicle until it becomes entirely stationary without colliding.

Identification	Collision	Time of Collision	Collision with AEB	Type of Brake
CCRm3C	√	20.4	X	PB
CCRs4B	√	9.6	X	PB

Table 4.10: Collisions analyses for special category

As usual, the model without AEB can't brake, so it collides with a lead car. It is essential to mention that Ego's braking maneuver tries to mimic the braking maneuver of the lead vehicle to keep a safe distance between cars at the current time. This allows Ego to gain some time to plan alternative actions and to wait for the traffic density to change in his favor. Unfortunately, there is only one adjacent lane, and vehicle 3 is particularly side by side with Ego, which makes it impossible for any action of a lane to change by Ego being forced to brake instead to avoid a collision.

4.7 Overall Results and Discussion

Two remaining test cases were built to test some specific scenarios already mentioned. Their objective is to try possible variations of outcomes. The target vehicles in the scenario simulate a reaction to Ego actions. All non-ego cars have a constant velocity and a fixed set of waypoints that define their trajectory. However, if the Ego reduces velocity (brakes) or does some dangerous maneuver in front of them, they don't have any reaction to it. This extra scenario has the function of making some vehicles simulate some reactions according to Ego vehicle behavior.

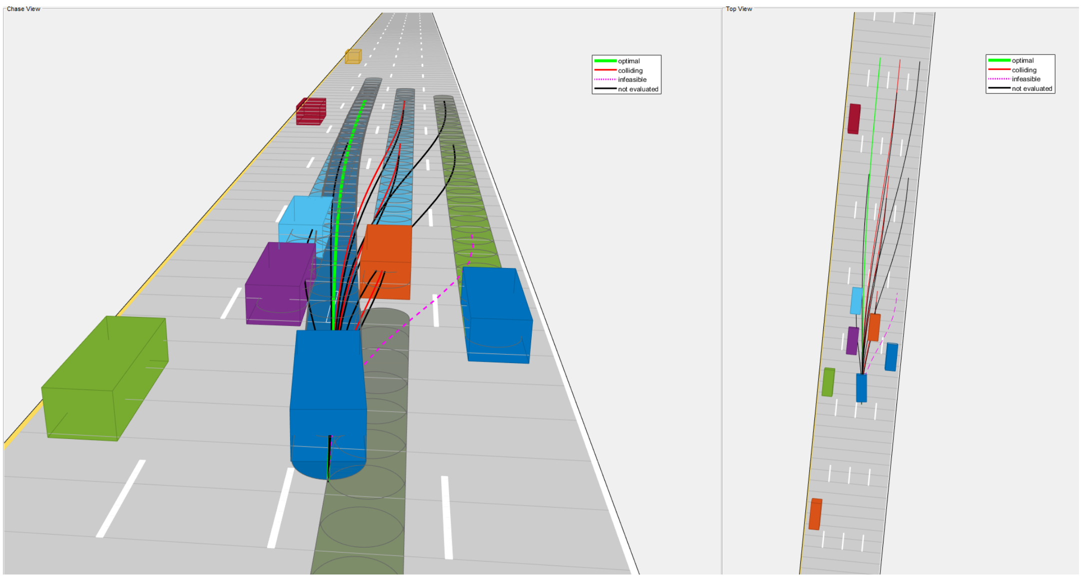

Figure 4.19: Special case CCRm3C, second left lane change

In the first case, the velocity of vehicle 7 increases by $2m/s$; this means that the previous deadlock that Ego will find itself when changing to middle-right after infringing the velocity gap with the lead vehicle 3. In the standard test case, Ego is located in the deadlock, lead vehicle 8 blocking the Ego lane, and car 7 occupying the adjacent right side-by-side with Ego. At the same time, the adjacent left lane is occupied by two cars. Vehicle 5 is in the lead and is followed by vehicle 3. This forces Ego to activate the AEB to avoid collision with the lead vehicle. However, in this test case, the velocity of