POLITECNICO DI TORINO

Master Degree in Mechatronic Engineering

Master Degree Thesis

Modeling and Analysis of an ADAS Overtaking System in Complex Driving Scenarios

Supervisors

Prof. Stefano Malan

Ing. Angelo Borneo

Candidate

Claudio Squicciarini

December 2024

Abstract

This master thesis presents research targeting a driver assistance system where it is possible to understand the environment and prepare a safe path for overtaking when necessary. Overtaking is known as one of the most dangerous driving phases; therefore, creating systems aimed at reducing errors made by drivers and unambiguity in their actions is essential. This work is positioned within the rapidly evolving domain of vehicle automation that from the end of the year 2023 has allowed a SAE level higher than level 2.

Furthermore, advanced simulation models are used to replicate realistic and challenging driving scenarios, testing the response of the system in critical situations such as overtaking with oncoming vehicles. These provide valuable insight into the performance of the system under critical circumstances and serve to further develop and test the capabilities of the system on different road conditions.

For this type of system, a set of basic principles is thoroughly analyzed, including modeling each element from the theoretical standpoint and the possible use of adequate software. To accomplish this aim, a method for the construction and validation of the performing system, irrespective of the specific scenario of its operation, is developed. Several test cases are constructed with the aim of identifying possible worst-case scenarios of the system. This makes it possible to analyze both satisfactory and unsatisfactory outcomes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The automotive sector aims for a higher automation of vehicles to optimize driving and reduce road accidents. According to the latest Figures from ISTAT, the percentage attributable to driver error concerning all accidents is about 90.1% [1]. This explains the intense research and development of advanced driver assistance technologies, better known as ADAS, whose purpose is to improve the safety of roads by employing automation.

In particular, research in this field focuses on the impact of these technologies on vehicle safety, with the aim of increasing efficiency and extending those functions manageable without direct intervention by the driver. Examining typical driving scenarios, one of the most dangerous and crucial maneuvers is definitely overtaking. The high speed at which this maneuver is normally performed, combined with possible risks such as invading oncoming lanes or overtaking on a bend, raises the danger substantially.

These maneuvers force the drivers to deal with a high volume of information in real time, which can surely compromise their ability to perceive road signs. Advanced ADAS systems are developed in this context, not only to avoid accidents in risky maneuvers but also to provide much more aware and safe driving.

Automation improves safety directly, while at the same time, automation stands for the cultural shift in driving. The changes that will be brought about by the ongoing progress of integrating these technologies lead to fewer accidents and safer driving for all.

1.1 Self-driving technologies

"Automatic driving technologies" are defined as innovative technologies for automatic driving based on various types of sensors, software for processing sensor data and interpreting traffic situations, learning software, software for making driving decisions and for their implementation, components for integration with the traditional vehicle, which fall within the scope of road testing. ADAS are part of automatic driving technologies, therefore, they are based on inputs from the vehicle sensors. These signals are then processed by the various assistance systems, which in some cases cooperate, in order to be able to evaluate the safety of the vehicle and its occupants, the achievement of a specific driving objective or even start decision-making processes.

1.1.1 Main ADAS systems

About what has just been described, it is useful for the design of an automatic overtaking system to examine the main technologies with which a vehicle can be equipped, to identify the sensors necessary to obtain the required inputs, and determine the main ADAS with which to communicate. The main advanced systems that can be supplied to the fleet of current cars include:

- ABS: Anti-lock Braking System. Works on braking efficiency by preventing wheel lock and prevents the vehicle from losing directionality.
- TCS: Traction Control System. Prevents wheel slippage during the acceleration phase.
- ESP: Electronic Stability Program. It acts in the event of a skid by regulating the power coming from the engine and the braking torque on the individual wheels.

It should be noted that this is only a small part of all the ADAS installed in modern cars. The ones mentioned above represent the most important safety systems that have been required by law for some time.

The ADAS reported below are also fundamental for the development logic of assistance systems such as the one studied in this thesis.

- ACC: Adaptive Cruise Control. Allows the vehicle to set a given speed, maintain it, and possibly adapt it to traffic conditions detected
- LKA: Lane Keeping Assist. Allows the driver to keep the vehicle within the lane over a wide speed range.

ACC and LKA are particularly useful for an automatic driving system, as their cooperation allows for control of both longitudinal and lateral dynamics at the same time.

Among the most widespread systems that implement commands directly on the vehicle, there is Park Assist. It is dedicated to autonomously carrying out entry and exit maneuvers related to a parking space. The cooperation with the vehicle sensors is particularly interesting, in order to precisely identify the surrounding scenario.

For an assistive system such as the one under study, where it must establish the maneuver to be performed in relation to the scenario analysis, cooperation with the Traffic-Sign Recognition (TSR) system is particularly useful. Finally, one of the mandatory systems that has a great impact on vehicle safety is the Automatic Emergency Braking (AEB). It uses sensors and cameras to detect the presence of obstacles in the area in front of the vehicle; emitting an acoustic and visual signal if an obstacle is detected and, if the driver does not intervene, it implements a braking action to avoid the collision or reduce the speed at the moment of a possible impact.

Lane Change Assist (LCA)

Among the many ADAS available on new cars, the LCA is becoming increasingly LCA. The Lane Change Assist allows the car to change lanes autonomously to overtake. Generally, the driver only has to operate the direction indicator or a button.

This is a technology developed to allow a safe lane change; it uses a set of sensors to detect and signal vehicles in adjacent lanes. In addition to assisting the driver during a maneuver

that requires maximum attention, it also allows to broaden the perception of what happens in the surrounding environment, since thanks to this system it is possible to identify vehicles that are in the so-called blind spots of the ego vehicle as shown in Figure 1.1.

Nowadays, two different types of LCA have been developed; which can be identified as passive LCA and active LCA. As the name suggests, the difference lies in the ability of the system to actively act on the steering and speed of the vehicle. The passive LCA is a system designed to provide additional information on the surrounding environment to the driver intending to overtake; therefore, it is a warning system that is now part of the standard equipment, as it has been developed and introduced by all car manufacturers.

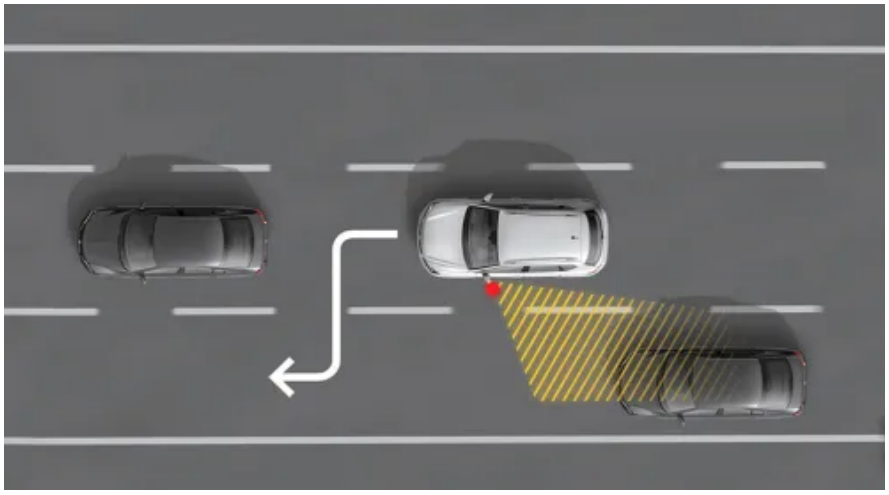

Figure 1.1: Blind spot detection.

The passive LCA simply signals the criticality of the maneuver through a warning light placed in the window and the warning is provided if vehicles are detected proceeding at speeds incompatible with a lane change.

The active LCA allows the vehicle, through the assistance of sensors and actuators, to change lane in a totally autonomous manner when this maneuver is requested by the driver and after having carried out all the necessary safety checks. Therefore, the active LCA can be considered as the anticipation of a totally autonomous overtaking system, such as the one integrated into a vehicle with an automation level higher than the second level.

1.1.2 Sensors of autonomous vehicles

Advanced Driver Assistance Systems (ADAS) are based, in addition to the signals that describe the characteristics of the vehicle and its operation, on the reconstruction of the scenario in which the vehicle is located. In order to receive precise inputs about what surrounds the vehicle, and then analyze them to allow the dedicated systems to evaluate the safety of the vehicle itself and its occupants, it is necessary to equip the car with a set of sensors capable of accurately reconstructing the scenario. To understand the operation of ADAS, some examples of which have been reported previously, the following is an examination of the sensors of modern vehicles.

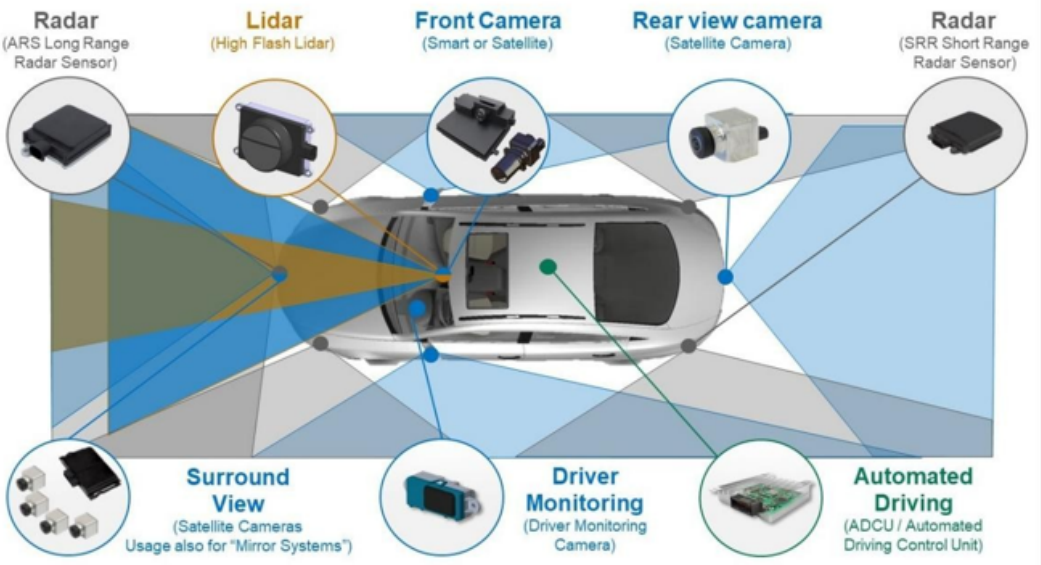

Figure 1.2: Block diagram of ADAS.

Figure 1.2 shows a representation of the main sensor components of a vehicle capable of having a 360° perception of the surrounding scenario.

The equipment includes:

- A set of cameras that completely covers the surrounding field of view of the car.
- A radar system that covers the so-called "blind spots", including areas near the rear pillars and the front of the vehicle.
- A LiDAR sensor with a forward-facing field of action.

- A set of cameras that provide the so-called "surround view", that assists the driver during parking maneuvers by combining all the information provided by 4 cameras placed around the vehicle that offers a 360-degree view of the area around the car [2].
- A camera that monitors the pilot's behavior.

In addition to those reported in Figure 1.2, a geo-localization system (GPS) is usually used and in some cases also a set of ultrasonic sensors. For a system like the one in this study, the useful data that can be obtained from the vehicle sensors are the lane location, the location of other vehicles present in the scenario, and the geometric definition of the scenario itself.

Radar

The main type of device from which to obtain the signals necessary for an automatic overtaking system is the radar. This is a device capable of detecting the distance, speed, and azimuth angle of any object that reflects part of the energy emitted by the transmitter to the receiver. Of military origin, it has been widely used in recent years in the cars produced, as it is necessary for well-known ADAS systems such as ACC and LCA; therefore, the advantage in its use is undoubted, as it is already supplied to modern cars.

LiDAR

LiDAR, which stands for "Light Detection And Ranging", is a device that allows to measure the distances between the sensor and objects in the scene with high precision, using laser technology. Unlike the microwaves used by the radar sensor, the medium with which the LiDAR performs the measurement, i.e. reflected light, allows for greater speed of data acquisition and better precision of the same.

However, the range is limited to a few meters.

Vehicle to Everything

In terms of data acquisition methodologies related to the scenario, the "Vehicle to Everything" (V2X) has a significant impact in the context of the progressive development of vehicle automation technologies. This is a communication system that uses the WLAN network for a real-time exchange of information between all the components of the road scenario. It is made up of different technologies, each of which uses the same system to exchange information with a specific category of actors or objects, which influence and/or help the data analysis conducted by the assistance systems. The individual communication channels are the following:

V2D: Vehicle to Device refers to the communication between the vehicle and devices such as the smartphone. It enables personalized services and improved user experiences, such as navigation updates or alerts based on the driver's device.

- V2G: Vehicle to Grid focuses on the communication between the vehicle and the power grid, enabling the exchange of energy. For example, electric vehicles (EVs) can communicate with the power grid to optimize charging times, feed energy into the grid, and balance demand, thus supporting smart grid capabilities.
- V2I: Vehicle to Infrastructure involves the interaction between vehicles and road infrastructure (e.g., traffic lights, road signs, toll booths) facilitating improved traffic flow, optimized signal timing, and better coordination between vehicles and infrastructure to improve safety and efficiency.
- V2N: Vehicle to Network involves the exchange of information between vehicles and larger network systems, such as cloud services or telecommunications networks. V2N ensures that vehicles are connected to external services for navigation, traffic updates and emergency assistance.
- V2P: Vehicle to Pedestrian is designed to improve the interaction between vehicles and pedestrians. It allows vehicles to detect and communicate with pedestrians to reduce accidents, providing alerts to both drivers and pedestrians in real time.
- V2V: Vehicle to Vehicle enables direct communication between vehicles, sharing

information about speed, location and road conditions. This improves safety by reducing collisions, providing cooperative driving strategies and supporting autonomous driving and platooning systems.

Developing systems of this type, bringing each vehicle to an omniscience relative to the context in which it is located, would lead to enormously impactful benefits in terms of safety and potential for the development of assistance.

1.2 SAE Automation Levels

The presence of systems and devices useful for automating vehicles has led to the creation of a fleet of cars with different potential in terms of maneuvers that can be performed without the assistance of the driver. This variety has led to the need for a taxonomy with detailed definitions for each level of automation; which was introduced by the J3016 standard by SAE International [3]. The levels it defines are the following:

- Level 0: No Driving Automation.
- Level 1: Driver Assistance.
- Level 2: Partial Driving Automation.
- Level 3: Conditional Driving Automation.
- Level 4: High Driving Automation.
- Level 5: Full Driving Automation.

As can be seen from the brief descriptions, starting from Level 0 with totally zero automation and proceeding towards increasing levels, the importance of driver assistance systems and devices within the vehicle has an increasing relevance. The maximum achievable level is Level 5 which identifies a fully autonomous vehicle. After having defined the individual levels, the SAE has made a further distinction into two groups. Under the heading "Driver Support" the first three levels (from 0 to 2) and the last three (from 3 to 5) are identified as "Automated Driving". This distinction was made to immediately understand who between

the Driver and the Automated Driving System has actual control of the vehicle. Of particular relevance for this thesis are the vehicles belonging to the Automated Driving group, as their automated driving implies the ability to overtake when necessary. Level 3 vehicles are designed to be able to overtake the vehicle in front of them during a motorway journey; by increasing the level, the scenario in which this maneuver can be carried out is wider.

1.3 Regulatory context and ethical aspect

The impact of ADAS systems on driving safety is so evident, making them indispensable for modern cars. For several years, some of the systems mentioned have become mandatory for cars to be introduced on the market, thus raising the required standards regarding the level of automation of new vehicles.

1.3.1 EU Regulation - Introduction of new mandatory ADAS

Starting from 7 July 2024, EU regulation 2019/2144 makes the presence of the following assistance systems mandatory for all vehicles to be registered, whether newly or previously approved [4]:

- Intelligent Speed Adaptation (ISA): which integrates the ACC and the Traffic-Sign Recognition (TSR) systems into a single system. In this way, the vehicle can adapt its cruising speed both to the presence of other vehicles in the scenario and to the speed limits imposed by road signs.
- Interface for the breathalyzer: it is made mandatory to prepare vehicles to install an alcohol interlock device.
- Attention and fatigue monitoring: which detects fatigue through the position of the hands on the steering wheel and/or the movement of the steering wheel of the driver.
- Emergency stop indicator: the aforementioned AEB.
- Object and pedestrian detection when reversing: which uses special sensors to detect

the presence of obstacles when reversing.

- Event data recording: which records all data relating to extraordinary events, such as accidents, inside the so-called "black box". This device allows access to the data only to law enforcement.
- Lane Keeping Assist: the aforementioned LKA.
- Tire pressure monitoring system (TPMS): which informs the driver of the tire pressure, obtained from the data of the ABS control unit.

1.3.2 Ethical aspect

A wide variety of ethical issues arise from the introduction of self-driving technologies. More importantly, since investments in developing autonomous systems are rather large and concern the public sector and broader political decisions, this automatically implies the rise of public debates and also political debates.

A crucial issue is the attribution of responsibility in case of accidents and traffic safety. To this day, these are attributed to the driver, but as the level of automation improves, as well as communication systems, the road and the vehicle itself take a larger relevance. Conclusively it is up to the designer to calibrate the principles that guide the decision-making processes of the autonomous driving system, weighing the choices based on a series of factors.

Public opinion carries considerable weight regarding this matter, since, in ethical fields, one never finds an objective answer to the question posed. This makes the expected tolerance for accidents with self-driving vehicles much lower compared with accidents involving traditional vehicles. The above considerations can lead to delays in their introduction on the market or to requests for limitations on their functionalities.

1.4 Thesis structure

The aim of this thesis is to conduct an examination of the principles and components that form the basis of an advanced driver assistance system dedicated to automatic overtaking.

Specifically, we prefer to offer a theoretical and complete vision of the issues that need to be understood to design an ADAS of this kind.

The same topics have been analyzed from a different perspective, the practical one, in a dedicated simulation environment. For this purpose, useful tools have been collected to obtain and analyze the data needed by the aforementioned ADAS system, including a set of functions dedicated to vehicle simulation and a basic model, from which the complete model used for the reported simulations has been built.

As the title of the research suggests " Modeling and Analysis of an ADAS Overtaking System in Complex Driving Scenarios", simulation plays the role of the main actor. The tools presented during the research were used to create simulations capable of providing results useful for highlighting the critical issues of a system such as the one under study. These were then explored in depth both on a theoretical and practical level; that is, by proposing a correct methodology for manipulating signals within the simulation environment and then verifying its efficiency through the analysis of the results. In the final part of the work, a set of complex scenarios was introduced, with the aim of testing the system responses to different inputs; as they will certainly be different in a real application context.

In conclusion, it can be stated that the structure of the thesis has been developed with the intent of providing, by proceeding through the various chapters, an increasingly detailed vision of the topic under examination; with the ultimate goal of providing a perspective that allows the development of a system that has a solid logic, therefore independent from the adopted scenario.

Chapter 2

Planning the trajectory

The most important part of a driver assistance system designed for automatic overtaking is undoubtedly the motion planner or path planner. This is the area of the system dedicated to planning the reference trajectory.

The characteristics that the system must have to achieve this objective are the following:

- Have the ability to generate n different alternative trajectories that lead the ego vehicle from the current point to a number n of different points.
- Have an algorithm capable of evaluating the actual feasibility of each of the n trajectories.
- Have an algorithm capable of assessing the risk of collision for each of the n trajectories.

In this chapter, the three themes have been analyzed in detail and exposed.

2.1 Trajectory generation

The main task of the assistance system is to generate a trajectory for the ego vehicle to follow. It is specified that the system plans a set of reference trajectory segments that, when combined, form a continuous path from the vehicle starting point to its destination. Each of these segments is planned at a given instant that in the course of this discussion will be called the "planning instant".

At the planning instant, a new reference trajectory segment is produced; two different strategies can be used:

- With spatial sampling: through which each segment is produced in such a way as to

make the vehicle travel a set distance.

- With a temporal sampling: through which each segment is produced in such a way as to make it travel through by the ego vehicle in a certain interval of time; this interval is called the Time Horizon.

In this study, the second solution was adopted for greater compatibility with the available simulation tools.

2.1.1 Trajectory generation methodologies

In the literature, different alternatives can be found regarding the technique for generating the trajectory segment that connects the starting point P0 to an arrival point P1. The solution adopted in this case study consists of planning the trajectory segments using fifth-order polynomial curves, which are optimal for minimizing jerk, as discussed further below.

Among the various available alternatives, the one mentioned was chosen because it is possible to demonstrate that, with the same extremes P0 and P1 and other conditions, it minimizes the following cost function C [5].

$$C = \frac{1}{2}\int_{t_0}^{t_1} (\dddot{x}^2 + \dddot{y}^2)dt$$

In the aforementioned equation, x(t) and y(t) represent the positional coordinates of the vehicle along the horizontal (x) and vertical (y) axes in an inertial reference frame. This cost function is defined as the integral of the square of the jerk, which is calculated with respect to an inertial reference system in the XY plane; this quantity is generally associated with the perception of comfort because it measures the smoothness of acceleration, with lower values indicating a smoother ride.

The equations which describe the trajectory within the same reference system have the following form.

$$\begin{cases} x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\ y(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5 \end{cases}$$

Consequently, generating the correct trajectory segment means identifying the 12 coefficients of the system, starting from the following initial conditions.

$$\begin{cases} x(t_0) = X_0 \\ \dot{x}(t_0) = V_{x,0} \\ \ddot{x}(t_0) = A_{x,0} \\ x(t_1) = X_1 \\ \dot{x}(t_1) = V_{x,1} \\ \ddot{x}(t_1) = A_{x,1} \end{cases}$$

$$\begin{cases} y(t_0) = Y_0 \\ \dot{y}(t_0) = V_{y,0} \\ \ddot{y}(t_0) = A_{y,0} \\ y(t_1) = Y_1 \\ \dot{y}(t_1) = V_{y,1} \\ \ddot{y}(t_1) = A_{y,1} \end{cases}$$

Where:

- t_0 and t_1: are the extremes of the time interval associated with the path of extremes P_0 and P_1.
- $[X, Y], [V_x, V_y]$ e $[A_x, A_y]$ are the components of position, velocity, and acceleration along the axes of the fixed coordinate system respectively.

In the case of using a reference system expressed in Frenet coordinates, the initial conditions that allow us to obtain the coefficients of the equations that describe the variation of the two coordinates $s(t)$ and $d(t)$ are the following:

$$\begin{cases} s(t_0) = s_0 \\ \dot{s}(t_0) = \dot{s}_0 \\ \ddot{s}(t_0) = \ddot{s}_0 \\ s(t_1) = s_1 \\ \dot{s}(t_1) = \dot{s}_1 \\ \ddot{s}(t_1) = \ddot{s}_1 \end{cases}$$

$$\begin{cases} d(t_0) = d_0 \\ \dot{d}(t_0) = d_0' \\ \ddot{d}(t_0) = d_0'' \\ d(t_1) = d_1 \\ \dot{d}(t_1) = d_1' \\ \ddot{d}(t_1) = d_1'' \end{cases}$$

Where:

- $[s_0, \dot{s}_0, \ddot{s}_0, d_0, d_0', d_0'']$: indicates the state of the system at the initial point P0 of the planned trajectory segment.
- $[s_1, \dot{s}_1, \ddot{s}_1, d_1, d_1', d_1'']$: indicates the state of the system at the end point P1 of the planned trajectory segment.

Please refer to paragraph 2.1.2 for a comparison between the two reference systems.

Different solutions present in literature

As anticipated, the fifth-order polynomial is not the only strategy used to calculate the trajectory in path planning systems. Among the most widespread solutions are:

- Sigmoid Curve: which is used specifically to plan trajectories dedicated to lane changes, where the trajectory is defined by vehicle speed and maneuver width to ensure smooth transitions; unlike the adopted solution that has allowed the implementation of a planning methodology based on temporal sampling, valid for all types of plannable trajectories. An example is shown in Figure 2.1.