vehicle 7 increases, making it much closer to lead car 5. This means that when the Ego vehicle starts to break slowly, this proximity between the two cars creates an avenue where the Ego car can pass to the left lane. This makes a particular case where the two non-ego vehicles are so close to each other that they almost collide, as shown in Figure 4.19, but this proximity allows Ego to have an avenue to change to the left lane. Still, with this case, we can emphasize the excellent gain for Path Planner to wait for the traffic density to change with time using a more gradual braking approach to the problem. After the vehicle gets into the middle-left lane, it faces a new issue (making decisions has a cascade of new challenges). With vehicle 4 cutting to the Ego lane right in front of it, Ego brakes to adjust his velocity with the lead vehicle for the third time, first in the deadlock and the second when following lead 3 soon after the first left lane change. While following vehicle 4, Ego makes a last lane change, ending in the leftmost lane, where it remains until the end of the simulation. We can see the overall performance of the Ego vehicle in Figure 4.20, where we can see all the times that the ego needs to brake to maintain the velocity gap defined by the AEB system. We can also see all three times when the Ego vehicle changes left lane.

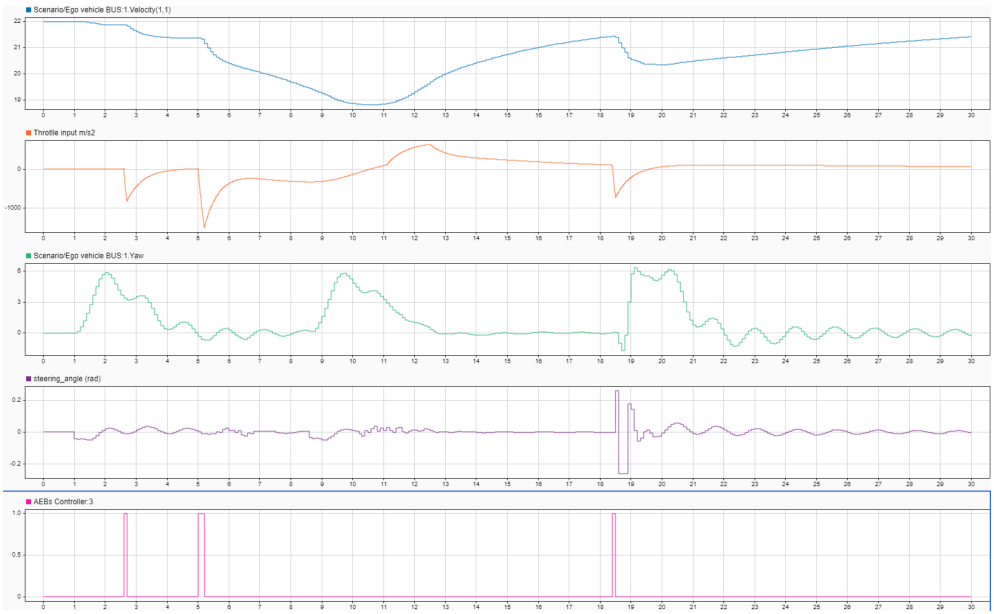

Figure 4.20: Special case CCRm3C Data Inspector

The final test case is quite simple, a variation of the test CCRs4B, but with a simple detail. In this scenario, Ego changes to the middle due to the velocity gap with vehicle 8, where it finds a stationary vehicle blocking the new Ego lane. In the standard test case, Ego tries to return to the rightmost lane but needs to brake to wait for the rear vehicle in that lane (car 3) to pass by Ego. However, in this scenario, vehicle 3 was designed to brake and allow the Ego Vehicle to pass in front. Simulating a species of recognition by that vehicle to the attempt of Ego return to the driving route. This means the Ego can generate a smoother and faster trajectory to return to its stride.