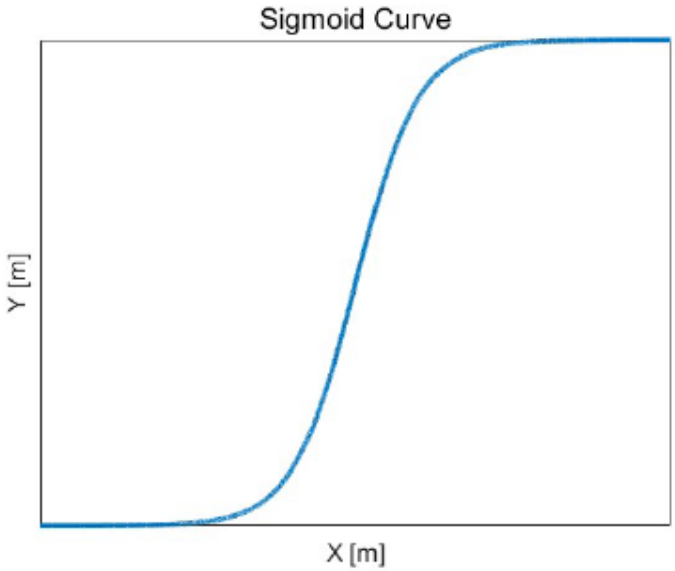

Figure 2.1: Example of a Sigmoid curve.

The equation that describes it is the following.

$$y(x) = \frac{1}{1 + e^{-x}}$$

- State Lattice: this methodology consists in the discretization of the space in which the vehicle can move, with the aim of forming a lattice of points in which it can be positioned. The advantage of adopting this solution lies in the lower computational cost, since the maneuvers between the points are predetermined, as can be seen in Figure 2.2 [6].

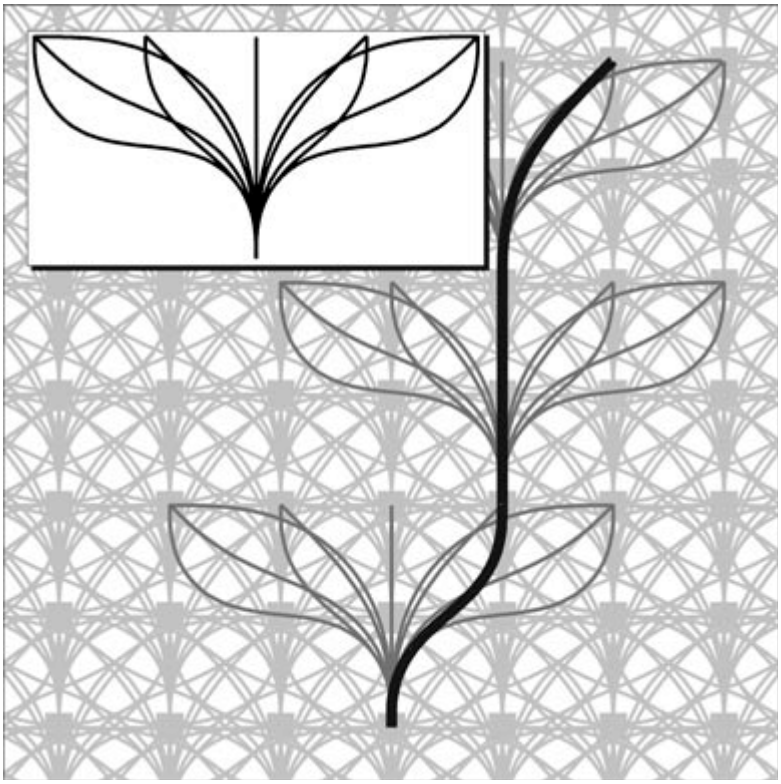

Figure 2.2: Possible trajectories in the State Lattice.

However, this strategy is not optimal, as it limits the vehicle maneuvers even in terms of the steering angle that can be achieved.

2.1.2 Frenet reference system

A well-known approach to the Trajectory planning problem is the so-called Frenet Frame method. It consists in adopting a mobile reference system where the point considered at a given instant is described by the normal and tangent vectors, $\overrightarrow{t_r}$ and $\overrightarrow{n_r}$, with respect to a given point of a curve reference known as center line. The center line is chosen arbitrarily by the designer; it can represent the ideal trajectory to follow, the center line of the lane or the result of a different path planning algorithm [5].

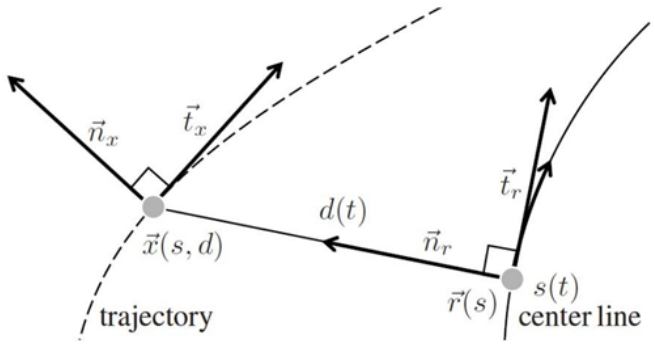

Figure 2.3: Generation of a trajectory with Frénet coordinates.

Consequently, rather than using the Cartesian coordinates $\vec{x}$ directly, the reference system is changed by generating a one-dimensional trajectory, dependent on the point $\vec{r}$ on the center line and on the offset d; as can be seen in Figure 2.3 and according to the following relation:

$$\vec{x}\big(s(t), d(t)\big) = \vec{r}\big(s(t)\big) + d(t)\overrightarrow{n_r}(s(t))$$

Where s indicates the length of the arc of the center line that has been traveled and $\overrightarrow{t_x}$ and $\overrightarrow{n_x}$ are the tangent and normal vectors to the resulting trajectory $\vec{x}\big(s(t), d(t)\big)$ [5]. All trajectories are then described by the coordinates $[s, d]$, which are referred to the pre-established center line.

The advantage of this method is that it simplifies the description and calculation of trajectory points by reducing the complexity of the coordinates, making it easier to define the vehicle path and also to understand from a numerical point of view. Figure 2.4 shows a graphic example of the reference trajectory of the ego vehicle, whose points are defined with respect to a non-inertial system, whose origin coincides with the position of the ego vehicle. Note how numerically complex it is to describe the evolution of the states associated with the blue trajectory, when this is done with the vehicle reference system. In this case, quantifying the vehicle deviation from the reference trajectory is also more complex.

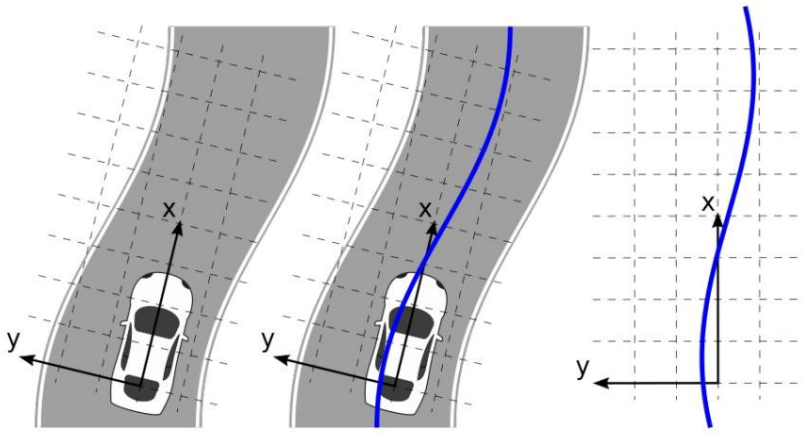

Figure 2.4: Reference trajectory expressed using vehicle coordinates.

In Figure 2.5 we have the representation of the same scenario expressed by Frenet reference system.

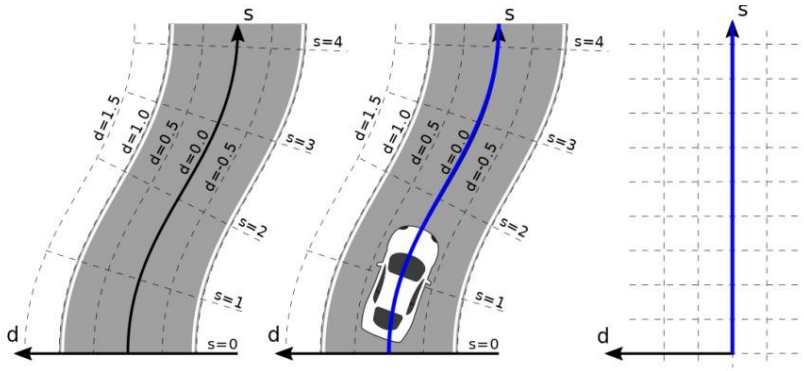

Figure 2.5: Reference trajectory expressed using Frenet coordinates.

It is noted that the reference trajectory of the Frenet system coincides with the centerline of the road and with the reference trajectory of the vehicle. Expressing the position of the ego lane within the carriageway is considerably simpler; since by assuming its lateral deviation from the centerline, the coordinate d already quantifies this quantity. During this study, the centerline of the carriageway was always adopted as the reference trajectory. Therefore, using this method, it is easy to implement a sampling strategy of the terminal points of the segments of the reference trajectory even in the case of an overtaking maneuver. Since to identify a point P_1 where the trajectory with which to perform the lane change ends, it is

sufficient to express the quantity Δd by which the vehicle is to be moved with respect to the current lane. For a better understanding of these theoretical issues, an example of the construction of the lane change trajectory has been created in the next paragraph.

2.1.3 Trajectory generation tools in the environment of simulation

In the MATLAB & Simulink software libraries, a set of functions dedicated to the topics covered and consistent with the choices made in terms of reference system and trajectory generation methodology is available.

global2frenet

The global2frenet function allows to switch from the global reference system to the Frenet reference system, by providing the function with the waypoints of the curve chosen as reference [7]. The change of coordinates is the following:

$$[x, y, \psi, \kappa, \upsilon, \alpha] \rightarrow [s, \dot{s}, \ddot{s}, d, d', d'']$$

With:

- $[x, y]$: global coordinates of the given point P.
- ψ: yaw angle.
- κ: curvature of the trajectory at the given point P.
- υ and α: velocity and acceleration respectively.
- s: length of the reference curve arc traveled.
- d: distance in the normal direction from the reference curve.

trajectoryGeneratorFrenet

In the libraries dedicated to the Navigation Toolbox there is a set of functions dedicated to

the generation of trajectories, which are based on the principles explained above. The trajectoryGeneratorFrenet function allows to generate trajectories that join the point P_0, currently occupied by the ego vehicle, with a set of n points in which to end the n trajectory segments produced. The generation of trajectories is performed using fifth-order polynomial curves, to which a specified time horizon $\mathrm{Th} = (t_1 - t_0)$ is associated .

Trajectory generation example

In order to make the explanation more immediately understandable, an example case study has been created; which is based on a scenario that, with the modifications progressively introduced in this chapter, has led to the definition of the scenario analyzed in the fourth chapter and called "Overtaking with Oncoming Vehicle".

The initial configuration of the scenario includes the presence of only two vehicles:

- Vehicle ego: vehicle being simulated which in this case has a speed $\mathrm{vego} = 20\mathrm{m/s}$.
- Lead vehicle: vehicle that precedes the ego vehicle while travelling in the same lane which in this case has a speed $\mathrm{vlead} = 15\mathrm{m/s}$.

They travel along a 300m long straight road, in the same lane, in the same direction of travel and are positioned at an initial distance of 40m.

It is shown in the Figure 2.6 a representation of the scenario at the initial time t = 0 s.

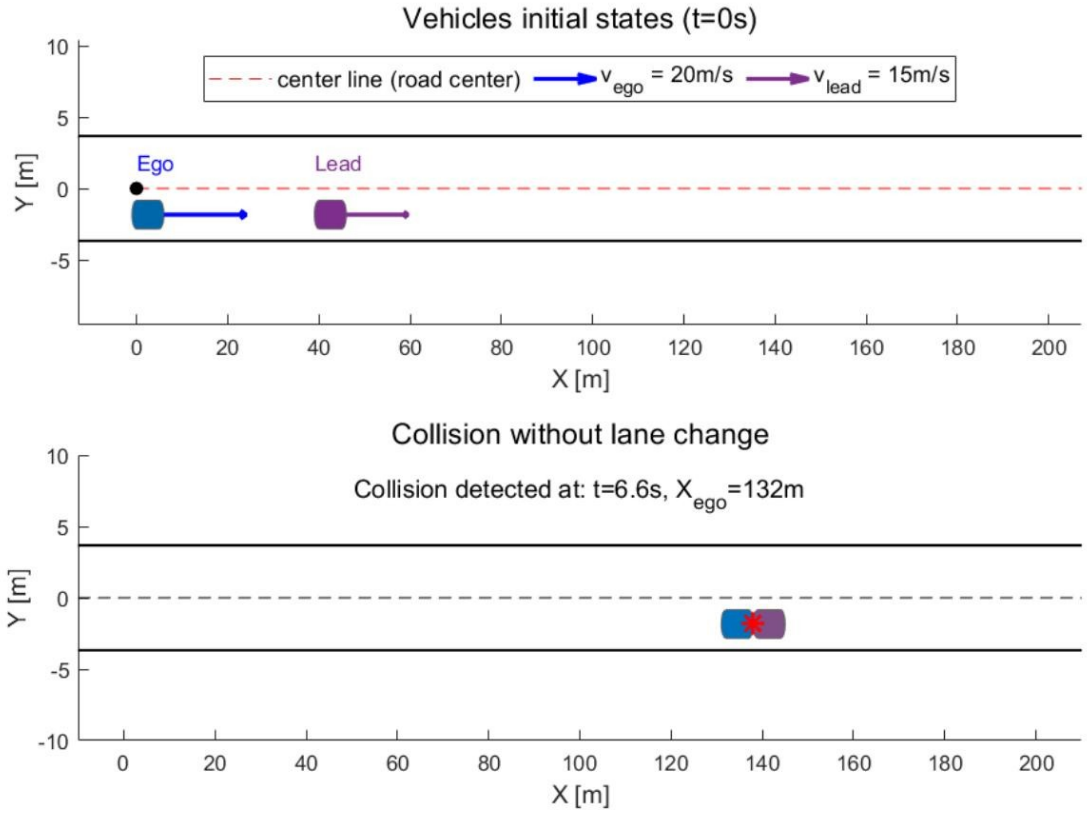

Figure 2.6: Initial states t = 0 s of the “lead vehicle overtaking" scenario.

The second graph shows the potential collision that would occur if a Cruise Control (CC) mode were maintained, according to which the ego vehicle maintains its trajectory and cruising speed unchanged.

The intervention of a welfare system such as the one being studied in this study is

that of identifying two different solutions:

- LCF mode: A reference trajectory is identified as a trajectory which involves maintaining the lane and reducing the speed of the ego vehicle by an amount sufficient to maintain a safe distance from the lead vehicle.
- LC mode: A reference trajectory is identified as a trajectory that brings the ego vehicle into a different lane than the one currently occupied. The assumption made in this study is to plan trajectories of changing lanes at a constant speed ṡ.

After presenting the case study, we then move on to planning a set of trajectories using the functions mentioned, in order to provide practical feedback on the methodology implemented within the model presented in the third chapter.

For each of the two modes, it was chosen to plan two different trajectory segments, one for each time horizon value chosen and expressed in seconds.

$$\text{TimeHorizons} = [1.5, 3]$$

The first step is to define the way points of the reference curve of the Frenet coordinate system . Defined through Cartesian coordinates in the $X - Y$ plane . As anticipated, in this discussion the reference curve chosen is always the center line of the carriageway, represented in red in Figure 2.6.

$$\text{wayPoints} = [0, 0;\ 300, 0]$$

After that, an object containing a curve connecting the way points is created using the dedicated function referencePathFrenet.

$$\text{refPath} = \text{referencePathFrenet}(\text{wayPoints})$$

This curve is necessary to create the object called "connector" which hosts all the information related to the reference curve of the Frenet coordinate system. It is created through the trajectoryGeneratorFrenet function.

$$\text{connector} = \text{trajectoryGeneratorFrenet}(\text{refPath})$$

The last step consists in obtaining the set of trajectories, expressed in global coordinates $[x, y, \psi, \kappa, \upsilon, \alpha]$, through the connect function. Its task is to connect the initial state with the set of terminal states; these trajectories are traversed in a time equal to the specified time horizon.

The arbitrarily created set of terminal states is composed of the expressed states in Frenet coordinates $[s, \dot{s}, \ddot{s}, d, d', d'']$ to which a given time horizon is associated. It has been structured in Table 2.1:

Mode	Terminal state	Time horizon
LCF	$[\mathrm{NaN}, \dot{s}_{\mathrm{lead}}, 0, 0, -\mathrm{laneWidth}/2, 0, 0]$	1.5 s
LCF	$[\mathrm{NaN}, \dot{s}_{\mathrm{lead}}, 0, 0, -\mathrm{laneWidth}/2, 0, 0]$	3 s
LC	$[\mathrm{NaN}, \dot{s}_{\mathrm{ego}}, 0, 0, +\mathrm{laneWidth}/2, 0, 0]$	1.5 s
LC	$[\mathrm{NaN}, \dot{s}_{\mathrm{ego}}, 0, 0, +\mathrm{laneWidth}/2, 0, 0]$	3 s

Table 2.1: Terminal states of the "lead vehicle overtaking " scenario.

It is specified that the value s of each terminal state has not been specified, in this case it is calculated autonomously by the function starting from the values of time horizon and velocity. The initial state is numerically expressed as:

$$\mathrm{initialState} = [s_0, \dot{s}_{\mathrm{ego},0}, 0, -\mathrm{laneWidth}/2, 0, 0]$$

Where $\dot{s}_{\mathrm{ego},0}$ coincides with $v_{\mathrm{ego},0}$. It is specified that the coordinate d associated with the center of the two lanes is equal to (Table 2.2):

Lane	d
Right	$-\mathrm{laneWidth}/2$
Left	$+\mathrm{laneWidth}/2$

Table 2.2: Coordinates of individual lanes - "overtaking the lead vehicle" scenario.

Where "right" and "left" have been defined from the point of view of the ego vehicle and "laneWidth" is the width of a single lane.

Using the connect function, the resulting set of trajectories is obtained, expressed in global coordinates.

$$\mathrm{trajGlobal} = \mathrm{connect}(\mathrm{connector}, \mathrm{initState}, \mathrm{terminalStates}, \mathrm{timeHorizons})$$

A graphical representation of the trajectories obtained starting from the set of terminal states described previously is reported in Figure 2.7.

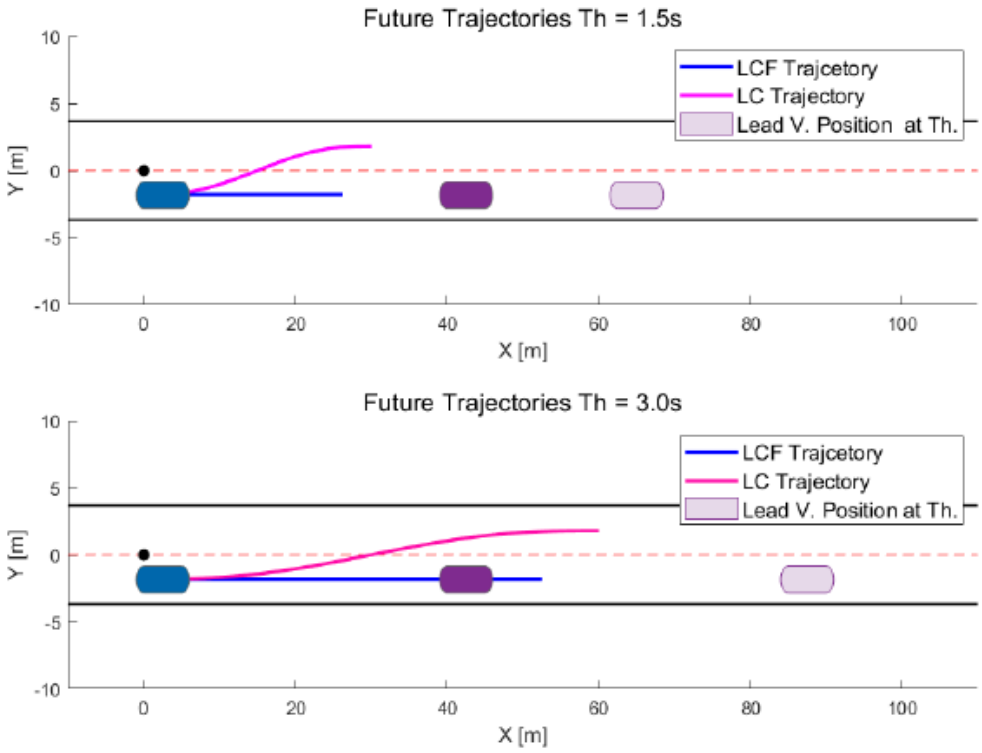

Figure 2.7: Set of alternative trajectories of the "overtaking the lead vehicle" scenario.

Note that a representation of the lead vehicle position at the given time horizon has been inserted, reported with semi-transparent coloring, so that it can be compared with the last point of the represented trajectories. Also, note that the terminal point of the LC trajectories has a higher s coordinate than the trajectories calculated with LCF mode, since the ego vehicle in LC mode has a higher speed $\dot{s}$.

Through this example it can be understood how simple it actually is to define the vehicle states in Frenet coordinates.

2.2 Feasibility assessment of the trajectories alternatives

The next step after creating a set of possible trajectories to follow is the evaluation of the results obtained. What is generally defined as a reference trajectory does not actually correspond to the best among the infinite possible combinations but to the one that, starting from a set composed of n alternatives and respecting all the imposed checks, minimizes a given function of cost.

The first check that is generally carried out concerns the practicability of the trajectory; where this term refers to the actual capacity of the vehicle to be able to meet the performance requirements necessary to follow, as faithfully as possible, the reference provided.

It must be taken into account that the quantities describing the vehicle behavior, which are associated with the sampled reference trajectories, are the result of numerous simplifications in dynamic terms and are often calculated with purely kinematic methodologies. This is related to a recurring theme in the world of simulations, as one always seeks the correct balance point between accuracy of the results, computational cost and simplicity of the introduced modeling.

Basic kinematic check

In the simplest case, the feasibility check can be limited to the insertion of some limit thresholds to be respected for all sampled points of a given trajectory.

Generally, the parameters on which the analysis is conducted are associated with quantities such as: longitudinal and lateral acceleration, jerk, yaw rate, trajectory curvature and so on. It is necessary to take into account the fact that these are estimates, as these values are calculated in anticipation of the actual response of the vehicle.

If the limit values were the result of a characterization of the vehicle response to different inputs, this methodology could be used to discard all the alternatives whose associated values are largely outside the tolerated threshold. The advantage of this solution is the speed and low computational cost of the verification. However, in more extreme operating conditions,

the vehicle dynamics would massively influence all the calculated quantities; therefore, for such conditions it is necessary to implement a more advanced feasibility verification technique.

2.2.1 Nonlinear analysis of vehicle stability

In working conditions where the required performance is very high, the vehicle may encounter the phenomenon of directional instability. Directional instability occurs when the vehicle loses its ability to maintain a steady course, often due to oversteer or changes in tire friction. Consequently, for advanced systems it is important to include a strategy that is able to evaluate the performance request of alternative trajectories, deleting from the set those that would lead to instability, therefore, to a condition of very high risk in terms of safety for the vehicle and its occupants.