From an overall perspective, the model has distinct ways to react to dangerous situations. In

CCRm, the vehicle prioritizes small brake maneuvers to maintain a healthy velocity gap with the lead vehicle. At the same time, it can't find an alternative solution to the low velocity of the leading car. Generally, in this case, when the simple brake, followed by a reduction of Ego velocity, allows the non-safe target to progress along the highway, which opens space for an alternative trajectory that can be explored by the Ego vehicle. This is why we can only find partial brake maneuvers in test cases of this category. Ego can only resort to these strategies due to the non-existence of objects that can block a highway lane. However, this same rule is not applied in the other two categories. Things are more complex in CCRs with the introduction of stationary vehicles in front of the Ego. It only requires the adjacent lanes to the Ego lane to be occupied by a moving car traveling side by side or close by Ego, that all possible lane changes are impossible to be applied, and that the Ego vehicle is forced to use the brake. When the Ego detects the stationary vehicle as one of the non-safe targets, it can immediately start to predict all trajectories that allow it to avoid that dangerous obstacle on the road (if any exist). When Ego breaches the velocity or distance gap, it will start braking and reducing its velocity to match with the lead. If the lead vehicle's velocity is zero, it will perform a slow and gradual braking maneuver until it stops its longitudinal displacement. Then, it tries to find any adjacent lane clear of non-safe targets where it performs a lane change. When this step is executable, it is possible to separate all test cases from the ones that end up in a complete brake maneuver, immobilizing Ego, or those that only need to execute a simple partial brake and can continue their simulation. The last category is the most challenging. In this case, CCRb presents lead vehicles moving at a constant velocity and suddenly starting to brake with considerably high deceleration rates. The problem is that the capsule block system is suitable for determining future dangers of zones of non-safe targets that are moving straightforwardly and maintain the current velocity, which is not applied to the test cases of this category since some cars tend to transgress the last norm. With future behavior prediction focusing more on continuous velocities rather than a variation of accelerations, the projections designed by Path Planning change significantly during the entire process when handling a lead vehicle in a braking process. The AEB reacts to this by adjusting its acceleration at every time cycle to the breach of velocity and distance gap since the AEB system is independent of the Path Planner. However, this negates, in many cases, the ability of the Path Planner to generate suitable trajectories to avoid the complete braking process. In theory, this category should be midground between the two previous categories, with the vehicle braking process giving time to Ego to find an alternative trajectory and the vehicle becoming entirely stationary when the target braking maneuver is completed. However, this is not true because this shift in acceleration undermines all the future projections of the Path Planner, making it more difficult for Ego to create a suitable trajectory for lane changing in the long term. All these conditions combine to make this category an utterly new challenge with new rules and dynamics. As stated, the AEB system is not directly dependent on the Path Planner's defined trajectories. However, it depends on blocks like the Controller, which receives some information from the Planner. In the case of the Path Panner, it can't define any trajectory to avoid collision with the lead vehicle. The AEB system will always ensure that the Ego reduces its velocity whenever it breaches any velocity or distance gap defined for any of the three stages of the AEB. That is why, in all cases, the vehicle could brake before colliding with the lead vehicle. But imagine any possible scenarios in which the road conditions are such that there is no distance or velocity gap enough to ensure sufficient brake time before the collision. In these hypothetical circumstances, the Ego vehicle still reduces its velocity to the maximum possible when colliding. The jerk force will be at the minimum possible for those circumstances, and the physical impact on the passenger is reduced to the lowest values possible for those cases.

Chapter 5

Conclusions and Future Works

The new era of technology is propagating over all branches essential for humans, with the locomotion of people not being different. Technological innovations are reaching sectors like aeronautics, drones, trains, and the most common means of transport, vehicles. With parking assistance and road congestion a center of innovation, the ultimate goal will be to reach a state of fully autonomous vehicles capable of handling all scenarios, which, on its own, represent a herculean effort of engineering and innovation. Actual tests on means of transportation represent a high economic risk, with errors and bugs having an enormous financial cost for each company in the development process. Simulations and methodic virtual tests are the way to reduce this cost, plus cutting the development time of setting an actual vehicle to every testing scenario imaginable. This dissertation implements an AEB system with a fully autonomous path planning and controller system in software-in-the-loop simulation using verification, validation, and testing methods to automatically test the prototype AV performance, precision, and accuracy on all specifically designed scenarios. This paper aims, firstly, to show a survey relative to the principal topic published in recent years, covering the most diverse topic of technology, especially software developed for vehicles. Then, it was successfully described how this model could avoid collision to ensure vehicle and passenger safety while driving autonomously in diverse highway environments. The Path Planner system consists of three main ADAS functions (CC, FLV, and LC), which are capable of alternating between them according to the best course of action. The decision-making can choose the vehicle's optimal driving mode over time based on the relative distance and velocity gap between the Ego and the target vehicles in the current lane and any of the adjacent lanes. With the predictions of future behavior, it is possible to control the car dynamically over time, adjusting to the most plausible development of the traffic density. Another core component is the adaptive MPC controller, which commands the acceleration and steering of the vehicle to achieve the best possible vehicle dynamic behavior to perform the path set by the Planner. Furthermore, the Controller is optimized by the AEB system, which aims to prevent any collision between the Ego and target vehicles, making it take a more cautious approach to the navigation execution of the overall system, prioritizing vehicle physical integrity and passenger safety over optimal performance. All these components are aggregated in a Matlab and Simulink test bench, which also contains tools and models specifically designed for this type of system and projects. All test cases are split over three categories, seven scenarios, and four modes of set Ego velocity. They are ensuring the ability to reach a high degree of variety that is essential for validating and verifying this type of Software-in-Loop test. Moreover, both test benches, with and without the integration of the AEB system, were performed to evaluate the overall contribution of the AEB, showcasing its many attributes and downsides. Considering the limited funds and simulation tools in this research, the results obtained satisfy the expectations of the dissertation topic. However, further work can increase the system's overall performance, significantly solving some of the preconditions defined for this project, like the implementation of Lane Center Keeping Assist to add some scenarios with highways presenting some degree of curvature, with curves and counter curves along the road length. It is also possible to incorporate other ADAS systems or integrate the AI algorithm into the Planner system to enhance the performance and reduce the system's computational cost. Another possibility is adding another scenario where the highway starts with three available lanes and is then reduced to only two or one traveling lane. A more advanced improvement can be the capacity to handle crossroads, interpret traffic lights or signs, and implement some V2X communication and data transfer protocols to enhance the information received about the surrounding environment. The final step moves to the next testing phase with the hardware-in-the-loop simulation, adding actual sensors and some independent pieces of the hardware to simulate vehicle dynamics and essential components, like brakes, lights, and others.

Bibliography

[1] R. V. Gilabert, E. T. Dill, K. J. Hayhurst, and S. D. Young, "Safeguard: Progress and test results for a reliable independent on-board safety net for uas," in *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, 2017, pp. 1–9. 1

[2] N. Rajabli, F. Flammini, R. Nardone, and V. Vittorini, "Software verification and validation of safe autonomous cars: A systematic literature review," *IEEE Access*, vol. 9, pp. 4797–4819, 2021. 2

[3] H. Kopetz and S. Poledna, "Autonomous emergency braking: A system-of-systems perspective," in *2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)*, 2013, pp. 1–7. 2

[4] L. Zhuo, "Research on reliability of autonomous emergency braking system based on prescan," in *2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, 2022, pp. 681–684. 6

[5] S. Rajendar and V. K. Kaliappan, "Recent advancements in autonomous emergency braking: A survey," in *2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2021, pp. 1027–1032. 6, 7

[6] C. Su, W. Deng, H. Sun, J. Wu, B. Sun, and S. Yang, "Forward collision avoidance systems considering driver's driving behavior recognized by gaussian mixture model," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 535–540. 7

[7] X. Sun and Y. Zhang, "Improvement of autonomous vehicles trust through synesthetic-based multimodal interaction," *IEEE Access*, vol. 9, pp. 28 213–28 223, 2021. 7

[8] R. Okuda, Y. Kajiwara, and K. Terashima, "A survey of technical trend of adas and autonomous driving," in *Proceedings of Technical Program - 2014 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*, 2014, pp. 1–4. 8