For this condition to occur, the vehicle must be in good condition of oversteer operation. In this case it is possible that the vehicle might present unstable equilibrium condition, requiring an opposite sign steering angle to regain stability. In the most extreme cases, the vehicle states may diverge, leading to a condition in which the trajectory has a zero radius of curvature, commonly called "head-to-tail".

Phase plan method

A well-known approach to the problem is the "phase plane method". This is a graphical method focused on vehicle dynamics studies in highly nonlinear operating ranges. This method is commonly used for the problem exposed since the stability criteria derived from linear dynamics models cannot be used, since the phenomenon of directional instability is centered on the non-linear operating field of the tires [8].

According to this theory, a series of parameters must be selected, appropriately chosen since they are influenced by the lateral dynamics of the vehicle, to trace the evolution of the trajectories in a given plane. The parameters generally used in the studies reported in the

literature are the following:

- v_y: lateral velocity.
- $\dot{\psi}$: yaw rate.
- β: trim angle, defined as $\beta = \mathrm{atan}(v_y/v_x)$.
- $\dot{\beta}$: variation of the trim angle over time.
- α_f and α_r: mean slip angles of the two axles.

The phase planes, within which it is possible to describe the so-called phase trajectories, which are graphical representations of a system state variables, are the following:

- $v_y - \dot{\psi}$
- $\beta - \dot{\psi}$
- $\beta - \dot{\beta}$
- $\alpha_f - \alpha_r$

Although each pair of parameters describes the same information in graphical form, there are some differences attributable to the following characteristics:

- Measurability: since some parameters such as $\dot{\psi}$ can be measured directly, unlike estimates that must be appropriately calibrated and carried out on other parameters, such as β.
- Relevance: as there are parameters such as β that have a greater influence on the stability conditions of the vehicle.
- Sensitivity: which is measured in terms of variation of the area of stability following changes in operating conditions.

In most cases, the $\beta - \dot{\psi}$ and $\beta - \dot{\beta}$ planes are used [8]. It must be specified that the tire characterization coupled with this type of analysis must be nonlinear. Furthermore, it must be specified that this type of analysis is conducted through 2DOF models that isolate the

lateral dynamics of the lane, making these planes phases dependent on a smaller number of parameters.

The aim of this method is to obtain maps in the above mentioned planes which graphically show the evolution of the parameters associated with different trajectories, each of which has different initial states. The evolution of the states, starting from a given initial condition, is represented by a vector field.

An example of a $\beta - \dot{\psi}$ phase plane is shown in Figure 2.8.

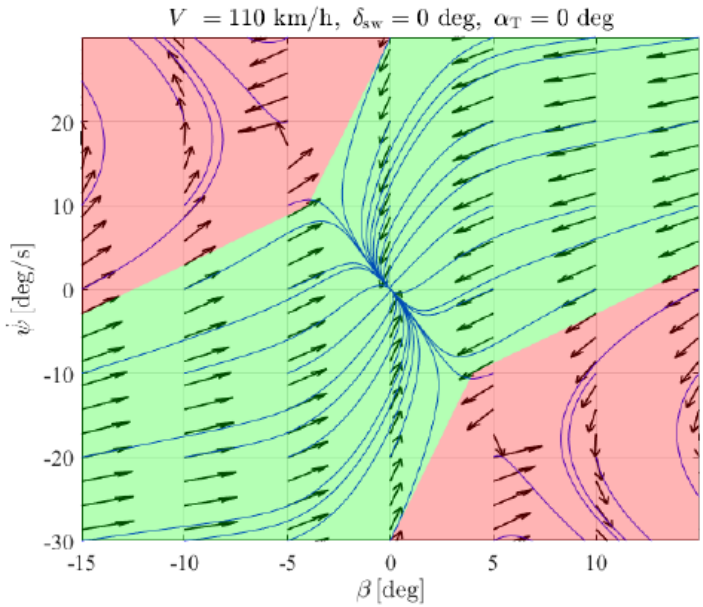

Figure 2.8: Phase plane $\beta - \dot{\psi}$.

To create the map shown in Figure 2.8, it was necessary to conduct a number n of simulations by varying the starting states β_0 and $\dot{\psi}_0$ and recording the evolution of these quantities. All simulations were carried out with other conditions such as speed, steering angle, asphalt friction coefficient and road surface inclination remaining the same.

Thanks to the graphic representation of the vector field, it can be seen that some of the trajectories described within the plane lead to a convergence of the states towards the stable equilibrium point. The green portion of the graph containing the trajectories that converge towards this point is the stability zone of the vehicle, also defined "Domain of attraction".

The areas that must absolutely be avoided are those in red that lead to the divergence of states, also called zones of instability. The graphical approach consists in identifying the limits that separate the stability zones from the instability zones. Different approaches to the problem have been identified in the literature; the Diamond method is mentioned, which consists in delimiting the area surrounding the stable equilibrium point with a rhombus, and the Two line method , which consists in identifying two lines that contain only points belonging to the domain of attraction. What is important to underline is that there is no universal method for delimiting the areas, however, they are strictly dependent on the position of the two unstable equilibrium points.

Identifying them means being able to exploit the delimitation methodology considered most appropriate and implementing the verification in the reference trajectory selection process. The position of the two unstable equilibrium points is strictly dependent on several parameters [9]:

- V_x: vehicle speed. By increasing the speed, with other parameters being equal, the stability zone reduces its width and the unstable equilibrium points acquire a smaller distance from the stable equilibrium point. Figure 2.9 shows a comparison between two maps obtained at $50\ \mathrm{km/h}$ and $110\ \mathrm{km/h}$ respectively.
- μ: road friction coefficient. As the grip of the tires worsens, the stability zone reduces its width and the unstable equilibrium points acquire a smaller distance from the stable equilibrium point.
- δ_f: steering angle of the wheels. By changing the angle δ_f the stable zone becomes asymmetric. The limiting condition of this phenomenon is the total disappearance of the domain of attraction for high steering angle values. A graphical confirmation of the phenomenon is found in Figure 2.10, in which the maps obtained in the phase plane $\beta - \dot{\psi}$ with steering angle δ_f equal to 0°, graph on the left, and 5°, graph on the right, have been found.

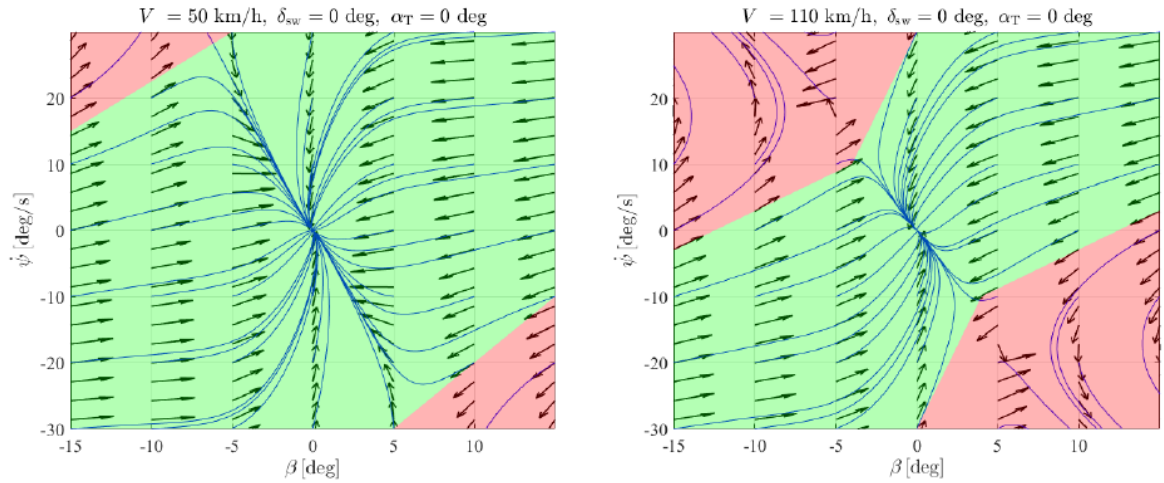

Figure 2.9: Influence of vehicle speed on the $\beta - \dot{\psi}$ phase plane

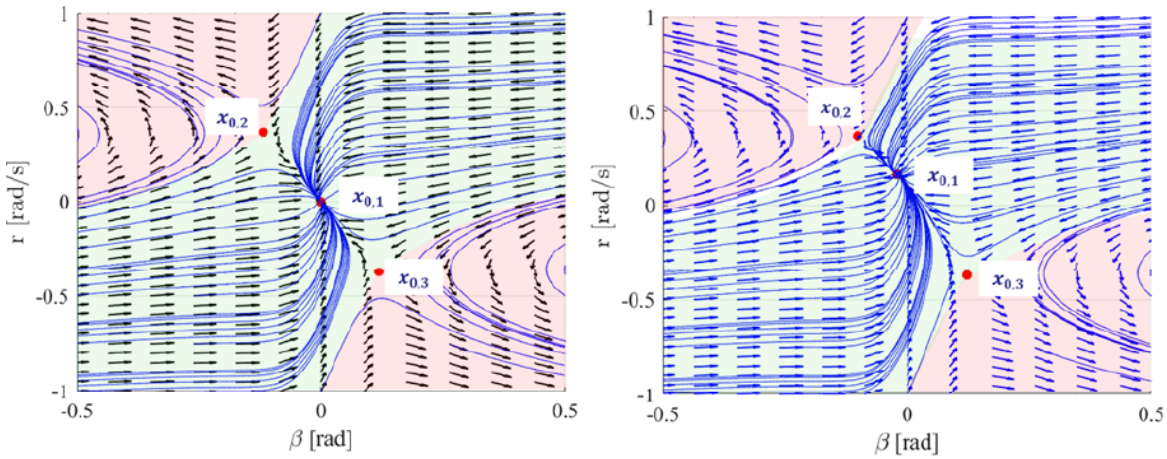

Figure 2.10: Influence of the vehicle steering angle on the $\beta - \dot{\psi}$ phase plane. δ_f graph on the left 0°, δ_f graph on the left 5°.

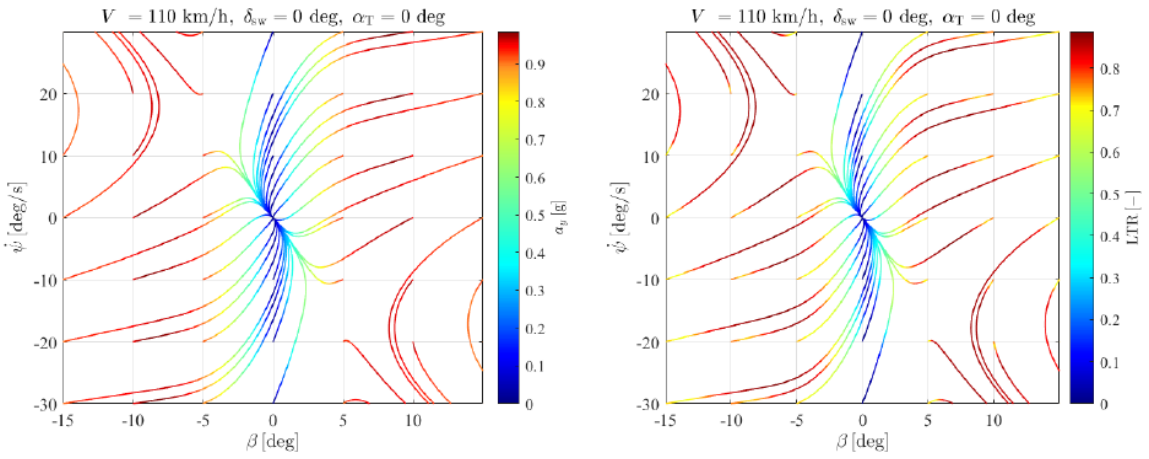

Figure 2.11: Correlation of the lateral acceleration ay and the LTR index with the operating points plotted on the $\beta - \dot{\psi}$ phase plane.

In Figure 2.11 the results of different simulations on the phase plane have been reported and, through a color gradient, the variation of lateral acceleration ay and vertical load transfer index (LTR) has been reported respectively, which gives an accurate indication of the load vertical transfer during the maneuver.

$$\mathrm{LTR} = \frac{\mathrm{Fz_L} - \mathrm{Fz_R}}{\mathrm{Fz_{tot}}}$$

Where $\mathrm{Fz_L}$, $\mathrm{Fz_R}$ and $\mathrm{Fz_{tot}}$ are the vertical forces acting respectively on: left pair of tires, right pair of tires and all four tires. From the examination of the graphs shown, it emerges that a greater distance from the equilibrium point corresponds to a higher criticality of the state.

Among the methodologies chosen to estimate the limits of the zones, more conservative approaches can be used , since in certain operating points within the domain of attraction, which are in any case distant from the stable equilibrium point, the transient that brings the vehicle into a stable condition turns out to be excessively slow.

2.3 Assessment of potential collisions for alternative trajectories

A further check that must necessarily be included in the reference trajectory selection process is the one regarding collisions with other vehicles present in the scenario. This check is based on an algorithm able to estimate the evolution of the trajectories of the other actors and on the knowledge of their relative dimensions.

Assuming that such data are available, a possible solution is to make a moment- by-moment comparison between the trajectories of the bodies present in the scenario. Having the descriptive parameters of the dimensions of the various actors, it is possible to exclude all the trajectories that contain an overlap of the bodies at a given time instant.

2.3.1 2D capsule method

In MATLAB & Simulink environment the already mentioned Navigation Toolbox provides the tools that allow to perform the collision check starting from the cited data. The set of functions available is well optimized as it allows to fulfill the purpose in a quick and simple way.

The main function is called dynamicCapsuleList, which creates two separate lists of objects. These lists contain respectively the bodies of the ego vehicle and those of the other actors present in the scenario; these objects are represented as capsules in a 2D space. Each of them is identified by three key elements [10]:

1. An index with which to identify the specific vehicle within the list.
2. A matrix containing the vehicle states in global coordinates $[x, y, \psi]$.
3. A structure containing the descriptive parameters of the vehicle geometry, such as the length of the capsule representing it, its radius and a 3x3 matrix representing the transformation useful for moving the local origin of the capsule at a point other than the default one (black point with coordinate (x, y).

The graphic representation of the capsule is shown in Figure 2.12.

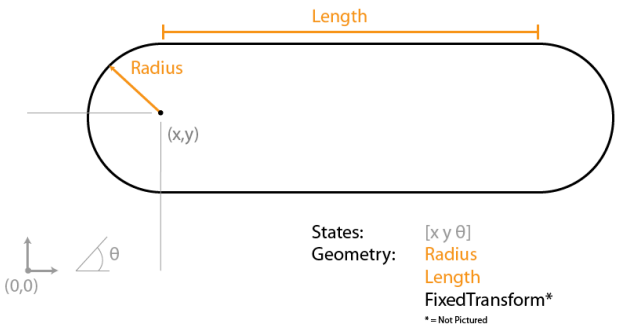

Figure 2.12: Geometry of 2D capsules [10].

In the following subsection, an example specifically created to facilitate the understanding of the dynamicCapsuleList tool and to provide a tangible demonstration of the collision checking process has been reported . The starting scenario is the one examined in the first paragraph in order to give continuity to the treatment.

Collision assessment example

The created example is divided into three different sub-cases, through which the scenario has been progressively modified. All vehicles have the following geometric characteristics:

Length	5 m
Radius	1 m
Transformation matrix	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Table 2.3: Capsule geometry - Ex. Dynamic Capsule-based List

Case 1 - checking for possible collisions starting scenario

The characteristics of the starting scenario are briefly reported, whose graphic representation is appreciable in Figure 2.6:

- A straight road with two lanes, each 3.6m wide.
- An ego vehicle that travels along one of the two lanes at a constant speed starts at $20\mathrm{m/s}$
- A vehicle that precedes the ego vehicle traveling along the road in the same lane and with the same direction at a constant speed of $10\mathrm{m/s}$.
- An initial distance between the two vehicles of $40\mathrm{m}$.

The objective is to check for the possible presence of collisions, leaving the initial speeds and trajectories of the vehicles unchanged. In this way, it is possible to understand the methodology to be used with the available tools. The process is carried out by points.

1. The initial phase of this method is to create the object containing the initialized structure of the two actor lists by calling the function dynamicCapsuleList:

obsList = dynamicCapsuleList;

   After that, an arbitrary Δt is determined as the time sampling unit and the time vector is created. After calculating the number of elements of the vector, which corresponds to the number of time steps of the simulation, the information is recorded inside the dynamicCapsuleList object.

2. The ego vehicle must be added to the appropriate list in the dynamicCapsuleList object, providing all the information that describes it. The linspace function was used to define the states of the ego vehicle in the first scenario, in order to create a "states" matrix with a constantly increasing abscissa and of an amount consistent with the speed of the vehicle. The geometry was created by inserting the three parameters described in Table 2.3 into a single "geom" structure. Once the three elements were obtained, it was possible to create the structure dedicated to the ego vehicle capsule.
   $\mathrm{egoCapsule} = \mathrm{struct}('\mathrm{ID}', \mathrm{egoID}, '\ \mathrm{States}', \mathrm{states}, '\ \mathrm{Geometry}', \mathrm{geom})$

addEgo(obsList, egoCapsule)

3. In a similar way, we proceed with the insertion of the obstacles, or other actors present in the scenario, into the appropriate list. In this case, the evolution of the lead vehicle states was also described using the linspace function . The lead vehicle capsule was created and inserted into the list with the following functions.

$$\text{obsCapsule1} = \text{struct}('\text{ID}', \text{obsID1}, '\ \text{States}', \text{obsState1}, \ldots$$
$$\ldots '\text{Geometry}', \text{geom});$$
$$\text{addObstacle}(\text{obsList}, \text{obsCapsule1});$$

4. Finally, the collision check of the vehicles during the time interval described is performed through the following simple command:

$$\text{collisions} = \text{checkCollision}(\text{obsList})';$$

The "collisions" output is a vector of boolean variables associated with each single time step and the value 1 corresponds to the collision state, i.e. an overlap of the areas occupied by the vehicles. Consequently, a given trajectory will be discarded during the check if there is even a single element true in the output of the checkCollision function.

The evolution of the system states obtained by 2D capsule representation is reported in Figure 2.13.

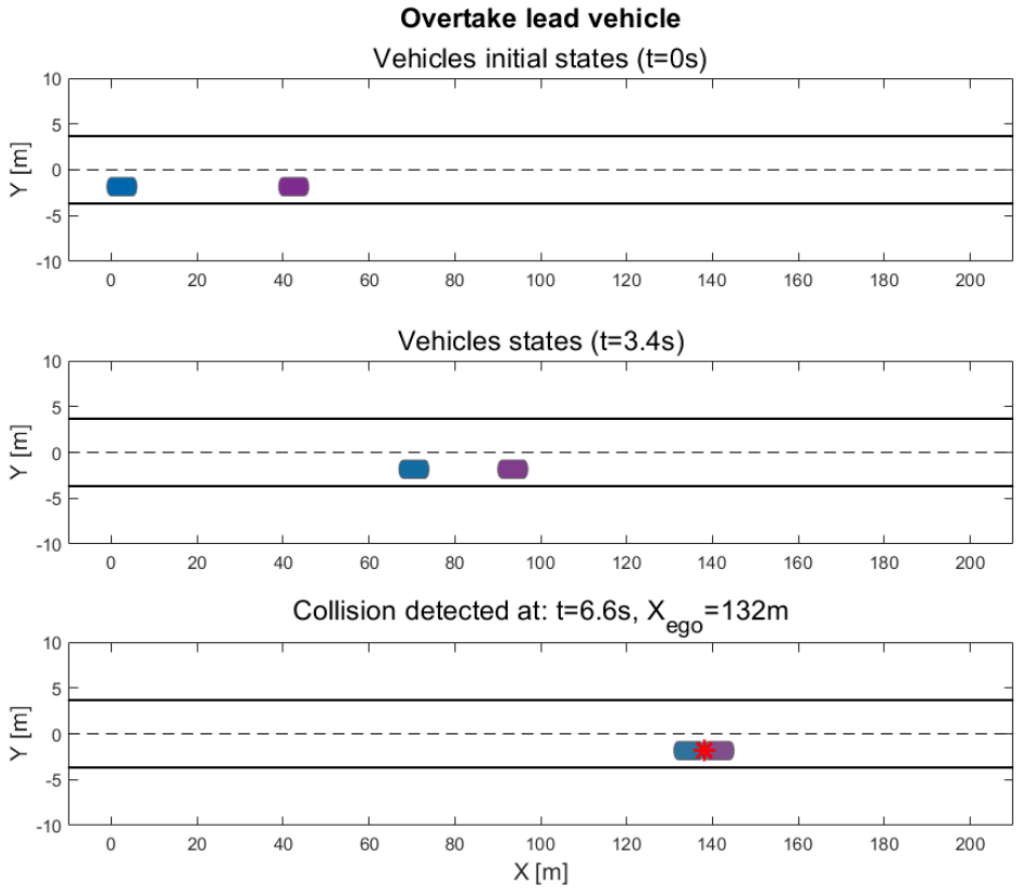

Figure 2.13: Evolution of vehicle states over time and collision check - " lead vehicle overtaking" scenario.

The last graph shows the collision detection due to the difference in speed of the two vehicles. It is specified that the initial distance of the two vehicles has been referred to the distance between the points to which these vehicles are associated, i.e. the respective center of the rear axle; however, to collide the vehicles must approach a distance less than that reported as the initial distance, since it is sufficient for the front part of the ego vehicle to come into contact with the rear part of the lead vehicle.

Case 2 - verification of possible collisions with overtaking trajectory

Assuming a system reaction to the detection of an imminent collision like the one represented in the first case, the scenarios of the second and third cases of this example were constructed

so that the ego vehicle performs an overtaking maneuver to the detriment of the lead vehicle.

The identified lead vehicle remained unchanged in all three elements that describe its modeling in capsule. Instead, the ego vehicle underwent modifications to the state matrix, as its trajectory is different. To develop a trajectory suitable for overtaking, the connect method described previously was used. The logic with which the lane change and return trajectory was constructed was the following:

- A first lane change is performed at instant $t = 0s$ with time horizon $Th = 3.0s$.
- The second lane change maneuver is performed only after the lead vehicle has completely passed. Also in this case, a time horizon $Th = 3.0s$ was adopted.
- For simplicity, it was chosen to keep the speed of the ego vehicle constant along the x coordinate, as well as along the s coordinate with the Frenet reference system.

After updating the "states" matrix, it was stored in the structure representing the relative capsule.

In this case no collisions were found, as can be verified from the graphs in Figure 2.14.