[9] Y. Zhao and H. Lin, "A study on brake control strategy for the uav all-electric brake system," in *2019 22nd International Conference on Electrical Machines and Systems (ICEMS)*, 2019, pp. 1–5. 8

[10] G. Xiog, T. Yang, M. Li, Y. Zhang, W. Song, and J. Gong, "A novel v2x-based pedestrian collision avoidance system and the effects analysis of communication delay and packet loss on its application," in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2018, pp. 1–6. 9

[11] J. Erickson, S. Chen, M. Savich, S. Hu, and Z. M. Mao, "Commpact: Evaluating the feasibility of autonomous vehicle contracts," in *2018 IEEE Vehicular Networking Conference (VNC)*, 2018, pp. 1–8. 9

[12] D. K. Murthy and A. Masrur, “Braking in close following platoons: The law of the weakest,” in *2016 Euromicro Conference on Digital System Design (DSD)*, 2016, pp. 613–620. 10

[13] S. Yang, M. Fu, Y. Yang, D. Tang, W. Song, and M. Zhang, “Autonomous vehicle following system in off-road environment,” in *2020 3rd International Conference on Unmanned Systems (ICUS)*, 2020, pp. 1173–1179. 10

[14] Q. H. Tho, H. C. Phap, and P. A. Phuong, “A predictive control solution for contingency motion planning for autonomous vehicle,” in *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*, 2019, pp. 1–6. 11

[15] F. Romeo, C. Campolo, A. Molinaro, and A. O. Berthet, “Denm repetitions to enhance reliability of the autonomous mode in nr v2x sidelink,” in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5. 11

[16] S. Mariani, G. Cabri, and F. Zambonelli, “Coordination of autonomous vehicles: Taxonomy and survey,” *ACM Comput. Surv.*, vol. 54, no. 1, feb 2021. [Online]. Available: https://doi.org/10.1145/3431231 12

[17] U. Z. A. Hamid, F. R. A. Zakuan, K. A. Zulkepli, M. Z. Azmi, H. Zamzuri, M. A. A. Rahman, and M. A. Zakaria, “Autonomous emergency braking system with potential field risk assessment for frontal collision mitigation,” in *2017 IEEE Conference on Systems, Process and Control (ICSPC)*, 2017, pp. 71–76. 12

[18] G. Savino, J. Brown, M. Rizzi, M. Pierini, and M. Fitzharris, “Triggering algorithm based on inevitable collision states for autonomous emergency braking (aeb) in motorcycle-to-car crashes,” in *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 1195–1200. 13

[19] G. Savino, F. Giovannini, M. Fitzharris, and M. Pierini, “Inevitable collision states for motorcycle-to-car collision scenarios,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2563–2573, 2016. 13

[20] J. Lee, G. Kim, and B. Kim, “Study on aeb performance improvement on curved road based on curvilinear coordinate system,” in *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, 2019, pp. 65–68. 14

[21] S.-G. Shin, D.-R. Ahn, Y.-S. Baek, and H.-K. Lee, “Adaptive aeb control strategy for collision avoidance including rear vehicles,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 2872–2878. 14

[22] M. Ariyanto, G. D. Haryadi, M. Munadi, R. Ismail, and Z. Hendra, “Development of low-cost autonomous emergency braking system (aebs) for an electric car,” in *2018 5th International Conference on Electric Vehicular Technology (ICEVT)*, 2018, pp. 167–171. 15

[23] V. A. Banks, N. A. Stanton, and C. Harvey, "What the crash dummies don't tell you: The interaction between driver and automation in emergency situations," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, pp. 2280–2285. 15

[24] C. Zhou, B. Li, S. Li, and K. Huang, "Rlda-based braking intention detection algorithm for l3 autonomous driving," in *2021 IEEE International Conference on Unmanned Systems (ICUS)*, 2021, pp. 554–559. 16