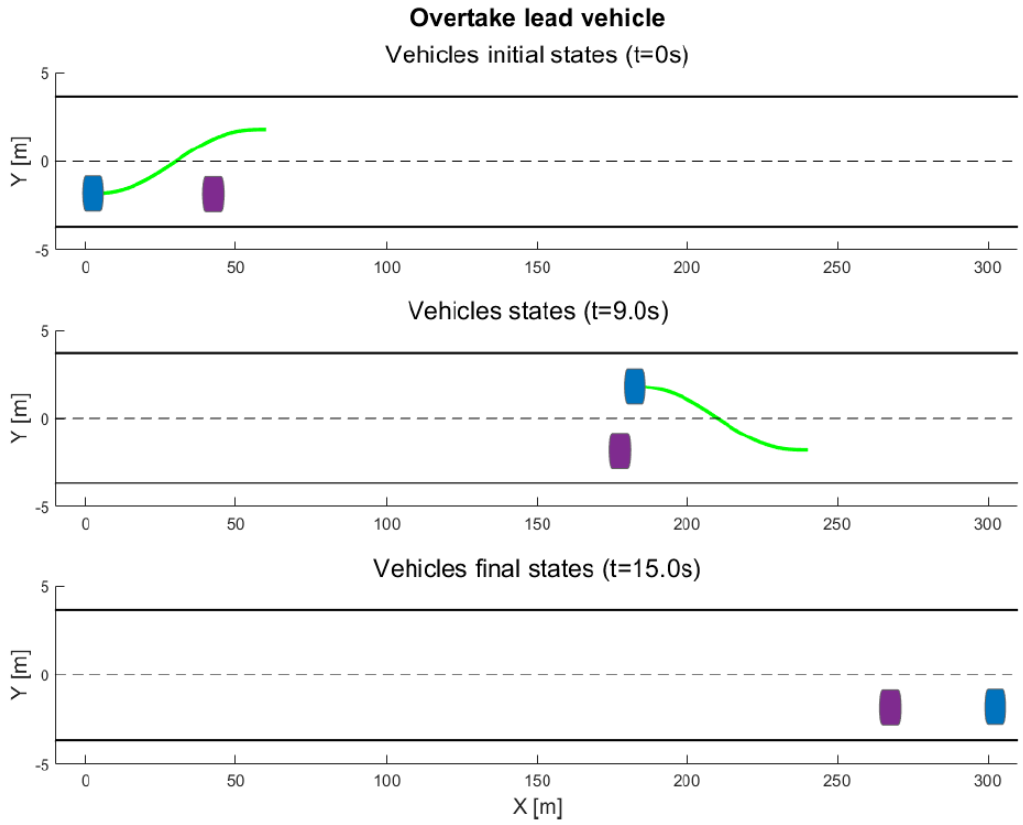

Figure 2.14: Evolution of vehicle states over time and collision check "lead vehicle overtaking with overtaking" scenario.

Case 3 - checking for possible collisions with a vehicle travelling in the opposite direction

In this third and final version of the example, the scenario has been further modified, increasing its complexity by adding a further vehicle travelling in the lane with the opposite direction of travel. It has been called the oncoming vehicle and has a speed of $10\mathrm{m/s}$ and an initial distance from the ego vehicle of $\Delta s = 200\mathrm{m}$.

Figure 2.15 shows a representation of the initial state of the scenario in the absence of collisions.

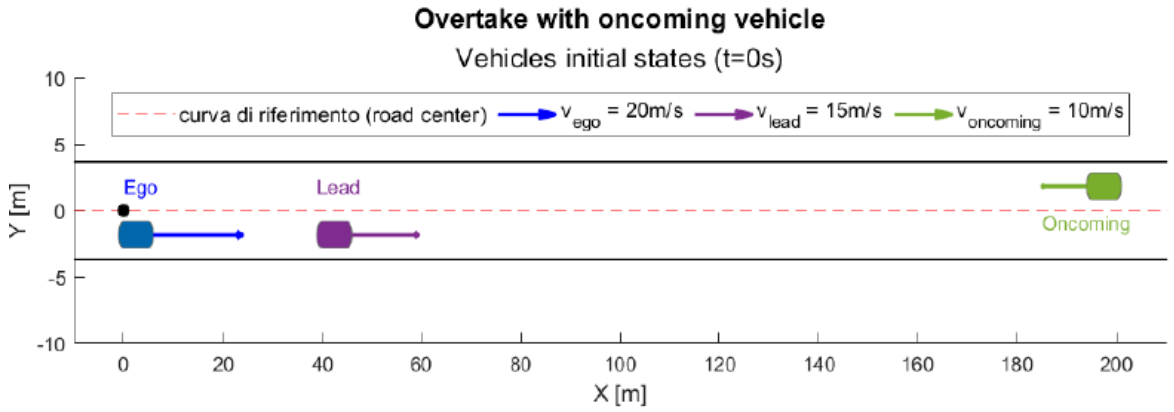

Figure 2.15: Initial states t = 0 s of the "overtaking with vehicle in opposite direction" scenario.

In this case the trajectories of the ego and lead vehicles are the same as in the previous case. Figure 2.16 shows the result of the collision analysis.

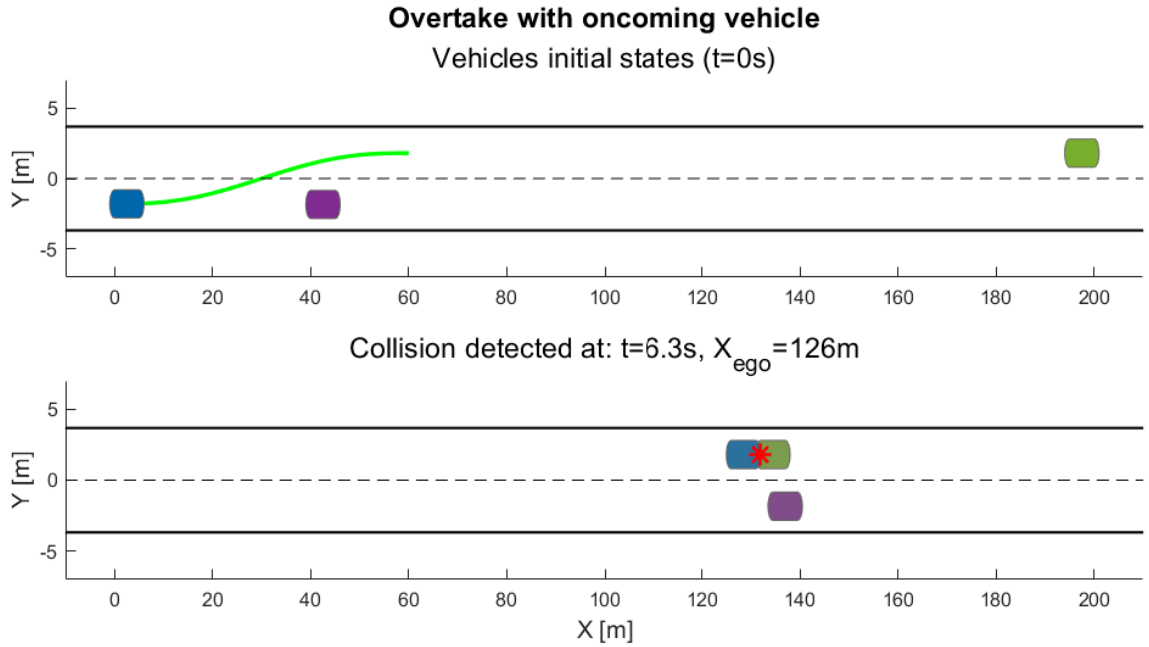

Figure 2.16: Evolution of vehicle states over time and collision verification - "overtaking with vehicle in opposite direction" scenario.

As can be seen, a collision was detected during the execution of the overtaking maneuver by overlapping the areas occupied by the capsules.

The current scenario configuration has been analyzed through complex simulations in the

fourth chapter, through which strategies have been implemented that are able to recognize the presented criticality and to modify the trajectory and/or speed of the ego vehicle in such a way as to avoid collisions.

This case study was developed with the aim of highlighting the criticality presented since in order to overcome it, it is necessary to equip the vehicle with an assistance system such as the one in this case study.

Chapter 3

Simulation Environment

The industrial environment continuously requires a constant acceleration of research and development processes; therefore, it is fundamental to have the correct tools to reduce design times, maximize efficiency and accuracy of results. The automotive sector is no exception, with companies increasingly relying on accurate virtual environments for design processes. Computer simulations have drastically reduced the costs and times of solving complex dynamics and control problems in the automotive field.

The new frontiers of simulation in the automotive field are leading to an ever-increasing integration of these tools within the vehicle. In highly automated vehicles, it is necessary to have on-board systems capable of analyzing sensor signals, determining appropriate commands, and predicting the evolution of scenarios and vehicle behavior.

The design and development phases of an advanced assistance system, such as the one studied here, are based entirely on simulation environments. These forecasts, relating to scenarios and vehicle behavior, are integral and fundamental to the healthcare system.

3.1 Simulink Model of the Assistance System

To conduct research focused entirely on the study of the overtaking maneuver and its potential for autonomous execution, instead of developing the simulation tool from scratch, it was decided to start from the basis of a model developed by MathWorks and available in its libraries. In the initial phase of the study, priority was given to the model components dedicated to trajectory planning, with particular attention to the possibility of modifying calibration parameters and their operational logic. Therefore, the choice initially fell on the development of a model that included only the planner; in such a way, this allowed for a better evaluation of the outputs produced by the assistance system, that is the generated trajectories, deleting from the results the influences of other components, such as the controller and the vehicle model.

The model used is part of the set provided with the Automated Driving Toolbox by MathWorks for MATLAB & Simulink software. This tool is dedicated to the design, simulation, and testing of ADAS and autonomous driving systems. It allows work on projects focused on vehicle sensors, controllers, or path planning, as in this case [11]. In addition to providing vehicle models, such as the one in this paragraph, the Automated Driving Toolbox is particularly efficient for the study of this topic, as it includes applications dedicated to some aspects of simulation in the Automotive field, such as the Driving Scenario Designer [12]. Moreover, compatibility with multiple formats, such as OpenDRIVE, allows the importation of files that are useful for creating specific scenarios. In order to leverage all the tools related to this study, the Navigation Toolbox [13] was also utilized. This is a tool, also developed by MathWorks, for the analysis of motion planning, simultaneous localization, and inertial navigation.

In this section, the aforementioned model will be analyzed in detail. However, during the study, it was integrated into a more complex and complete model.

3.1.1 Highway Lane Change Planner and Controller

The model used in this research is the Highway Lane Change Planner and Controller, available within the Automated Driving Toolbox of MATLAB [14]. This model allows the ego vehicle to change lanes autonomously within highway scenarios and manages different driver behaviors for the same scenario.

For example, depending on the context, the vehicle can activate the cruise control (CC), lead car following (LCA) or automatic overtaking (LC) functions. The model was developed in MATLAB & Simulink environment and its general structure can be appreciated in Figure 3.1. Each of the represented blocks was developed in subsystems and the functioning of the entire model is regulated by MATLAB scripts.

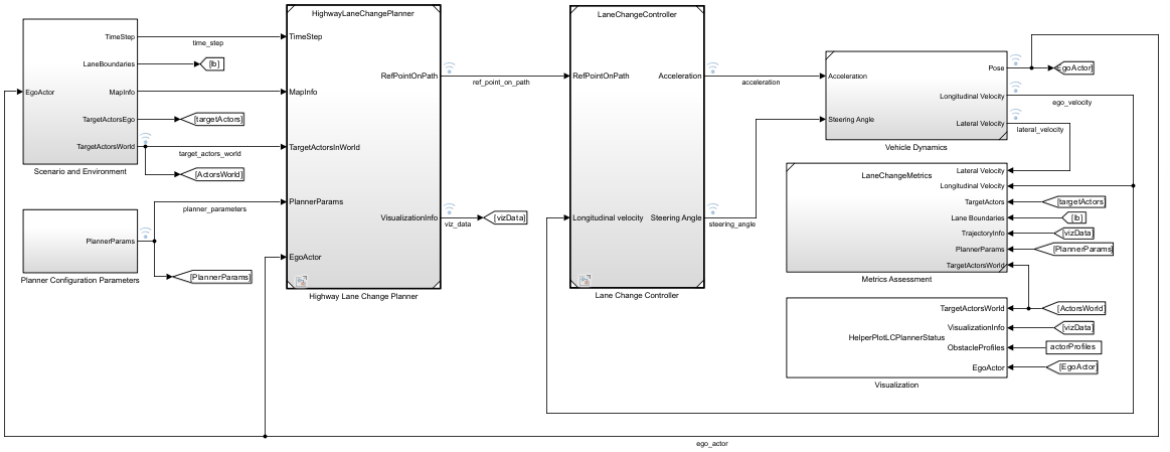

Figure 3.1: Highway Lane Change Planner and Controller general structure.

The main input of the system is the scenario, which is loaded by means of a specific script and provides information about the road and the main actors, such as: ego vehicle, other vehicles present in the scenario, barriers, obstacles and so on. The characteristics of this driving-scenario object, the related methodologies for creating and inserting it into the model have been discussed in paragraph 3.3.

As can be seen in Figure 3.1, the model is composed of various main subsystems, each of which has been analyzed in detail below.

A brief description of the individual subsystems and the operating logic of the model is

provided.

- Scenario and Environment: Subsystem dedicated to reading the driving-Scenario object received as input, useful for producing the signals necessary for the simulation.
- Planner Configuration Parameters: Subsystem dedicated to translating the parameters needed to configure the trajectory planner into signals; this data is imported directly from the WorkSpace.
- Highway Lane Change Planner: This block contains the core of the assistance system. The signals from the previous block are analyzed and compared with the data describing the motion of the ego vehicle to establish the trajectory to follow.
- Highway Lane Change Controller: It contains the MPC controller and it compares the output data from the previous block and the current longitudinal velocity of the vehicle to establish the acceleration and steering angle needed to follow the trajectory established by the planner and maintain the reference speed.
- Vehicle Model or Vehicle Dynamics: The main core of the Vehicle Dynamics subsystem is a Bicycle Model block which is used to model the ego vehicle. The aforementioned block implements a rigid model of the body of a two-axle single-track vehicle to determine longitudinal, lateral and yaw motion.
- Metrics Assessment: Subsystem dedicated to the evaluation of the signals produced during the simulation. It generates alarms or interrupts the simulation if the imposed limit parameters are not respected.
- Visualization: Subsystem dedicated to the creation and updating of the graphic interface.

To perform a synthetic analysis of how the signals input to the model are processed, it is necessary to start from the instant $t = 0$, in which the current coordinates of the vehicle coincide with the initial coordinates "egoInitialPose". Which are taken directly from the WorkSpace via the Scenario and Environment block. These coordinates are compared with the "scenario" file, also taken from the WorkSpace, to produce the states of all the actors in both global and local coordinates (calculated with respect to the position of the ego vehicle).

The output is then processed by the Highway Lane Change Planner block, whose purpose is to provide an output signal containing the trajectory that the vehicle must follow, calculated after having:

1. Compared the current state with that of the vehicles closest to it.
2. Eliminated all possible solutions belonging to trajectories concluding with collisions or not compatible with the vehicle dynamics.
3. Evaluated those that can be traveled based on the objective chosen by the planner, the evaluation is carried out by associating a specific cost parameter to each trajectory.

The presence of a vehicle model allows us to get even closer to a real system, since the planned trajectories are processed by a controller, which provides commands to the ego vehicle. Thanks to the presence of the vehicle model, it is possible to verify whether the reference trajectories produced by the planner are physically practicable by the ego vehicle, taking into account the related implementation delays and the physical characteristics that describe their behavior.

Subsequently an evaluation is performed by the Metrics Assessment block, within which checks are carried out on a wide range of parameters. If errors are found, the simulation stops; otherwise, the signal is provided as input to the Scenario and Environment block as the new current state of the ego vehicle. In this block, a comparison is made between the new position and the scenario, in order to obtain the Buses containing the new relative positions between the ego vehicle and all the other vehicles present in the scenario. After that, the mechanism repeats until the "StopTime" is reached or an error occurs.

3.1.2 Scenario and Environment

The Scenario and Environment subsystem is responsible for providing the model with signals relating to actors, lane boundaries, and roads.

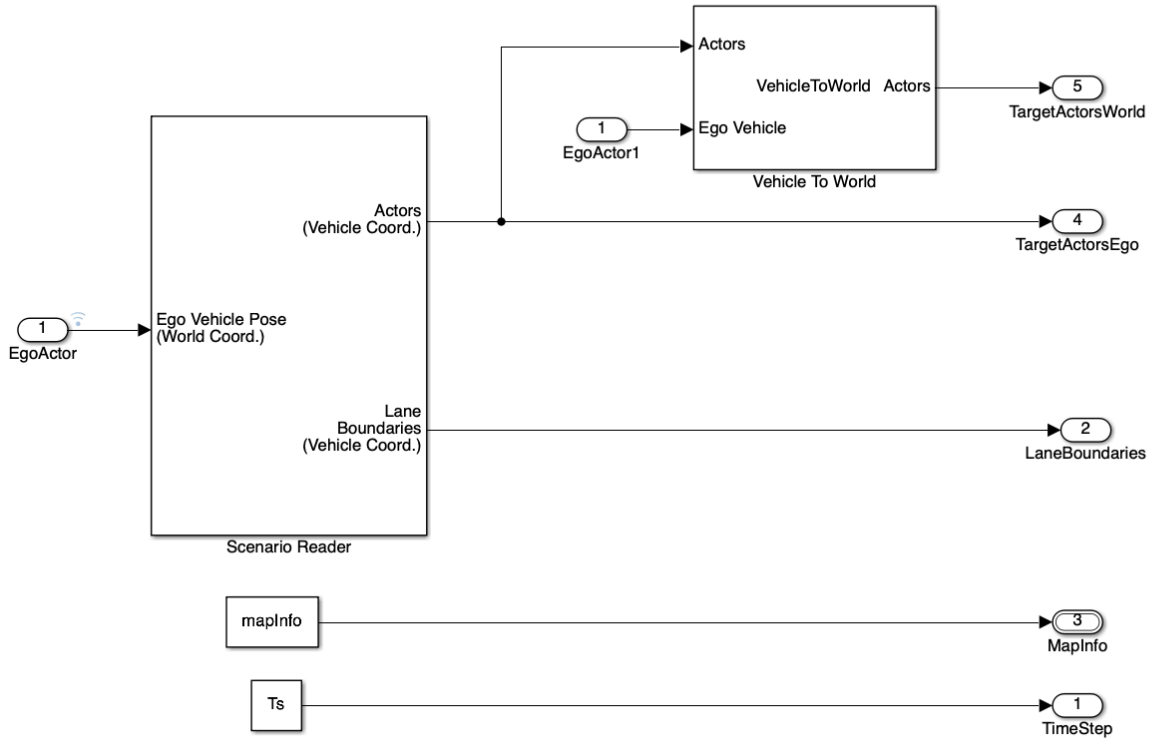

Figure 3.2: Scenario and Environment subsystem.

In Figure 3.2, two blocks that make up the subsystem can be identified. The first block is the Scenario Reader block while the second one is the Vehicle To World block and both are found in the Simulink mode under the Libraries tab or the Automated Driving Toolbox. We proceed below with a detailed analysis of their functioning.

Scenario Reader

This block is used for reading roads and actors from a drivingScenario file that can be generated directly as an object of drivingScenario or as an output of the Driving Scenario

Designer tool [15]. The latter is an application aimed at developing scenarios for simulations in the automotive industry, the functioning of which is discussed in paragraph 3.3. As can be seen in Figure 3.2, this block provides as output the coordinates of the actors and the position of the lane limits, with a local reference system (related to the ego vehicle). The input block consists of the "EgoActor" signal, containing the coordinates of the ego vehicle with a global reference system, the "egoInitialPose" signal, containing the starting state of the vehicle, and the scenario file present in the WorkSpace. The block configuration sheet has been shown in Figure 3.3 in order to allow a better understanding of its operation.

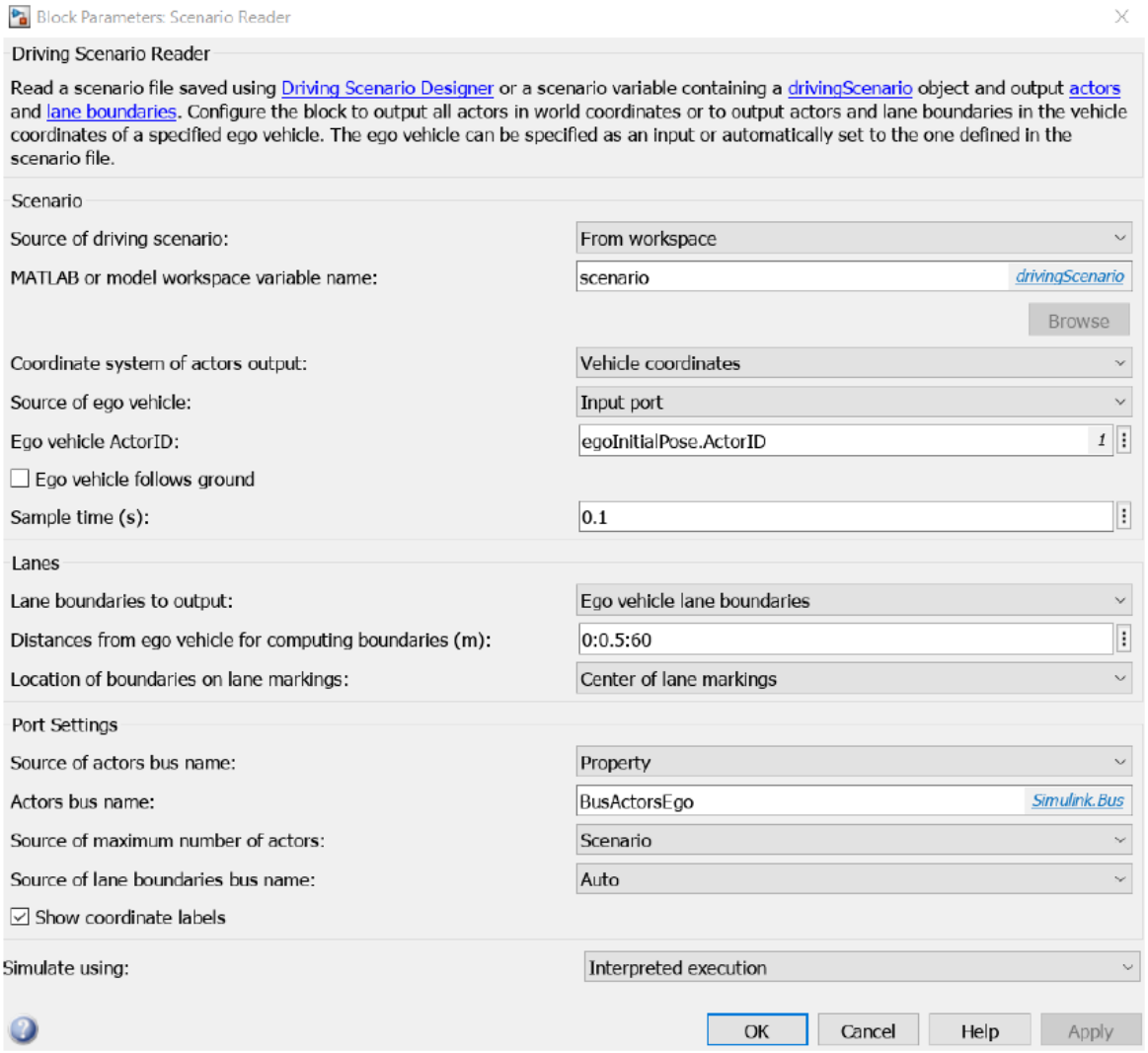

Figure 3.3: Scenario Reader block configuration tab.

Among the configuration options of the analyzed block, the possibility to choose the type of output coordinates (local or global) and the distance from the ego vehicle to be considered for the calculation of lane boundaries is particularly noteworthy.

Vehicle To World

This block is used to modify the type of coordinates of the actors. It receives as input, from the Vehicle Dynamics subsystem, the local coordinates of the actors and the global coordinates of the ego vehicle; by computing the two signals, it derives the global coordinates of all the actors.

In addition to the signals coming from the previously analyzed blocks, the Scenario and Environment subsystem provides "MapInfo" and "Ts" as outputs. These are two signals coming from two Constant blocks, through which the sampling time and road information are imported from the WorkSpace. "MapInfo" contains:

- The number of lanes of the road.
- The width of the lanes.
- The waypoints forming the central line of the road are referred to as "roadCenters".
- The number of waypoints.

3.1.3 Planner Configuration Parameters

In the Planner Configuration Parameters subsystem, some of the variables that are available in the WorkSpace are provided as input to the model via Constant blocks and the signals of which, in turn, form a Bus structure created through the Bus Creator block. The resulting Bus is then connected to the part of the model dedicated to the trajectory planner.

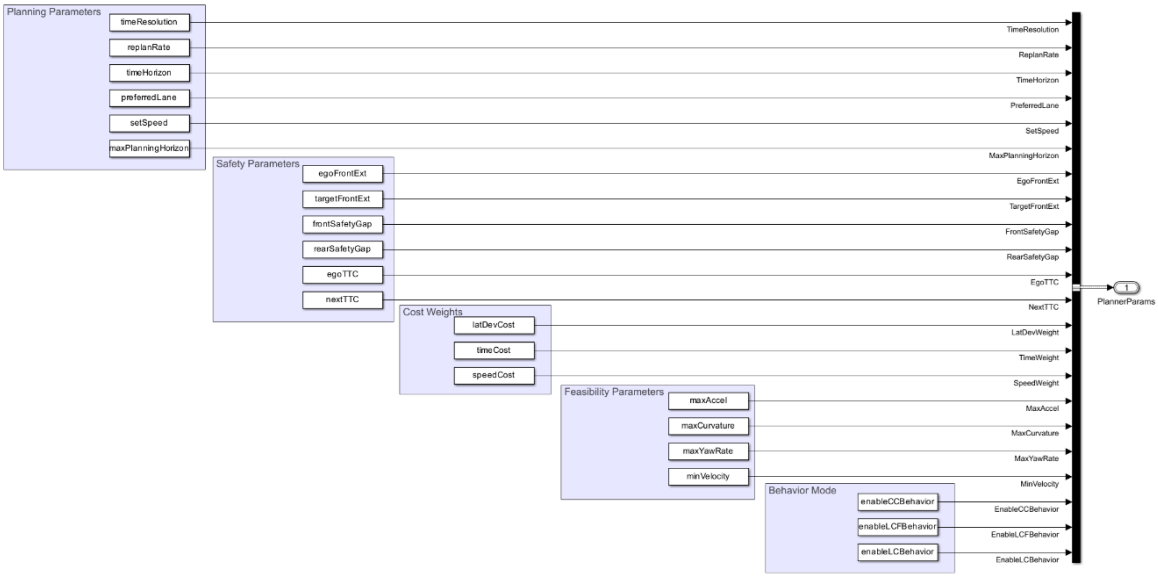

Figure 3.4: Planner Configuration Parameters subsystem.

As shown in Figure 3.4, the configuration parameters have been grouped into areas. Below is the list and description of the signals that make up the Bus.

- Planning Parameters: The first three parameters in this area define time variables in seconds. "timeResolution" is used for the sampling time, "replaneRate" defines the trajectory replanning time, and "timeHorizon" is a vector of time horizons for trajectory sampling. As planning parameters, the following were created: "preferredLane," which identifies the preferred lane for the ego vehicle, "setSpeed," which defines its speed, and finally "maxPlanningHorizon," which specifies the maximum longitudinal planning distance.
- Safety Parameters: This area defines parameters useful for maneuver safety checks. Specifically, "egoFrontExt" and "targetFrontExt" identify the frontal extension of the

ego and target vehicles, data used during collision verification. "frontSafetyGap" and "rearSafetyGap" were introduced to specify the safety distance between the ego vehicle and other vehicles in the scenario. Finally, "egoTTC" and "nextTTC" define the limits related to the collision time with other vehicles in the scenario.

- Cost Weights: In this area, the variables used to determine the weight of the contributions of the cost function, used to rank the terminal states sampled by the planner according to a preferential logic, have been grouped. These variables are: "latDevCost", "timeCost," and "speedCost," which identify the weight of lateral deviation, time horizon, and speed, respectively.
- Feasibility Parameters: In this area, the following parameters have been introduced: "maxAccel," "maxCurvature," "maxYawRate," and "minVelocity." These variables define limits that are useful for verifying the feasibility of a certain trajectory in kinematic terms. They relate to acceleration, curvature, yaw rate, and vehicle speed.
- Behavior Mode: This last area is dedicated to the model behavior. As mentioned in paragraph 3.1.1, it can operate in the following modes: cruise control (CC), lead car following (LCF), and lane change (LC). Through the variables "enableCCBehavior," "enableLCFBehavior," and "enableLCBehavior," some so-called flags, that allow disabling the use of a particular operating mode, are introduced.

Table 3.1 reports the default values of the listed parameters, which were chosen to perform the simulations. These parameters are modifiable; however, it was considered useful to provide an example of the chosen values, in order to offer a comparison parameter for possible model sensitivity studies.

Area	Parameter	Value	Unit
Planning Parameters	timeResolution	0.1	
	replaneRate	1	s
	timeHorizon	[1, 2, 3]	s
	preferredLane	/	
	setSpeed	/	m/s
	maxPlanningHorizon	80	m
Safety Parameters	egoFrontExt	5	m
	targetFrontExt	5	m
	frontSafetyGap	30	m
	rearSafetyGap	10	m
	egoTTC	4	s
	nextTTC	4	s
Cost Weights	latDevCost	1	
	timeCost	-1	
	speedCost	1	
Feasibility Parameters	maxAccel	5	m/s^2
	maxCurvature	1	1/m
	maxYawRate	80	°/s
	minVelocity	0	m/s
Behavior Mode	enableCCBehavior	1	
	enableLCFBehavior	1	
	enableLCBehavior	1	

Table 3.1: Planning Configuration Parameters values.

It is specified that, unlike all the other listed parameters, which are defined during the calibration phase of the planner, the "setSpeed" parameter and the initial value of the "preferredLane" parameter are defined within the scenario. For a more detailed definition of the listed quantities, please refer to the analysis of the individual functions in the model, where these parameters are processed, as reported below.

3.1.4 Highway Lane Change Planner

The Highway Lane Change Planner subsystem is the most important part of the model of the same name, since it is the one dedicated to the trajectory planner. From the general block diagram in Figure 3.1, it emerges that the set of input signals of the subsystem under examination is formed by the outputs produced by the Scenario and Environment and Planner Configuration Parameters subsystems, respectively analyzed in paragraphs 3.1.2 and 3.1.3, and by the "egoActor" Bus, output of the Vehicle Dynamics subsystem, containing the current state of the ego vehicle.

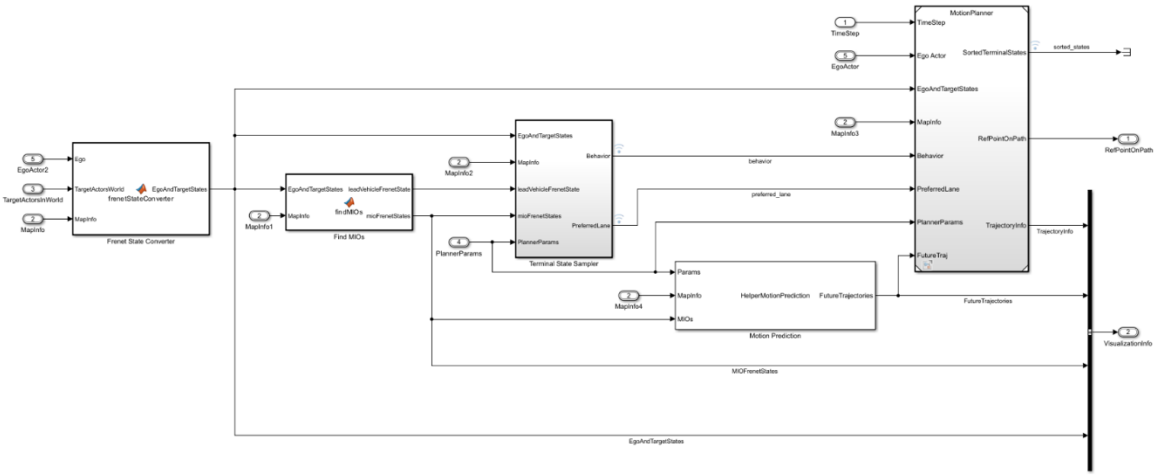

Figure 3.5: Highway Lane Change Planner Subsystem.

As can be seen in Figure 3.5, the first block to process the input signals to the subsystem is the Frénet State Converter. This function block contains a MATLAB code that communicates with the Simulink model only through its input and output signals. It converts the coordinates of the input from the starting global reference system to the Frenet reference system.

The employed solution is common in path planning studies and involves the adoption of a mobile reference system based on a curvilinear coordinate and its normal, analyzed in detail in the second chapter.

Frenet State Converter:

The frenetStateConverter algorithm starts with processing the input data, which includes: the Ego State in global coordinates, representing the current state of the lead vehicle; the states of surrounding or relevant vehicles, also expressed in global coordinates; essential information regarding the current road configuration and conditions.

As mentioned previously, this function is designed to perform a transformation of coordinates from the global system to Frenet coordinates. However, it is not solely focused on a simple conversion; matter of fact it also creates an output structure called "EgoAndTargetStates" Bus, which conveys all the scenario information into a single output, within the new reference system.

Additionally, it is important to highlight that the model assumes that the Frenet coordinates reference curve is aligned with the centerline of the road. This assumption is made every time we perform the coordinate transformation. Consequently, the vehicle position is described by two key values: s, which is the distance along the path, and d, the distance from the centerline.

$$[x, y, \psi, \kappa, \upsilon, \alpha] \rightarrow [s, \dot{s}, \ddot{s}, d, d', d'']$$

It is also worth mentioning that this function calls another one, HelperLCPlannerDefaultData, to avoid cluttering the main Simulink functions. The project development involved using various helper functions like this one, which helped simplifying tasks that used across the model multiple times.

Find MIOs:

The bus signal created through the previous block serves as the input of a further MATLAB function block.

The Find MIOs block aims to identify the vehicles that are deemed most crucial for developing the maneuver in the given scenario.

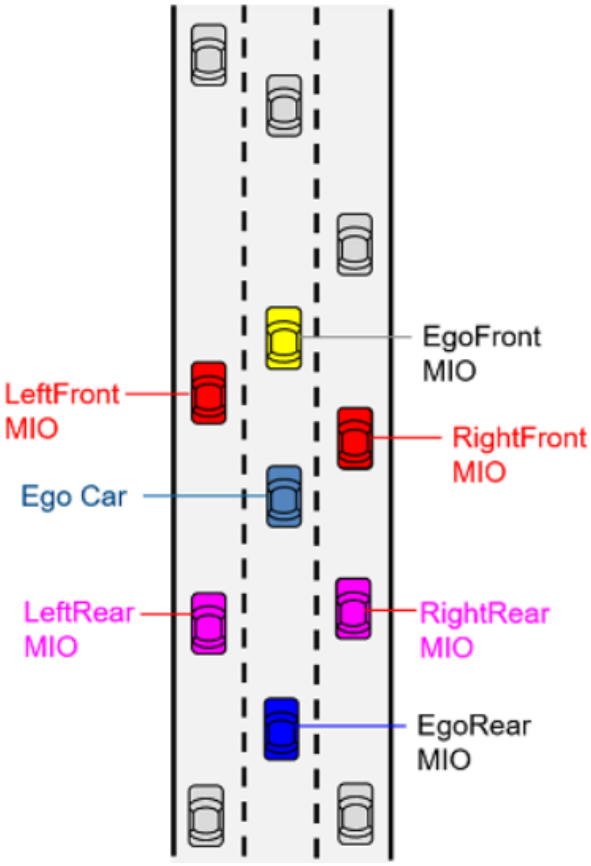

Figure 3.6: Most Important Objects [14].

As can be seen from Figure 3.6, multiple vehicles can be defined as “important”; what distinguishes them is their immediate proximity to the ego vehicle in a given direction.

The function called findMIOs inside this block handles and receives as input the following data: the ego vehicle state (Frenet), the target vehicle states (Frenet), the target IDs, and the road data.

For each individual vehicle, the lane number is determined using the helperDetectLaneNumber function which calculates the lane based on the road information and the lateral distance of the vehicle from the reference line.

A vector called “validLanes” is then created that initially includes all possible lanes. This vector is then modified and invalid or non-existent lanes are removed from the vector.

For each remaining lane in “validLanes”, the following steps are performed. First, the vehicle

preceding the ego vehicle with the smallest positive distance in each valid lane is identified. A check is then performed to see if the currently scanned lane is the one in which the ego vehicle is located. If so, the vehicle identified as preceding becomes relevant data. Additionally, the vehicle following the ego vehicle with the smallest negative distance in each valid lane is identified.

After iterating through all valid lanes, the algorithm produces an output that includes the IDs of the MIOs identified as preceding and following vehicles, as well as the IDs of the leading vehicles and their current states.

Terminal State Sampler Subsystem

The Terminal State Sampler block is a part of the Highway Lane Change Planner subsystem and is itself a subsystem defined by several MATLAB function blocks. Its function is to process the signals coming from the two blocks described above, namely Frenet State Converter and Find MIOs, also using the signals “mapInfo” and “PlannerParameters”. The goal is to sample different terminal states based on the active operating mode (CC, LC, or LCF). The terminal state represents the set of parameters describing the state of the ego vehicle at the end point of the trajectory segment provided as output by the planner. Initially, several terminal states are identified, each of which will be evaluated and classified based on an associated cost parameter. Each terminal state is tied to a certain time horizon, which represents the interval between the current instant and the moment in which the ego vehicle will reach the endpoint of the planned trajectory. This time horizon is also associated with an operating mode of the planner (CC, LCF or LC). The combination of all these elements leads to the creation of a complete set of terminal states. Among these, the one corresponding to the trajectory planned and provided as output by the Highway Lane Change Planner subsystem will be selected. Figure 3.7 shows a block diagram of the subsystem.

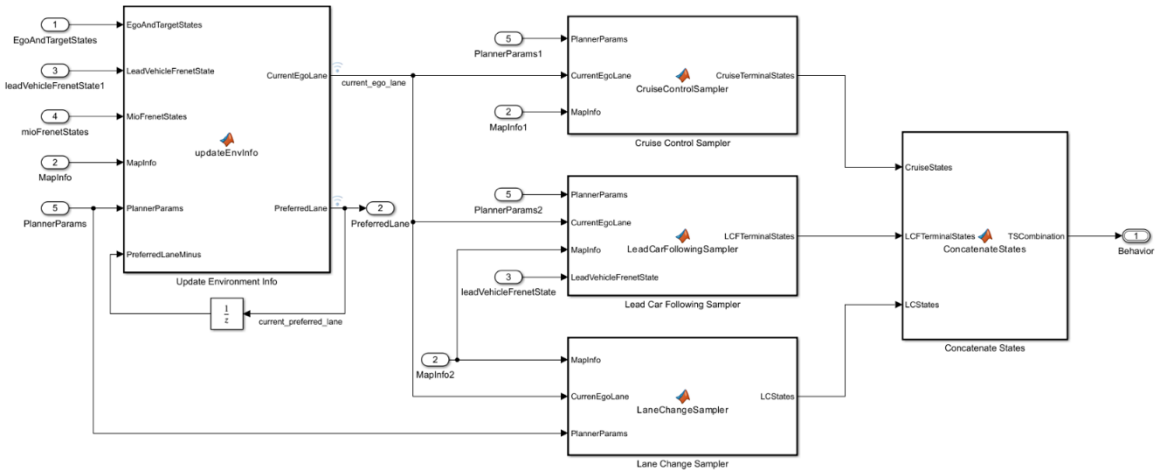

Figure 3.7: Terminal State Sampler Subsystem.

Looking at the processing flow of the input signals, the Terminal State Sampler subsystem is composed of the following main blocks:

- Update Environment Info: Block dedicated to identifying the current lane of the ego vehicle and the preferential lane.
- Cruise Control Sampler, Lead Car Following Sampler, and Lane Change Sampler: Three blocks responsible for sampling the terminal states, each specialized in an operating mode of the planner.
- Concatenate States: Block that concatenates the calculated terminal states.

After this general overview of the components of the Terminal State Sampler subsystem, it is appropriate to move on to a more detailed analysis.

The function contained in the updateEnvInfo block was created to process the input signals and correctly provide the data to the function that follows, called updateEnvironmentInfo. This function, which is upstream of the process of determining the preferential lane index, is responsible for bringing the preferred line index back to the value defined in the Planner configuration parameters block, i.e. the value corresponding to the trajectory of the ego vehicle described in the scenario. If the subsequent verification of this index is positive, it is maintained, thus always giving priority to the lane corresponding to the trajectory planned

in the scenario.

The second function, updateEnvironmentInfo, has the task of determining the current lane of the ego vehicle, by calling the helperDetectLaneNumber function, and of providing the necessary signals to the helperFindPreferredLane function, containing the main algorithm aimed at determining the preferential lane. The helperFindPreferredLane function initially collects crucial information such as the current lane, relative distance between vehicles, TTC and classifies MIO vehicles as safe or unsafe. Using this information, an initial assessment of the ego vehicle current lane is made and if there are no unsafe vehicles in this lane, the lane will be maintained as preferred. However, if unsafe vehicles are detected, the system checks the lanes to the left and then to the right of the ego vehicle for unsafe vehicles. If one of these lanes is found to be free of unsafe vehicles, it is designated as the new preferred lane. If there are no alternative lanes available that are safe, then the ego vehicle current lane remains unchanged. This approach reduces the risk of collisions and improves overall route safety by dynamically adapting the vehicle position and responding to immediately relevant traffic and safety conditions.

To better understand the logic behind how preferred lane selection works, it is necessary to delve deeper into the Safe/Unsafe concept and the parameters it depends on. A lane is declared Unsafe when one of the MIO vehicles travelling in it is considered unsafe. This happens if at least one of the TTC and Relative Distance values, calculated between the MIO vehicle and the ego vehicle, exceeds the pre-established limits. The Relative Distance corresponds to the absolute distance between the ego vehicle and the MIO; the TTC (Time to Collision), instead, indicates the time necessary for a collision between the ego Vehicle and the MIO, assuming that there are no changes in speed or direction of both vehicles. It is calculated as the ratio between the Relative Distance and the Relative Speed of approach of the two vehicles, i.e. the projection of the relative speed along the connecting line, i.e. the straight line segment that passes through the center of the rear axle of both vehicles.

These parameters are fundamental to understanding the operation of the planner, and therefore in paragraph 3.4 a more in-depth analysis of the logic that governs the Safe/Unsafe evaluation is carried out.

To determine whether a vehicle is safe or not, the two signals are processed by the checkSafety function, which compares the Relative distance with the SafetyGap values defined in the Planner Configuration Parameters block. The function also performs a TTC check against the defined parameters. If one of the two checks related to a MIO is negative, the lane in which that vehicle travels is declared unsafe. If at a given instant in time the preferential lane becomes unsafe (Unsafe), a procedure is activated to modify the preferredLine index; otherwise, the current index is maintained. At the end of the preferential lane determination, the latter is provided as input to the function itself in the next time step, under the name of PreferredLaneMinus, as shown in Figure 3.7.

The three main blocks dedicated to the ego simulation mode, the Cruise Control Sampler, the Lead Car Following Sampler, and the Lane Change Sampler, play a key role in sampling the terminal states for their respective operating modes.

Each block builds a matrix where each row represents the Frenet coordinates of the terminal state for that mode, with the last element of the row indicating the associated time horizon. During Cruise Control (CC) mode, the ego vehicle is set to maintain its current lane and reach a consistent speed as determined by the 'setSpeed' parameter. This procedure starts with measuring the distance from the reference curve to the center of the ego current lane, which is referred to as "lateralOffset". Using this information, a terminal state vector is constructed in Frenet coordinates, with the velocities initially undefined (NaN), but with the speed set to "setSpeed". This vector, replicated for different values of the "TimeHorizon" vector, is stored in an array, called "CruiseTerminalStates", which contains all possible future time configurations of the vehicle in CC mode.

Similarly, in the Lead Car Following (LCF) mode, the ego vehicle stays in the current lane but adapts its speed to that of the vehicle in front, called the "lead vehicle". Again, the process starts with the acquisition of the lateral distance of the ego vehicle with respect to the reference curve. However, unlike the CC mode, the final speed $\dot{s}$ is not determined by the "setSpeed" parameter but is set based on the speed of the lead vehicle. As in the CC mode, the terminal state vector is replicated for different values of "TimeHorizon”, but this time stored in an array called "LCFTerminalStates".

Lane Change (LC) mode introduces additional complexity, as the ego vehicle not only maintains the cruising speed set by "setSpeed", but also moves laterally into one of the adjacent lanes. As in the other modes, the process starts with the acquisition of the lateral distance to the considered lanes, defined as "adjacentLanes". The key difference in LC mode is the greater variety of terminal states, as the number of states depends on the number of adjacent lanes and different time configurations, represented by the "TimeHorizon" vector. The terminal state matrix, built with predefined speeds and variable lateral distances, is replicated and stored in the "LCTerminalStates" matrix, which reflects future scenarios with lane changes.

Although the three blocks follow a similar basic logic of data acquisition, construction and replication of the terminal state vector, each one is distinguished by its specific operating mode. CC mode exclusively focuses on maintaining the speed while LCF mode introduces adaptation to the speed of the lead vehicle and LC mode combines speed management with the possibility of changing lanes.

At the end of the process for all modes, the terminal state matrices generated by the three blocks are concatenated into a single structure through the Concatenation States block. This block unifies the outputs, creating a bus containing the entire combination of terminal states, and also generating a vector that identifies the mode of operation for each state (CC, LCF or LC).

Motion predition

The Motion Prediction block is designed to calculate the future states of vehicles identified as MIO, based on the assumption that the velocity components remain constant and starting from the current states of the vehicles. Among the relevant inputs for this block, we find the planner configuration parameters defined in the Planning Configurator Parameters block, the information relating to the road and the states in Frenet coordinates of the MIO vehicles. These data are processed by the stepImpl function, which is an integral part of the HelperMotionPrediction.

In detail, the process begins with the storage of essential variables such as the "Time Horizon" vector, the "Time Resolution" parameter, and the matrix containing the road waypoints. Using the "Time Horizon" and the "Time Resolution", the number of future states to predict is calculated, creating a time vector that extends from the current time up to the maximum of the considered time horizon.

Subsequently, a matrix containing the current Frenet states for each MIO vehicle is created, replicated for each future time instant defined by the time vector. This initial replication preserves the current state of each MIO vehicle throughout the entire forecast horizon.

The next step involves overwriting this matrix by incrementing the longitudinal and lateral coordinates, s and d, based on assumed constant increments for the velocity, computed as Δs and Δd. This iterative procedure progresses until the forecast time horizon is complete.

Finally, the data obtained, still in Frenet coordinates, are converted to global coordinates through the frenet2global function. This transformation is crucial because, although the Frenet coordinates facilitate the generation of trajectories thanks to their simplicity of manipulation, the final position evaluations and any navigation interventions must be performed in global coordinates to ensure correct interaction with the surrounding environment and with other autonomous driving system modules.

Motion Planner Subsystem

The Motion Planner subsystem is the central part of the planner, as it is dedicated to generating the ego vehicle trajectory.

To generate a trajectory, it is necessary to start from the sampled terminal states and the planner configuration parameters.

Planning occurs with a certain frequency and, having to follow a single trajectory, it is necessary to implement cost functions with the aim of discarding the worst options.

After generating the alternative trajectories, it is necessary to verify their feasibility and the presence of any collisions; after this the trajectory to follow will be obtained.

In Figure 3.8 it is possible to appreciate the block diagram of the subsystem.