[25] J. Fu, X. Zhang, W. Yu, J. Li, M. M. Atia, H. Wang, C. Li, and Z. Hao, "Decoding passenger's eeg signals from encountering emergency road events," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 2214–2219. 16

[26] T. Weiskircher, Q. Wang, and B. Ayalew, "Predictive guidance and control framework for (semi-)autonomous vehicles in public traffic," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 6, pp. 2034–2046, 2017. 17

[27] L. Xiong, M. Liu, X. Yang, and B. Leng, "Integrated path tracking for autonomous vehicle collision avoidance based on model predictive control with multi-constraints," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 554–561. 17

[28] L. Guo, P. Ge, and D. Sun, "Variable time headway autonomous emergency braking control algorithm based on model predictive control," in *2020 Chinese Automation Congress (CAC)*, 2020, pp. 1794–1798. 18

[29] Y.-T. Lin, C.-C. Yao, and B.-H. Lin, "Diagnostic system and system failure protection for autonomous vehicle based on driving path and visual information," in *2021 International Automatic Control Conference (CACS)*, 2021, pp. 1–5. 18

[30] R. Hajiloo, M. Abroshan, A. Khajepour, A. Kasaiezadeh, and S.-K. Chen, "Integrated steering and differential braking for emergency collision avoidance in autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 3167–3178, 2021. 19

[31] F. P. Ebrahimpour and H. Ferdowsi, "Multi-constraint predictive control system with auxiliary emergency controllers for autonomous vehicles," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, 2021, pp. 274–279. 20

[32] M. Cao, R. Wang, J. Wang, and N. Chen, "An integrated mpc approach for fwia autonomous ground vehicles with emergency collision avoidance," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2432–2437. 20

[33] L. Wang, Z. Zhan, X. Yang, Q. Wang, Y. Zhang, L. Zheng, and G. Guo, "Development of bp neural network pid controller and its application on autonomous emergency braking system," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1711–1716. 20

[34] D. Katare and M. El-Sharkawy, "Embedded system enabled vehicle collision detection: An ann classifier," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 0284–0289. 21

[35] S. Lim, S. Lee, and S.-C. Kim, "Clustering of detected targets using dbscan in automotive radar systems," in *2018 19th International Radar Symposium (IRS)*, 2018, pp. 1–7. 22

[36] D. Belgiovane and C.-C. Chen, "Micro-doppler characteristics of pedestrians and bicycles for automotive radar sensors at 77 ghz," in *2017 11th European Conference on Antennas and Propagation (EUCAP)*, 2017, pp. 2912–2916. 22

[37] J. Kim, W. Y. Jung, S. Kwon, and Y. Kim, "Performance test of autonomous emergency braking system based on commercial radar," in *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, 2016, pp. 1211–1212. 23

[38] M.-g. Cho, "A study on the obstacle recognition for autonomous driving rc car using lidar and thermal infrared camera," in *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*, 2019, pp. 544–546. 23

[39] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon, "Kaist multispectral day/night data set for autonomous and assisted driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 934–948, 2018. 23

[40] T. Liu, Selpi, and R. Fu, "The relationship between different safety indicators in car-following situations," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1515–1520. 24

[41] C. Li, C. He, Y. Yuan, and J. Zhang, "Braking evaluation of integrated electronic hydraulic brake system equipped in electric vehicle," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2019, pp. 2361–2365. 25

[42] C.-H. Yu, Y.-Z. Chen, and I.-C. Kuo, "The benefit of simulation test application on the development of autonomous driving system," in *2020 International Automatic Control Conference (CACS)*, 2020, pp. 1–5. 25

[43] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa, and C. S. Pan, "Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 595–607, 2017. 26

[44] P. Riegl, A. Gaull, and M. Beitelschmidt, "Parameterization of automated driving functions in virtual environments based on characteristic test scenarios," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 511–518. 27

[45] H. Felbinger, F. Klück, Y. Li, M. Nica, J. Tao, F. Wotawa, and M. Zimmermann, "Comparing two systematic approaches for testing automated driving functions," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, 2019, pp. 1–6. 27