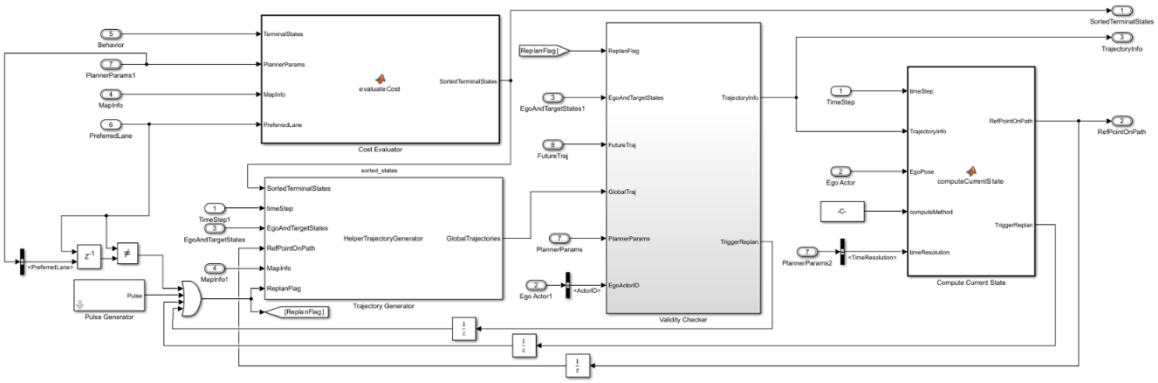

Figure 3.8: Motion Planner Subsystem.

Furthermore, in this case the subsystem is characterized by the presence of different blocks, each of which has a specific function in relation to the operations performed by the subsystem that have been described previously. Analyzing the system inputs, appreciable in Figure 3.5, we note that the first block to process the signals is the one intended for the evaluation of the cost functions. This is a MatLab function block that receives as input the concatenated terminal states relating to the three driving modes, the information on the road, the index of the lane considered as preferred and the configuration parameters of the planner. The function inside it is the helperEvaluateCost and is responsible for reordering the terminal states following a preferential logic.

Each state present in the set is associated with a cost parameter. The function adopted to calculate these quantities is the following:

$$\text{costTS} = \text{latCost} + \text{timeCost} + \text{speedCost}$$

Each addend is calculated individually using the following formulas:

$$\text{latCost} = \text{latDeviation} \cdot \text{latDevWeight}$$

$$\text{timeCost} = \text{timeHorizon} \cdot \text{timeWeight}$$

$$\text{speedCost} = |\dot{s} - \text{setSpeed}| \cdot \text{speedWeight}$$

- latCost: Quantifies the distance from the lane indicated as preferential of the lane that the ego vehicle would occupy at the terminal point of the trajectory segment to be planned.
- timeCost: This parameter is greater the shorter the time horizon of the given terminal state. This is possible because the value of "timeWeight" is negative. In this way it is preferable to plan longer trajectory segments rather than many short segments.
- speedCost: Quantifies the difference in absolute value between the speed of the ego vehicle at the terminal point of the trajectory segment and the cruising speed "setSpeed" of the ego vehicle, defined in the Planner Configuration Parameters block.

As can be seen from the previous equations, each contribution of the cost function has in its formulation a costWeight parameter, defined by a specific MatLab script, allowing to vary the relative weight of the addends of the costTS parameter.

Replan flag

To activate the trajectory generator, a Boolean variable has been inserted, as well as a binary variable capable of assuming only two Boolean values, namely true or false, called "ReplanFlag". When this assumes the logical value "true" (1) it expresses the need to analyze the set of terminal states to plan a new trajectory segment.

The "ReplanFlag" signal is the output of an OR block with four input signals; since it is a block that has the function of a logical operator, the input signals are also Boolean variables.

As can be seen in Figure 3.8, the inputs of the aforementioned block are:

- Output signal of the NotEqual block: Compares the "Preferred Lane" parameter coming from the output signal to the Terminal State Sampler subsystem with the same signal processed by a Delay block. Consequently, as soon as there is a variation in the "Preferred Lane" value, a new trajectory planning is requested.
- Pulse Generator block output signal: Generates a positive boolean variable with a fixed frequency. The replane rate is set to $1s^{-1}$. This quantity is governed by a

parameter defined in the Planner Parameters block.

- Validity Checker subsystem output signal.
- Compute Current State subsystem output signal.

The "ReplanFlag" signal and the "SortedTerminalStates" structure are part of the Trajectory Generator system input group.

Trajectory Generator Subsystem

The Trajectory Generator subsystem is dedicated to the creation of reference trajectory segments for the ego vehicle. They are generated starting from the set of terminal states ordered according to the cost function, using the trajectoryGeneratorFrenet function provided by the MatLab Automated Driving Toolbox and explored in depth in the second chapter.

The function that executes the central part of the algorithm has been called stepImpl and is part of the HelperTrajectoryGenerator function. It is activated when the signal called "replanFlag" assumes the logical value "true". Through feedback of the signal called "RefPointOnPath", appropriately processed by a Delay block, the current reference state of the ego vehicle is obtained. It is used as the first point of the new trajectory segment. Therefore, the new trajectory segments to follow are planned so that the reference trajectory is a single continuous curve.

The "connect method" is used to generate the n trajectories that connect the reference point to the n terminal states. Starting from the waypoints of the reference trajectory, which in this case has also been associated with the center line of the roadway, an object called "connector" is obtained. This is then processed by the connect function that extrapolates the information about the reference curve, to generate the n trajectories in global coordinates.

Once the creation of the trajectories associated with the terminal states has been completed, the flow of information proceeds through the Validity Checker subsystem, dedicated to the exclusion of non-passable trajectories.

Validity Checker Subsystem

The Validity Checker subsystem is dedicated to verifying the feasibility of the generated trajectories; this is carried out on two fronts, one purely kinematic and the other linked to possible collisions with the target vehicles.

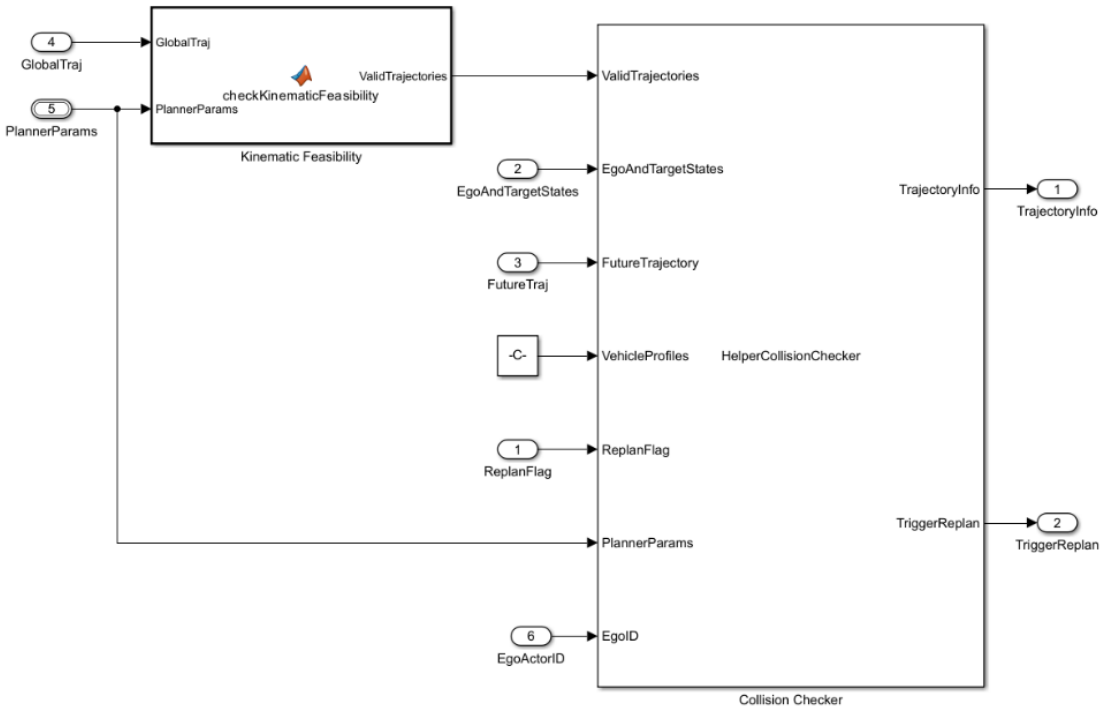

Figure 3.9: Validity Checker subsystem.

As can be seen in Figure 3.9, the subsystem is composed of two different blocks, each of which is dedicated to a specific verification. The kinematic Feasibility block is of the MatLab function type and hosts the helperKinematicFeasibility function. Its task is to establish whether a trajectory is valid or not, based on its kinematic feasibility.

Inside it, the peak values of the main quantities associated with the alternative trajectories are analyzed, such as longitudinal and lateral acceleration, yaw velocity and curvature, as well as a minimum limit for the speed. If all the quantities mentioned respect the limit parameters defined in the Planner parameters configuration block, the trajectory is declared

"valid". It is specified that the condition relating to lateral acceleration was added later, as it was considered of primary importance during the evaluation of a trajectory.

The verification of possible collisions is carried out within the Collision Checker block; it contains an algorithm capable of comparing the trajectories generated with those of the vehicles defined as MIO.

The algorithm allows to identify the obstacles present in the scenario (MIO vehicles) and takes into account the relative geometries in order to be able to evaluate the risk of collision with the trajectory of the ego vehicle. This goal is achieved thanks to the help of the Dynamic capsule-based obstacle list, as well as the modeling of the vehicles in two-dimensional capsules, whose operating logics have been described in detail in the second chapter, paragraph 2.

Each of the alternative trajectories is analyzed by comparing it with the predictions of the trajectories of the MIO vehicles, which, it should be remembered, are identified by the Motion Prediction block. Using the list of capsules, it is then possible to compare the extension of the bodies instant by instant in search of overlaps, which correspond to possible collisions between vehicles. If a trajectory were to lead to a collision, the StepImpl function contained within the HelperCollisionChecker function associates a Boolean variable called "isColliding" with the logical value "true".

Compute Current State

The last block to be described to complete the analysis of the Motion Planner subsystem is called Compute Current State.

This is a MatLab function block dedicated to the estimation of the next state of the vehicle; that is, starting from the reference trajectory, output of the Validity Checker block, it identifies the coordinates of the reference point for the next time step.

The function that performs the described task differs based on the model in which the trajectory planner is inserted; this happens because different outputs are needed.

We can distinguish two different cases:

- Case A: The model with only the trajectory planner is used. In this case it is assumed that the reference trajectory, output of the planner, is exactly the trajectory of the ego vehicle and, to identify the point of the next time step and the descriptive quantities of the vehicle motion at that point, it is necessary to base the calculation on time. Three variables are of fundamental importance:
  - timeResolution: Parameter containing the time sampling unit, i.e. the time that elapses between the current state and the next time state.
  - replanTime: Time instant in which the new trajectory segment is planned.
  - timeDelta: Time that elapses between the step in which the new trajectory segment is planned and the next time step. It is calculated with the following formula.

$$\text{timeDelta} = \text{time} - \text{replanTime} + \text{timeResolution}.$$

- Case B: The model that integrates the planner with a controller and a vehicle model is used. The reference trajectory, output of the planner, is used to calculate the errors from a comparison with the actual position of the ego vehicle. Therefore, unlike the previous case, the reference point is calculated as an interpolation performed geometrically and not through the time vector.

As can be seen in Figure 3.8, the inputs of the block are:

- Trajectory Info: Output structure of the Validity Checker subsystem containing the reference trajectory.
- timeStep: Parameter containing the time sampling unit.
- EgoPose: Structure containing the current state of the ego vehicle in global coordinates.
- computeMethod: Flag indicating which of the two functions mentioned must be used.

Distance between points of a trajectory

The distance between the points of the reference trajectory depends on several factors, with the points defined on a temporal basis. The ego vehicle takes an amount of time equal to the temporal sampling unit TimeStep to travel the distance between one point and the next. The number of points of a given trajectory segment is calculated as:

$$n_{pt} = \frac{\text{TimeHorizon}}{\text{TimeStep}} + 1$$

Consequently, as shown in Figure 3.10, by decreasing the speed, the distances between the points are reduced, since the distance traveled is smaller but the number of points remains constant; by decreasing the TimeStep, the distance between the points is reduced since the number of points increases and by decreasing the Time Horizon, the distance between the points increases when there is a non-zero lateral velocity Δd, since the vehicle must travel the same lateral distance in less time.

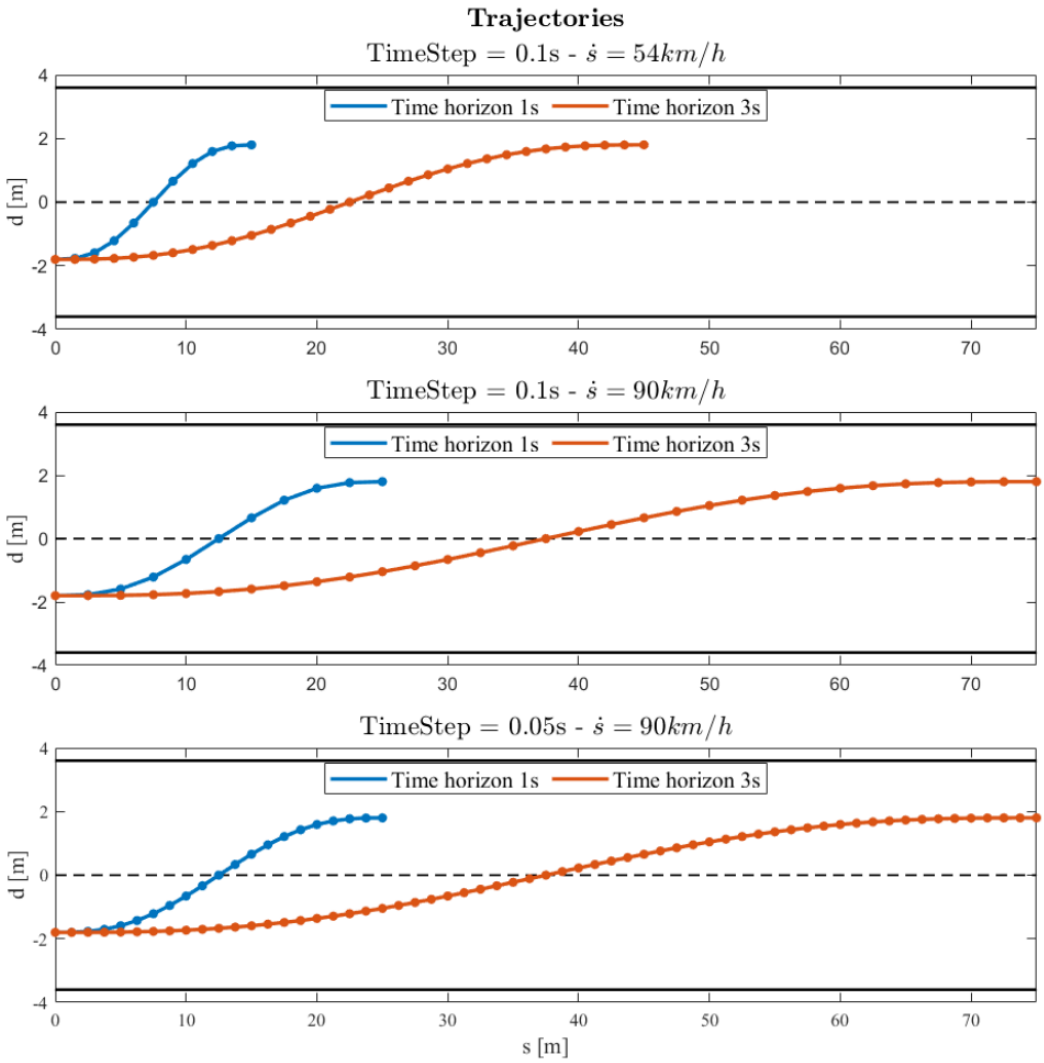

Figure 3.10: Distances between points of the reference trajectory.

3.1.5 Metrics Assessment

The Metrics Assessment subsystem does not directly provide an active contribution to the assistance system, as it does not alter the signals present in the model. At each time step it carries out a check of the scenario and the quantities that characterize the vehicle motion; if the imposed limit conditions are not respected, alarm signals are produced, and the simulation is interrupted in the event of a collision between vehicles.

The general structure of the subsystem is represented in Figure 3.11.

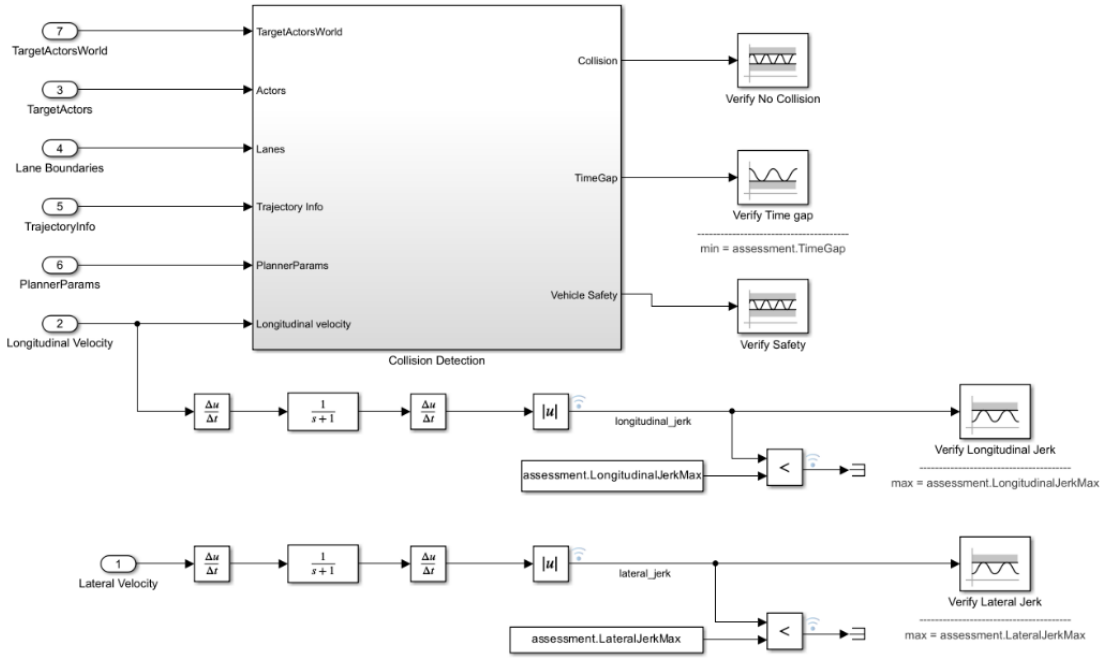

Figure 3.11: Metrics Assessment subsystem.

The verification is therefore carried out on quantities expressed in the Frenet reference system.

Starting from the two accelerations, the quantities J_{long} and J_{lat} are obtained, which indicate the longitudinal jerk and the lateral jerk respectively.

Collision Detection

The Collision Detection subsystem verifies the absence of collisions, the actual distance between the ego vehicle and the lead vehicle, and establishes whether the ego vehicle is in safe conditions. Figure 3.12 shows the general structure.

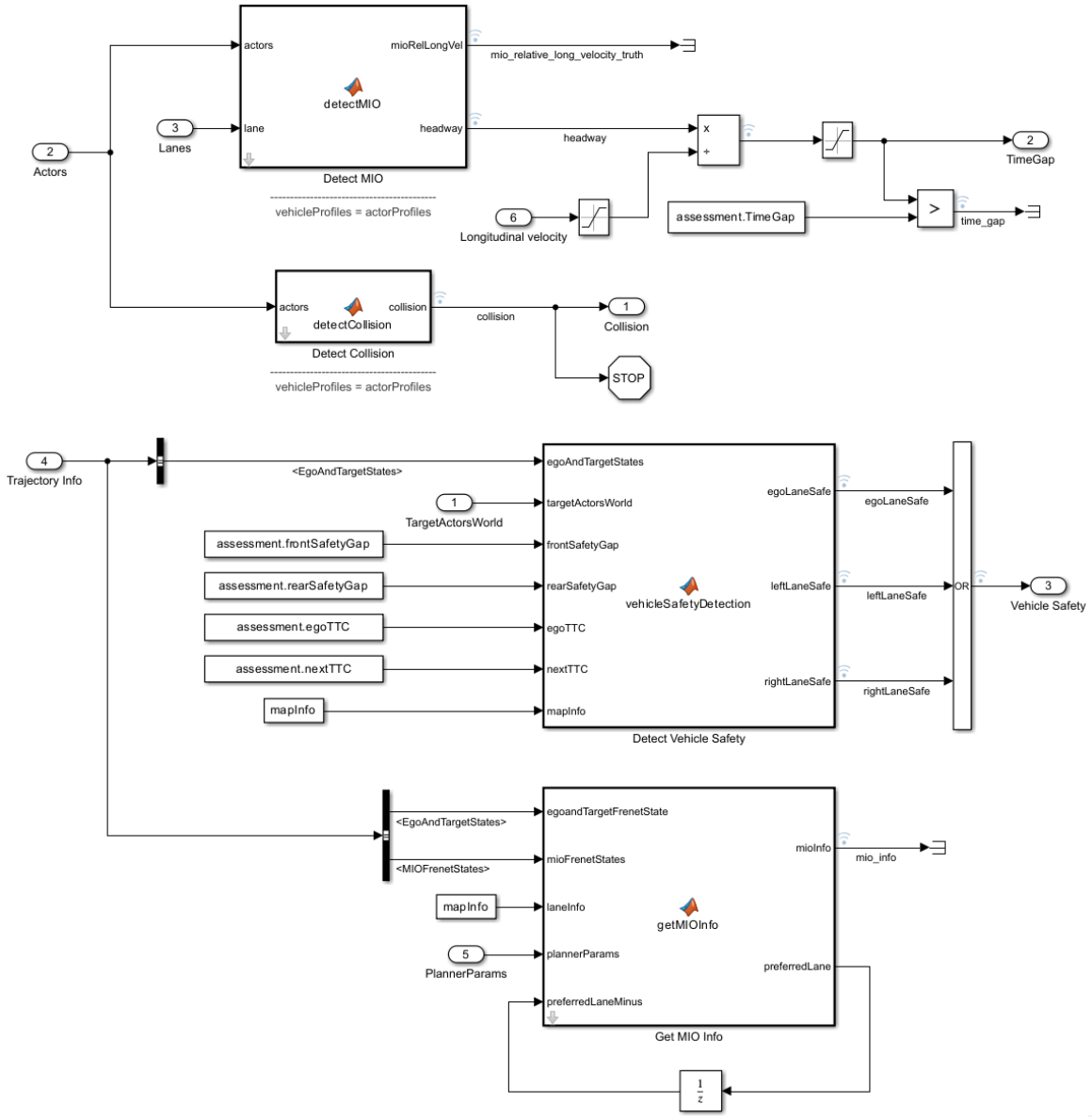

Figure 3.12: Collision Detection subsystem.

It is composed of several MatLab function blocks, each of which is briefly described:

- Detect Collision: Receives as input the position and geometric characteristics of the actors present in the scenario. The presence of overlaps is checked by comparing the areas occupied by the individual vehicles. In the case of overlapping areas, a collision occurs, and the simulation is interrupted with a Stop block.
- Detect Lane Vehicle: Receives as input the position and geometric characteristics of the actors in the scenario. Identifies the lead vehicle and calculates the headway parameter, which indicates the distance between the front of the ego vehicle and the rear of the lead vehicle. Subsequently, the TimeGap parameter is calculated with the formula:

  $$\text{TimeGap} = \frac{\text{Headway}}{\text{Longitudinal velocity}}$$

  It indicates the time needed for the ego vehicle to reach the point currently occupied by the lead vehicle.
- Detect Vehicle Safety: Establishes whether the lane occupied by the ego vehicle or the adjacent lanes are defined as Safe/Unsafe. The methodology is the same as used in the Terminal State Sampler Subsystem (paragraph 3.4). The "Vehicle Safety" parameter assumes the logical value "true" (1) when one of the three lanes is Safe using a logical OR operator.
- Get MIO Info: Dedicated to the creation of the "mioInfo" structure in output. This signal is not processed in the model but is saved instant by instant in the WorkSpace for post-processing examination. The output structure contains the information on each MIO vehicle: distance, relative speed, TTC calculated with respect to the ego vehicle, and the "IsSafe" parameter, which assumes the logical value "false" when one of the quantities does not respect the imposed limits. The structure is obtained using the helperFindPreferredLane function, already examined previously.

Alarm signals

For the quantities obtained from the two analyzed blocks, checks are performed that generate alarm signals if they are not respected; these signals are visible in real time in the WorkSpace. The checks performed are the following:

- Verify No Collision: If the signal is different from 0, and therefore a collision is detected, the simulation is interrupted.
- Verify Time Gap: If the signal is lower than the minimum threshold set, a string is generated in the WorkSpace that communicates to the user the instant in which the signal exceeds the threshold.
- Verify Safety: If the signal is different from 1, a string is generated in the WorkSpace that communicates to the user the instant in which the lane occupied by the ego vehicle and the two adjacent lanes are simultaneously unsafe.
- Verify Longitudinal Jerk and Verify Lateral Jerk: If the signals go outside the limits between the maximum threshold and the minimum threshold, a string is generated in the WorkSpace that communicates to the user the moment in which one of the two thresholds is exceeded.

It should be noted that the limits on Time Gap and Jerk are defined manually using a dedicated script. The default parameters are shown in Table 3.2.

Parameter	Value	Unit
Time Gap	0.8	s
J_{long} min	-5	m/s^2
J_{long} max	5	m/s^2
J_{lat} min	-5	m/s^2
J_{lat} max	5	m/s^2

Table 3.2: Planning Configuration Parameters values.

3.1.6 Visualization

The Visualization block is dedicated to the creation of a graphic feedback of the simulation, provided and updated in real time. It receives as input the main Bus signals of the model, containing data related to the ego vehicle, to the other vehicles present in the scenario, to the planner outputs, including all the non-optimal trajectories, and the geometric characteristics of the actors. It hosts a MatLab code dedicated to the post processing of the signals received as input. Since it does not make changes to the aforementioned signals, it is not an active component of the assistance system, but a useful tool for its development; since it allows to visually analyze the evolution of the vehicle states in the scenario. For this reason, it was considered appropriate to focus on the output of the block. It consists of a live plot, called Lane Change Status Plot, whose interface is shown in Figure 3.13.

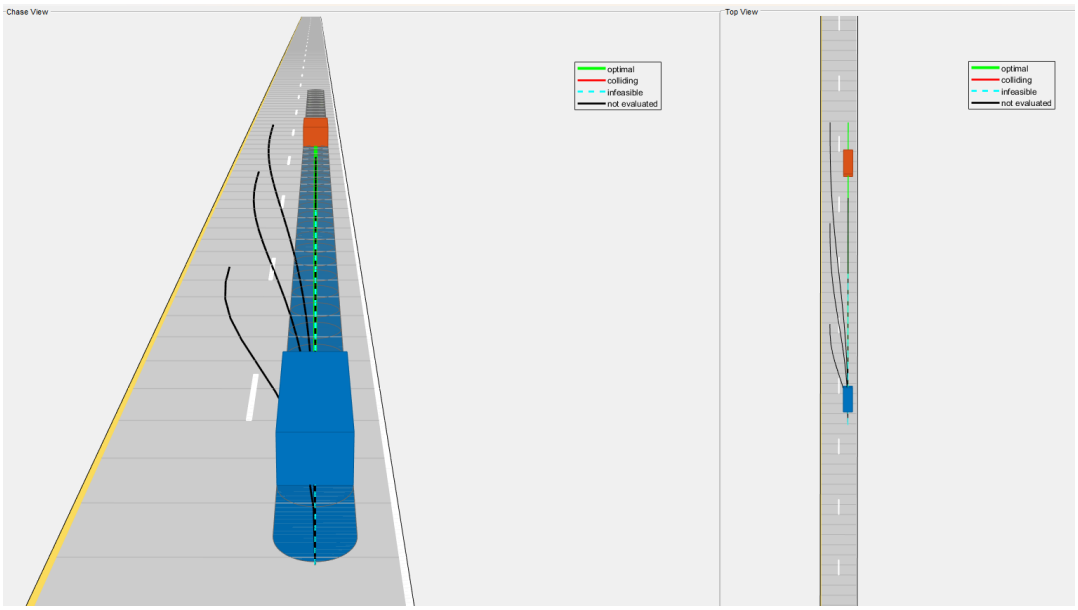

Figure 3.13: Visualization subsystem - Lane Change Status Plot.

There are two different views:

- "chase view": chase view, which offers a close-up view of the ego vehicle, with an angle that allows to view a large part of the scenario.
- "top view": view from above, useful for understanding the actual distances between vehicles.

The information translated into graphic form is the following:

- All planned trajectories: Each step graphically shows all the trajectories available in the set of alternatives, with different colors to indicate their status. The red segments indicate trajectories that lead to a collision, the blue ones trajectories that do not respect the kinematic feasibility limits, and the black ones trajectories not evaluated (i.e. those with a cost parameter higher than the one chosen as a reference).
- The reference trajectory: It is represented with a green segment.
- The vehicles in the scenario: Represented with parallelograms of different colors.
- 2D Capsules: The capsules used for collision verification are represented on the road surface, with the future states of the vehicles associated with the current trajectory.

Another tool for post processing is the Lane Change Simulation Analysis. This is a tool that analyzes the "logsout" structure, which contains the trend of all the model signals over time. They are saved using the inspector within specific fields of the "logsout" structure and can be added or removed manually. The Lane Change Simulation Analysis also has a graphical interface that includes the chase view, as shown in Figure 3.14.

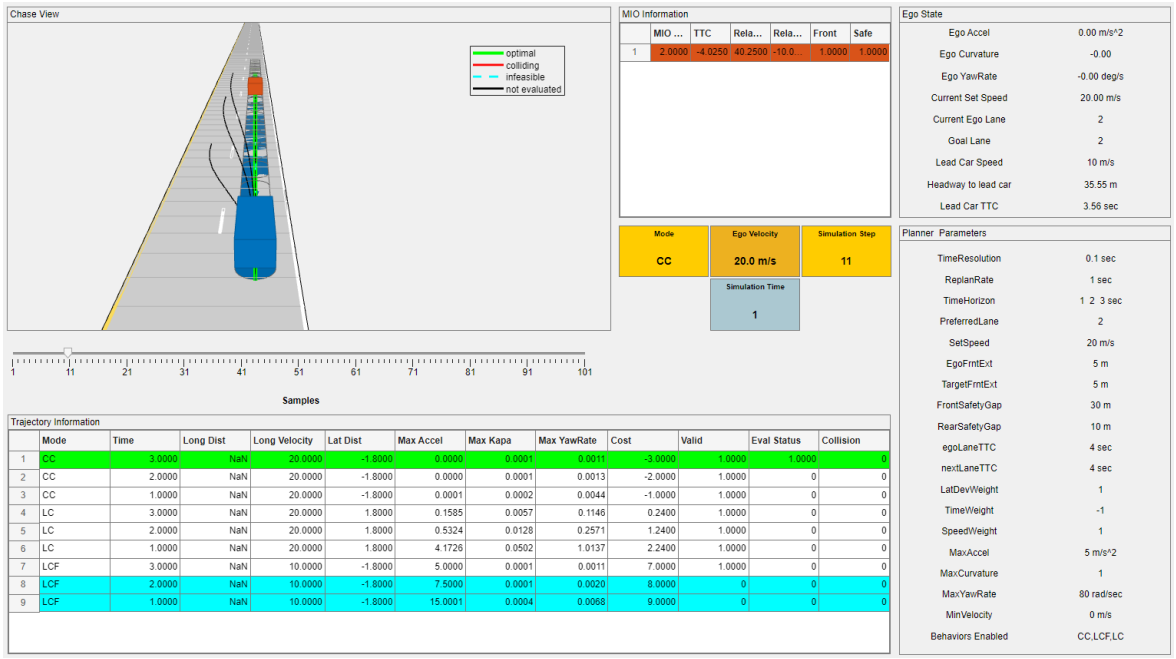

Figure 3.14: Visualization subsystem - Lane Change Simulation Analysis.

The interface has a cursor that allows to change the step to analyze. The information is grouped in the following tabs:

- Chase view: Provides visual feedback of the scenario in the selected step.
- Trajectory Information: Shows, for the selected step, the set of available trajectories ordered according to a preferential logic.
- For each one it is possible to display: the time horizon, the value of the cost parameter, the "IsValid", "IsEvaluated" and "IsColliding" flags, the speed, the distance from the center of the lane and the peak values of the associated quantities.
- MIO Information: Provides information on the MIO vehicles present in the scenario, such as: index, TTC, distance, relative speeds and Safe/Unsafe evaluation.
- Ego State: Provides information on the main quantities associated with the motion of the ego vehicle.
- Planner Parameters: Shows the values of the trajectory planner configuration parameters used for the simulation.

- Finally, through specific windows, it is possible to display the mode used (CC, LCF, LC), the speed of the ego vehicle, the step and the current simulation time.

3.1.7 Lane Change Controller

The Lane Change Controller subsystem is responsible for processing the input signals, which contain information about the trajectory to follow, and produces the commands necessary for controlling the ego vehicle. The structure of the subsystem is represented in Figure 3.15.

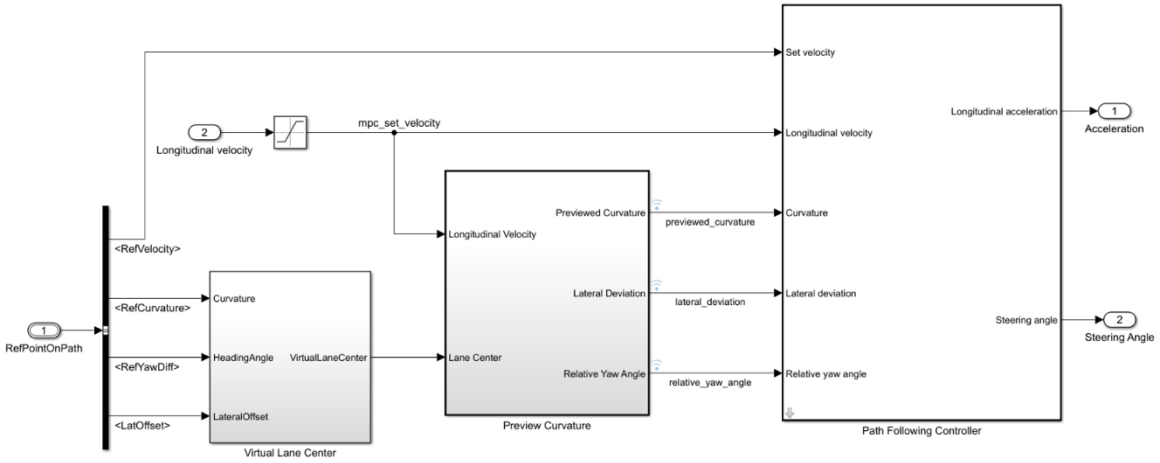

Figure 3.15: Lane Change Controller subsystem.

The subsystem input signals are the following:

- LatOffset: Absolute distance between the actual position of the ego vehicle and the reference position at the same time instant; this value quantifies the lateral error, represented in Figure 3.15 as $E_{lateral}$.

  The formula used for the calculation is the following:

$$\text{LatOffset} = \sqrt{\left(\left(X_{ego} - X_{ref}\right)^2 + \left(Y_{ego} - Y_{ref}\right)^2\right)}$$

  The data needed to estimate this quantity are generally obtained from a GPS system with Real Time Kinematic (RTK) correction or from cameras capable of recognizing

lane markings.

- Heading Angle: The difference between the actual yaw angle of the ego vehicle and the reference yaw angle at the same time instant. This value quantifies the angular error. In Figure 3.31, it is represented as Eyaw. The formula used for the calculation is:

$$\text{Heading Angle} = \psi_{\text{ref}} - \psi_{\text{ego}}$$

  The data needed to estimate this quantity are generally obtained from sensors capable of detecting the vehicle position within the road, such as cameras that recognize lane markings.

- RefCurvature and RefVelocity: Signals representing the reference curvature and velocity.

- Longitudinal velocity and Lateral acceleration: Signals representing the longitudinal velocity and lateral acceleration corresponding to the current actual state of the ego vehicle.

In the previous discussion, the commonly used sensors for estimating the quantities from which the errors are calculated have been mentioned. In this model, these quantities are calculated based on the exact position and orientation data output by the vehicle model. This is an important assumption to keep in mind during model analysis.

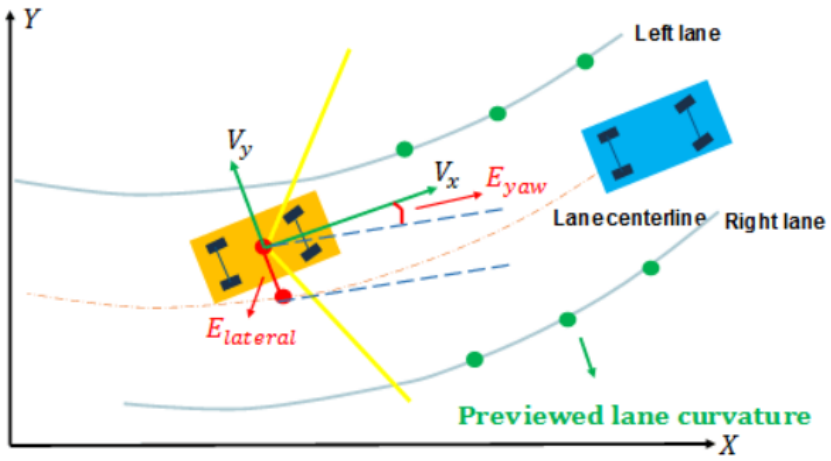

Figure 3.16: Path following controller [16].

The signals that quantify the lateral and angular errors are essential for the controller, as they allow for the correlation between the required trajectory corrections and the acceleration adjustments. Additionally, the presence of the longitudinal velocity signals allows the system to calculate the relative error; this operation is carried out within the Path Following Controller block.

The Virtual Lane Center block creates a Bus signal containing individual signals related to: lateral offset, angular error, and the curvature of the reference trajectory. These signals are processed in the Preview Curvature block, adapting them to the sign conventions required by the Path Following Controller block.

The Path Following Controller represents the heart of the subsystem dedicated to the controller. This block, available in the Automated Driving Toolbox libraries, simulates a path following controller that provides the necessary commands to follow the reference trajectory and maintain a safe distance from the vehicle ahead. However, in this model, the safety distance maintenance function has been disabled, as the planner determines the correct trajectory based on the positioning of all the vehicles in the scenario.

MPC Controller

The Path Following Controller combines the functionalities of an LKA and an ACC, using an adaptive model predictive controller or adaptive MPC.

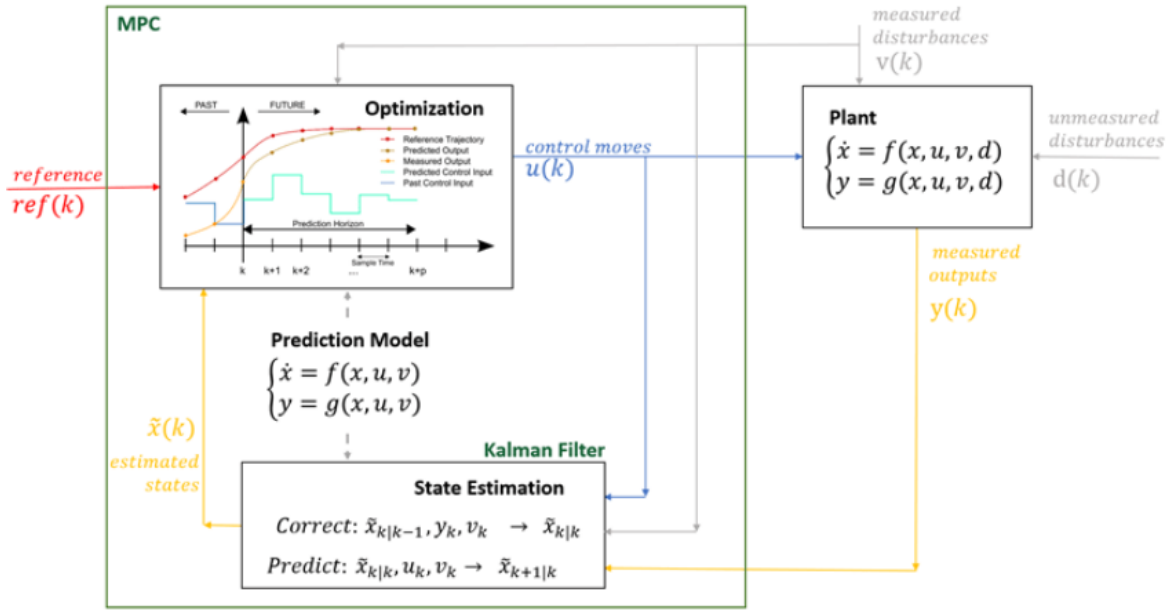

Figure 3.17: Basic control cycle [17].

Figure 3.17 shows the basic operation scheme of an MPC controller. Given a model to be controlled (Plant), the MPC is able to estimate the states of the controller and predict the outputs of the Plant thanks to an integrated vehicle model. Using the predicted outputs, it programs a sequence of control actions aimed at solving an optimization problem that minimizes a cost function on a given time horizon. This function is aimed at reducing the errors between the reference and the output of the Plant. At each time step, the controller performs the first programmed control action, sending the processed signals to the Plant model. At the next time step, the action is repeated [17]. It is important to note that, in the specific case of this study, no measured or unmeasured disturbances were included in the model and, as already anticipated, the assumption of knowing the actual states of the vehicle exactly was introduced; therefore, estimates are made only in the planning phase.

The MPC controller predicts the future behavior of the vehicle model, using a further integrated model of the linear-time invariant (LTI) type. Where LTI is defined as a system

that produces output signals starting from input signals subject to linearity and time invariance constraints.

The adaptive MPC represents an improvement of the standard MPC controller, and is designed to work with models with non-linear characteristics or that vary strongly over time. It uses a fixed model structure, of the LTI type, but unlike the standard MPC controller, it updates the parameters of the integrated model and the nominal conditions at each time step, improving the overall response of the controller [18]. The parameter updated at each time step in the original model is the longitudinal velocity (v_x). Subsequently, the update of the cornering stiffness parameters was also implemented, given the strong variation that they undergo in the non-linear operating range. The adaptive MPC used in this study combines the functionalities of an ACC, dedicated to producing a command on the accelerator, with those of an LKA, dedicated to producing a command on the steering wheel. The two mathematical models and the integrated model, including the ACC and LKA models, are described in state space by the following system of equations:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

Where:

- A, B, C, D are the matrices describing the system.
- x(t) defines the states of the dynamic system.
- u(t) indicates the input of the system.
- y(t) corresponds to the output of the system.

Adaptive Cruise Control Predictive Model

The Adaptive Cruise Control Predictive Model integrated in the controller provides the longitudinal velocity v_x as output, receiving the longitudinal acceleration a_x as input. The state equations are reported in extended form:

$$\begin{cases} \tau \dot{a}_{x,out} + a_{x,out} = a_{x,in} \\ \dot{v}_{x,out} = a_{x,out} \end{cases}$$

Where τ is the time constant. The states of the dynamic system x(t), the output y(t) and the input u(t) are defined as follow:

$$x(t) = \begin{bmatrix} a_{x,out} \\ v_{x,out} \end{bmatrix};\ y(t) = v_{x,out};\ u(t) = a_{x,in}$$

The matrices of the state-space model are defined as follows:

$$A_1 = \begin{bmatrix} -\frac{1}{\tau} & 0 \\ 1 & 0 \end{bmatrix}; B_1 = \begin{bmatrix} \frac{1}{\tau} \\ 0 \end{bmatrix}; C_1 = [0 \quad 1]; D_1 = 0$$

The purpose of this model is to introduce a delay in the acceleration signal, in order to simulate the delay in actuating the command and the inertia of the vehicle. Therefore, the dynamics between acceleration and vehicle speed is governed by the following transfer function:

$$\frac{v_{x,out}}{a_{x,in}} = G(s) = \frac{1}{s(\tau s + 1)}$$

Lane-Keeping Predictive Model

The Lane-Keeping Predictive Model integrated in the controller, provides as output the lateral velocity v_y and the angular velocity $\dot{\psi}$, receiving as input the steering angle at the wheels δ. These are the equations that describe the motion of a single track or bicycle vehicle model. The state equations are reported in extended form:

$$m\left(\dot{v}_y + v_x\dot{\psi}\right) = -Cr\frac{v_y - b\dot{\psi}}{v_x} + C_F\delta_f - C_F\frac{v_y + a\dot{\psi}}{v_x}$$

$$I_z\ddot{\psi} = bC_R\frac{v_y - b\dot{\psi}}{v_x} + C_Fa_f - C_Fa\frac{v_y + a\dot{\psi}}{v_x}$$

Where:

- v_x: Longitudinal velocity of the vehicle.
- v_y: Lateral velocity of the vehicle.
- m: Total mass of the vehicle.
- $\dot{\psi}$: Yaw rate of the vehicle.
- I_Z: Moment of inertia about the z-axis.
- δ_f : Steering angle.
- a: Front half-step.
- b: Rear half-step.
- C_F: Front axle cornering stiffness.
- C_R: Rear axle cornering stiffness.

The states of the dynamic system x(t), the output y(t) and the input u(t) are defined:

$$x(t) = \begin{bmatrix} v_y \\ \dot{\psi} \end{bmatrix};\ y(t) = \begin{bmatrix} v_y \\ \dot{\psi} \end{bmatrix};\ u(t) = \delta_f$$

The state-space model matrices are defined as follows:

$$A_2 = \begin{bmatrix} \frac{-(C_F + C_R)}{mv_x} & \frac{-(C_Fa + C_Rb)}{mv_x} - v_x \\ \frac{-(C_Fa + C_Rb)}{I_Zv_x} & \frac{-(C_Fa^2 + C_Rb^2)}{I_Zv_x} \end{bmatrix};\ B_2 = \begin{bmatrix} \frac{1}{m} \\ \frac{a}{I_Z} \end{bmatrix};\ C_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix};\ D_2 = 0$$

Combined Model

The two models are combined into a single integrated state-space model having the following state vector and output vector.

$$x(t) = \begin{bmatrix} a_{x,out} \\ v_{x,out} \\ v_{y,out} \\ \dot{\psi}_{out} \end{bmatrix}; u(t) = \begin{bmatrix} a_{x,in} \\ \delta_f \end{bmatrix}$$

The matrices of the overall state-space model are defined as follows.

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}; B = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix}; C = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix}; D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}$$

This integrated model receives precisely these two parameters as input signals ($a_{x,in}, \delta_f$). They are processed by the state-space model to produce the signals of longitudinal acceleration (processed by a transfer function G(s)), longitudinal and lateral velocity, and yaw velocity. It should be remembered that the model is of the LTI type, therefore the configuration parameters of the matrices (A, B, C and D) are constant. However, the controller is adaptive, so some of the parameters of the matrices mentioned are updated at each step. For this reason, there is feedback of the calculated longitudinal velocity and the input of the cornering stiffnesses in the state-space block, which are calculated based on the current value of lateral acceleration. Finally, these data are subsequently processed to calculate the lateral and angular errors, e_y and e_ψ respectively.

MPC Controller Cost Function

The Model Predictive Control (MPC) controller solves a quadratic optimization problem (QP) at each control interval. The solution of the problem allows to identify the output variables to be provided to the Plant model in the next simulation step. The problem is solved with the aim of minimizing a given cost function.

The standard cost function $J(z_k)$ is composed of four terms, each of which is centered on a

particular aspect of the controller.

$$J(z_k) = J_y(z_k) + J_u(z_k) + J_{\Delta u}(z_k) + J_\epsilon(z_k)$$

Where z_k represents the solution of the quadratic optimization problem solved at simulation time k, also called "vector of QP decision variables".

$$z_k^T = [u(k|k)^T, u(k+1|k)T, \ldots, u(k+p-1|k)^T, \epsilon_k]$$

Output reference tracking ($J_y(z_k)$):

The MPC controller must keep the system outputs close to reference values, so a performance measure is performed in terms of distance from the given reference.