[46] M. Matsubara and M. Aoyama, "An analysis method of safety requirements for automotive software systems," in *2017 24th Asia-Pacific Software Engineering Conference (APSEC)*, 2017, pp. 408–416. 27

[47] M. Barbier, A. Renzaglia, J. Quilbeuf, L. Rummelhard, A. Paigwar, C. Laugier, A. Legay, J. Ibañez-Guzmán, and O. Simonin, "Validation of perception and decision-making systems for autonomous driving via statistical model checking," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 252–259. 27

[48] A. Dixit, P. D. Devangbhai, and C. R. Kumar, "Modelling and testing of emergency braking in autonomous vehicles," in *2021 Innovations in Power and Advanced Computing Technologies (i-PACT)*, 2021, pp. 1–6. 28

[49] M. Li, J. Yang, and H. Zhu, "Study on occupant out-of-position displacement and injuries under aeb braking," in *2022 14th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 2022, pp. 837–842. 28

[50] Y. Gao, Z. Xu, X. Zhao, G. Wang, and Q. Yuan, "Hardware-in-the-loop simulation platform for autonomous vehicle aeb prototyping and validation," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6. 29

[51] Y. Zhu, R. Xu, H. An, A. Zhang, and K. Lu, "Research on automatic emergency braking system development and test platform," in *2022 Fifth International Conference on Connected and Autonomous Driving (MetroCAD)*, 2022, pp. 1–6. 29

[52] S. Cheng, L. Li, H.-Q. Guo, Z.-G. Chen, and P. Song, "Longitudinal collision avoidance and lateral stability adaptive control system based on mpc of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2376–2385, 2020. 30

[53] Y. Fu, C. Li, F. R. Yu, T. H. Luan, and Y. Zhang, "A decision-making strategy for vehicle autonomous braking in emergency via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5876–5888, 2020. 31

[54] F. Klück, F. Wotawa, G. Neubauer, J. Tao, and M. Nica, "Analysing experimental results obtained when applying search-based testing to verify automated driving functions," in *2021 8th International Conference on Dependable Systems and Their Applications (DSA)*, 2021, pp. 213–219. 31

[55] L. Yu, R. Wang, and Z. Lu, "Autonomous emergency braking control based on inevitable collision state for multiple collision scenarios at intersection," in *2021 American Control Conference (ACC)*, 2021, pp. 148–153. 32

[56] I.-C. Han, B.-C. Luan, and F.-C. Hsieh, "Development of autonomous emergency braking control system based on road friction," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, 2014, pp. 933–937. 32

[57] D. Wang, K. Nazem Tahmasebi, and D. Chen, "Integrated control of steering and braking for effective collision avoidance with autonomous emergency braking in automated driving," in *2022 30th Mediterranean Conference on Control and Automation (MED)*, 2022, pp. 945–950. 33

[58] K. Yoneda, T. Iida, T. H. Kim, R. Yanase, M. Aldibaja, and N. Suganuma, "Trajectory optimization and state selection for urban automated driving," vol. 23, no. 4, 2018, p. 474–480. 33

[59] Y. Liu, J. Lu, X. Long, R. Zhou, L. Liu, and Y. Wu, "Research on emergency braking strategy of driverless vehicle based on overload optimization control," in *2019 22nd International Conference on Electrical Machines and Systems (ICEMS)*, 2019, pp. 1–7. 33

[60] W. Zhou and X. Wang, "Calibrating and comparing autonomous braking systems in motorized-to-non-motorized-vehicle conflict scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 20 636–20 651, 2022. 34

[61] T. D. Gillespie, *Fundamentals of vehicle dynamics (Vol. 114)*. SAE Technical Paper, 1992. 35

[62] T. Takahama and D. Akasaka, "Model predictive control approach to design practical adaptive cruise control for traffic jam," *International Journal of Automotive Engineering*, vol. 9, no. 3, pp. 99–104, 2018. 52