This quantity is the first of the four terms that make up the cost function [19].

$$J_y(z_k) = \sum_{j=1}^{n_y} \sum_{i=1}^{p} \left\{ \frac{w_{i,j}^y}{s_j^y} \left[r_j(k+i \mid k) - y_j(k+i \mid k) \right] \right\}^2$$

Where:

- k: Current step of the simulation.
- p: Predictive horizon.
- n_y: Number of variables in output from the Plant.
- z_k: Vector of the decision variables of the QP.
- $y_j(k+i \mid k)$: Predicted values of the j-th output of the i-th prediction step.
- $r_j(k+i \mid k)$: Reference values of the j-th output of the i-th prediction step.
- s_j^y: Scale factor of the j-th output of the Plant.
- $w_{i,j}^y$: Calibration parameter that establishes the weight of the j-th output of the Plant in the i-th prediction step.
- ϵ_k: Slack variable in simulation step k, which provides an indication of the deviation

from the current reference.

The controller receives the reference values $r_j(k+i \mid k)$ for the entire time horizon. It predictively calculates the Plant outputs $y_j(k+i \mid k)$, which depend on the manipulated variables zk, the disturbances and the state estimation, remembering that, in the case in question, the disturbances are neglected and the exact knowledge of the current states of the Plant is assumed.

Tracking of manipulated variables ($J_u(z_k)$):

$$J_u(z_k) = \sum_{j=1}^{n_u} \sum_{i=1}^{p-1} \left\{ \frac{w_{i,j}^u}{s_j^u} \left[u_j(k+i \mid k) - u_{j,target}(k+i \mid k) \right] \right\}^2$$

Where:

- n_u: Number of manipulated variables.
- $u_{j,target}(k+i \mid k)$: Reference value of the j-th manipulated variable in the i-th prediction step.
- s_j^u: Scale factor of the j-th manipulated variable.
- $w_{i,j}^u$: Calibration parameter that sets the weight of the j-th manipulated variable in the i-th prediction step.

The controller receives the reference values of the manipulated variables $u_{j,target}(k+i \mid k)$ for the entire time horizon.

Reduction of variation of the manipulated variable in the interval T_s ($J_{\Delta u}(z_k)$):

$$J_{\Delta u}(z_k) = \sum_{j=1}^{n_u} \sum_{i=1}^{p-1} \left\{ \frac{w_{i,j}^{\Delta u}}{s_j^u} \left[u_j(k+i \mid k) - u_{j,target}(k+i \mid k) \right] \right\}^2$$

Where $w_{i,j}^{\Delta u}$ is the calibration parameter that establishes the weight of the variation of the j-

th manipulated variable in the i-th prediction step.

Violation of constraints ($J_\epsilon(z_k)$):

In the practice of using this controller, one can inevitably incur in the violation of the imposed thresholds. Consequently, the model uses soft constraints that in some cases can be violated. The slack variable ϵ_k quantifies the violation of the constraint in the worst case.

$$J_\epsilon(z_k) = \rho_\epsilon \epsilon_k^2$$

Where ρ_ϵ represents the weight of the penalty for the violation of the given constraint.

MPC controller error calculation

Examining what has been described so far, it emerges that for each simulation instant the controller establishes a strategy to follow based on the predictions made with the integrated model, minimizing the cost function exposed in the previous subsection.

As can be seen from the Tracking of manipulated variables equation, within the cost function there is a contribution that depends on the error calculated in the entire forecast horizon for each j-th output parameter y_j of the integrated vehicle model with respect to the relative reference r_j.

The goal is to write the system of state equations as a function of the orientation error e_ψ and the position error e_y, in order to simplify their implementation within the cost function.

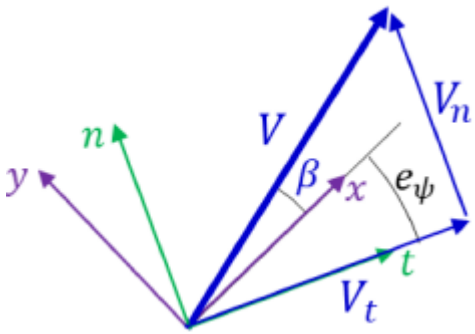

Figure 3.18: Reference system fixed with the vehicle and the reference trajectory.

As a first step, it is necessary to write the components of the velocity of the vehicle ego with respect to a mobile reference system integral with the reference trajectory, represented in green in Figure 3.18.

$$\begin{cases} V_t = \dfrac{V\cos(\beta + e_\psi)}{1 \pm \rho_{ref} e_y} \\ \dot{e}_y = V\sin(\beta + e_\psi) \end{cases}$$

Then the fundamental equation of kinematics is applied.

$$V_t = \dot{s} \pm \dot{\psi}_{ref} e_y = \dot{s} \pm \rho_{ref} \dot{s} e_y$$

The lateral deviation error can be expressed as:

$$V_n = \dot{e}_y = V\sin(\beta + e_\psi)$$

Therefore, the following system can be written, which describes the relationship between the vehicle velocity V and the velocity along the reference trajectory $\dot{s}$.

$$\begin{cases} \dot{s} = \dfrac{V\cos(\beta + e_\psi)}{1 \pm \rho_{ref} e_y} \\ \dot{e}_y = V\sin(\beta + e_\psi) \end{cases}$$

These equations can be linearized by assuming small trim angles β and orientation errors e_ψ in the vicinity of the equilibrium conditions. The trigonometric functions can be rewritten as follows:

$$V\cos(\beta + e_\psi) = V\left[\cos(\beta)\cos(e_\psi) - \sin(\beta)\sin(e_\psi)\right] \simeq V - V\,\beta e_\psi = u - v e_\psi$$

$$V\sin(\beta + e_\psi) = V\left[\sin(\beta)\cos(e_\psi) + \cos(\beta)\sin(e_\psi)\right] \simeq v + u e_\psi$$

So, the previous system is rewritten as follow:

$$\begin{cases} \dot{s} = \dfrac{u - v e_\psi}{1 \pm \rho_{ref} e_y} \\ \dot{e}_y = v + u e_\psi \end{cases}$$

The velocity of the lateral deviation error is than obtained:

$$\ddot{e}_y = v + u e_\psi$$

Similarly, the velocity and acceleration of the angular error are obtained:

$$\dot{e}_{\psi} = \dot{\psi} - \dot{\psi}_{ref}$$

$$\ddot{e}_{\psi} = \ddot{\psi} - \ddot{\psi}_{ref}$$

Substituting the expressions of v, $\dot{v}$, $\psi, \dot{\psi}$ as a function of the two errors obtained in the equations of state of the single-track model, it is possible to write the associated state-space system as follows:

$$\dot{e} = A_p e + B_{p1}\delta + B_{p2}\dot{\psi}_{ref} + \ddot{\psi}_{ref}$$

Where:

$$e = \left[e_y, \dot{e}_y, e_{\psi}, \dot{e}_{\psi}\right]; A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{C_F + C_R}{mV} & \frac{C_F + C_R}{m} & \frac{C_F b + C_R a}{mV} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{C_R b - C_F a}{J_z V} & \frac{C_F a - C_R b}{J_z} & -\frac{C_F a^2 - C_R b^2}{J_z V} \end{bmatrix};$$

$$B = \begin{bmatrix} 0 \\ \frac{C_F}{m} \\ 0 \\ \frac{C_F a}{m} \end{bmatrix}; C = \begin{bmatrix} 0 \\ \frac{C_R b - C_F a}{mV} - V \\ 0 \\ -\frac{C_F a + C_R b^2}{J_z V} \end{bmatrix}$$

Changes introduced

Although the adaptive MPC controller is more suitable than a standard MPC for nonlinear models, the predictions are still made using a linear state-space model, different from the one used as a vehicle model in the simplifying assumptions introduced and in the structure. Consequently, the results are influenced by approximations that, in the most extreme cases, can have a significant impact.

For this reason, it was considered appropriate to modify the controller structure, inserting the cornering stiffness characteristics of the axles, specific to the vehicle model used.

In the original controller configuration, the cornering stiffness parameters required as input

were constant. However, to improve the performance of the model at high lateral accelerations, the characteristics of each axle were inserted using look-up tables. Specifically, the look-up tables receive as input the actual lateral acceleration value of the vehicle, returning the current values of $C_{\alpha F}$ and $C_{\alpha R}$.

This modification does not change the type of model integrated in the controller, since all predictions are made using the cornering stiffness values provided by the look-up tables, even for future instants. What this modification actually introduces is a further adaptation of the parameters of the integrated model to the different conditions of the vehicle in the various steps of the simulation. With the proposed solution, a sensitivity to the non-linear response of the tires is introduced into the model integrated in the controller. The introduced characteristics are shown in Figure 3.19.

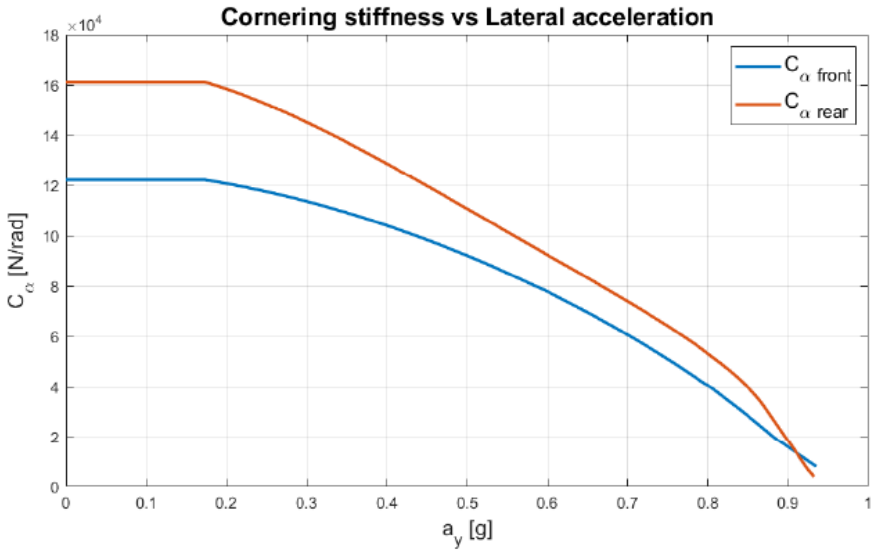

Figure 3.19: Trend of the cornering stiffness of the axles.

Following the implementation of this modification, simulations were performed to test its actual benefits. The results obtained are reported in the fourth chapter, specifically in paragraph 4.5.

The modification introduced regarding the cornering stiffnesses $C_{\alpha,F/R}$ presents some limitations in the operating range since the formula used for the calculation of the lateral forces in describing the state space of the bicycle model is the following:

$$F_{yR} = C_{\alpha}\alpha_{R}$$

In the case of operation in a strongly non-linear field, it returns a value of F_y higher than the real one. This happens because, in this operating range, the lateral force equation no longer takes the form of a straight line passing through the origin, but becomes non-linear.

For further clarity regarding the nonlinear tire model, please refer to the next paragraph.

Consequently, this criticality represents a limit of the current model, from which to start again in the case of future implementations.

Although the introduced modifications have extended the correct operating range of the MPC controller, when the vehicle characteristics deviate excessively from the linear approximations, the controller outputs are not sufficiently accurate.

To provide a practical confirmation of what has just been stated, a simulation was conducted by providing an ideal trajectory that requires high controller performance, characterized by lateral accelerations of up to 1 g.

The scenario adopted is a straight road, along which an ego vehicle and a lead vehicle travel. The maneuver performed by the ego vehicle is a double lane change, similar to the one presented in the example introduced in the previous chapter. Figure 3.20 shows a comparison between the reference trajectory and the actual trajectory.

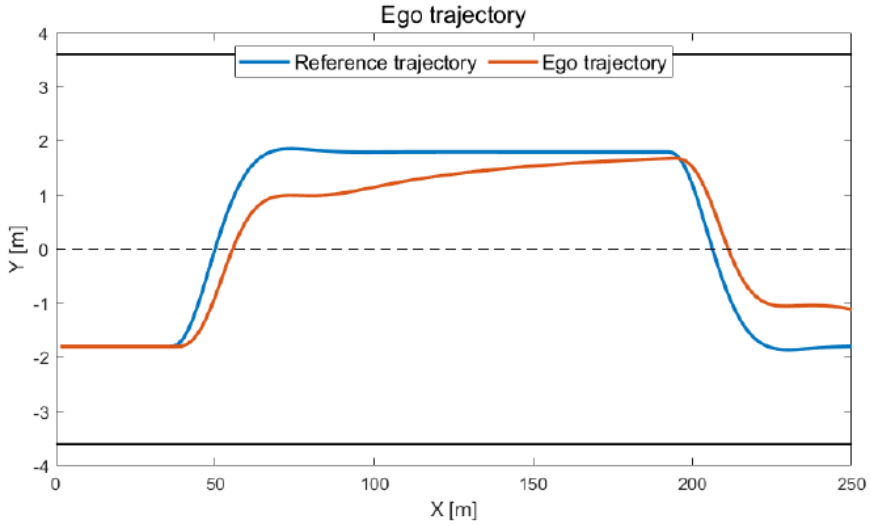

Figure 3.20: Comparison of actual/reference trajectory with high required performance.

As can be observed from the comparison of the trajectories and from the trend of positioning and orientation errors in Figure 3.21, following such a high lateral acceleration request, the vehicle is unable to follow the reference trajectory precisely.

The orientation error is compensated for quickly after the steering command is actuated, well before the ego vehicle reaches the center of the overtaking lane.

This causes a residual lateral distance error, which is compensated for by excessively long transients.

This behavior is exacerbated when the performance demand is even higher.

It should be emphasized that, although the ego vehicle struggles to follow the reference trajectory, it does not experience directional instability and the system states do not diverge.

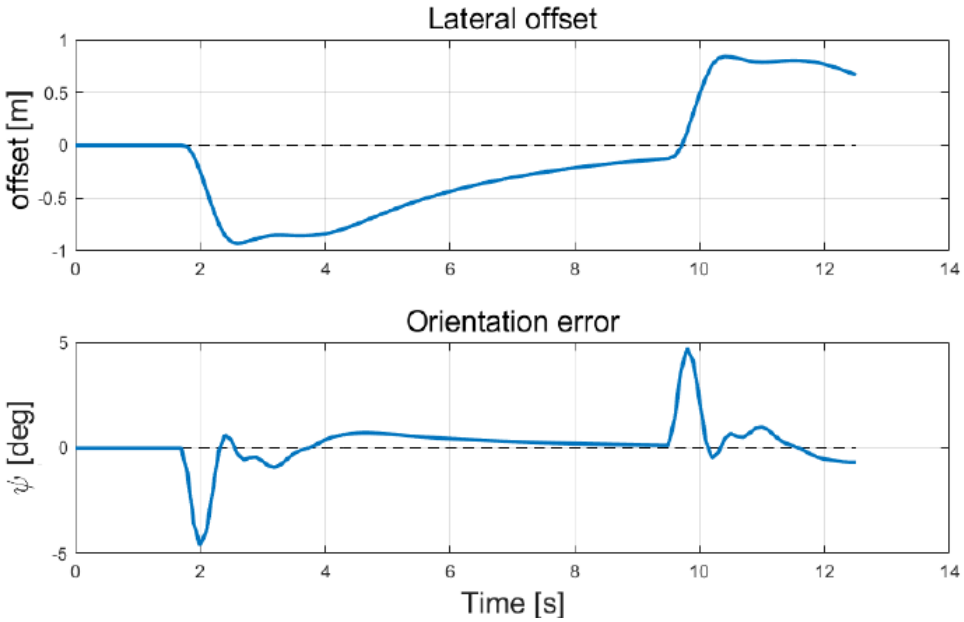

Figure 3.21: Trajectory controller errors at high performance demands.

3.1.8 Vehicle Model

The Vehicle Dynamics subsystem has integrated the vehicle model adopted to simulate the behavior of the ego vehicle in relation to the inputs received from the controller. It is a 4 degree-of-freedom model of the Double track with roll and non-linear tire model type.

The general structure of the subsystem is represented in Figure 3.22.

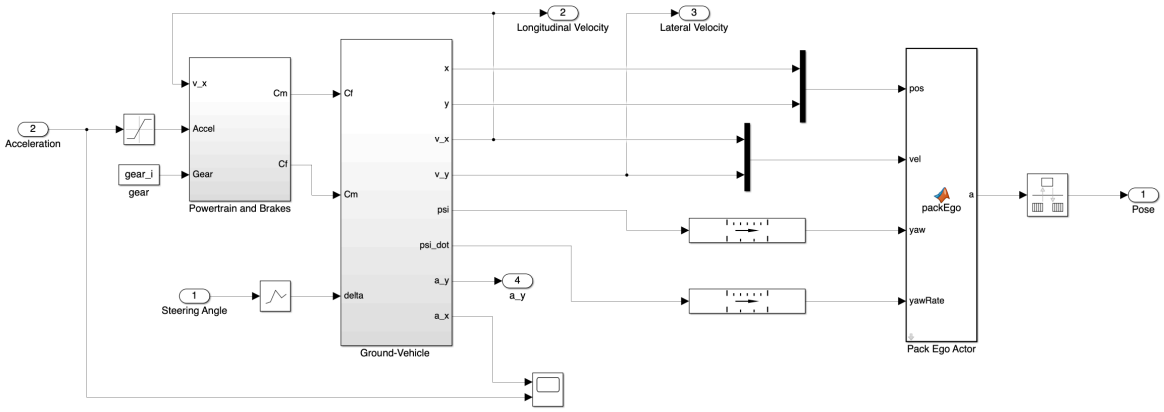

Figure 3.22: Vehicle Model subsystem.

Furthermore, a detailed overview of the Ground-Vehicle subsystem is provided in Figure 3.23.

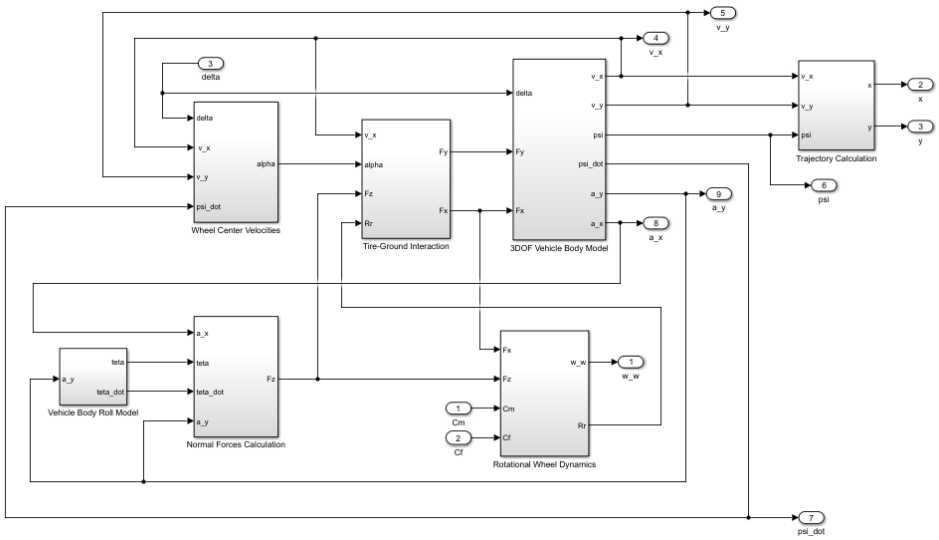

Figure 3.23: Ground-Vehicle subsystem.

Powertrain and brakes

The vehicle drivetrain is designed to allow the vehicle to switch between front-wheel drive (FWD), rear-wheel drive (RWD) and all-wheel drive (AWD). An open differential is installed, which allows the wheels to turn at different speeds, which is especially essential when cornering. When a vehicle turns, the outside wheels have to travel a greater distance than the inside wheels. An open differential allows power to be distributed between the wheels without locking up any of them, allowing for a smooth ride.

In the engine and transmission block, to calculate the torque needed to obtain a certain longitudinal acceleration (a_x), the aerodynamic and rolling resistance are considered, as well as the vehicle mass and the wheel radius. The main formula describing this relationship is:

$$C = R_0 \cdot (m \cdot a_x + F_{aero} + F_r)$$

Here, C represents the torque needed to generate the desired acceleration. The undeformed wheel radius R_0 directly affects the calculation, as do the vehicle mass m and the longitudinal acceleration a_x provided by the controller. The resistive forces are aerodynamic resistance (F_{aero}) and rolling resistance (F_r).

We then proceed with the calculation of the engine torque C_m, which depends on the selected transmission ratio:

$$C_m = C/\text{gear}$$

The transmission ratio (gear) determines how much torque the engine must supply to the wheels to satisfy the required acceleration.

If the calculated torque C is positive, this implies a request for torque to the engine. Furthermore, using two look-up tables, the minimum and maximum torque values that can be delivered by the engine at the given rpm are calculated; these are the saturation values of the calculated C_m parameter.

Moving on to the braking system, the system has been designed to equally distribute the braking action between the four wheels. If the torque C is negative, it indicates that the vehicle must brake. In this case, the total braking torque C_f is calculated as the absolute value

of C, and therefore divided evenly between the four wheels:

$$C_{f,tot} = | C |$$

The braking torque for each wheel is then determined as:

$$C_f = C_{f,tot}/4$$

Steering angle actuation

Regarding the steering angle input, this is processed through a transfer function to simulate the delay of the steering actuation system. The transfer function used is the following:

$$\frac{\delta_f}{\delta_{fcontroll}} = \frac{1}{\tau s + 1}$$

Following specific tests conducted through simulations, a fast actuation system was chosen, with a value of τ equal to 0.01 s. Furthermore, for simplicity in formulation and analysis, the steering capability is limited exclusively to the front axle wheels:

$$\delta_R = 0$$

So, from now on we will refer to δ_f simply as δ.

3DOF vehicle model

To describe the vehicle motion in three degrees of freedom corresponding to longitudinal (v_x), lateral (v_y) and rotational ($\dot{\psi}$) around the vertical axis (yaw), the following equations are employed. The latter are based on the vehicle dynamic equilibrium equations, which derive from Newton's second law applied to translation and rotation.

The longitudinal component of the equation of motion, which takes into account the sum of the longitudinal forces F_x and the effect of the yaw angular velocity, is:

$$m \cdot a_x = F_x - m \cdot v_y \cdot \dot{\psi} + F_{aero}$$

Where F_{aero} is the aerodynamic resistance.

The lateral component of the motion, which includes the lateral forces (F_y) and the effect of the yaw angular velocity, is:

$$m \cdot a_y = F_y + m \cdot v_x \cdot \dot{\psi}$$

The yaw dynamics (rotation about the vertical axis) is governed by the torque resulting from the lateral forces on the front and rear tires. This equation describes the change in the yaw angular velocity:

$$I_z \cdot \dot{\psi} = a \cdot F_{yF} - b \cdot F_{yR}$$

Where:

- I_z is the moment of inertia about the vertical axis.
- a and b are the distances of the center of gravity from the front and rear axles, respectively.
- F_{yF} and F_{yR} are the lateral forces on the front and rear axles.

The longitudinal acceleration a_x and the lateral acceleration a_y are derived from the longitudinal and lateral velocities respectively:

$$a_x = \frac{F_x - m \cdot v_y \cdot \dot{\psi} + F_{aero}}{m}$$

$$a_y = \frac{F_y + m \cdot v_x \cdot \dot{\psi}}{m}$$

In Figure 3.24 the free body diagrams of the vehicle are reported, realized respectively in the X-Y plane.

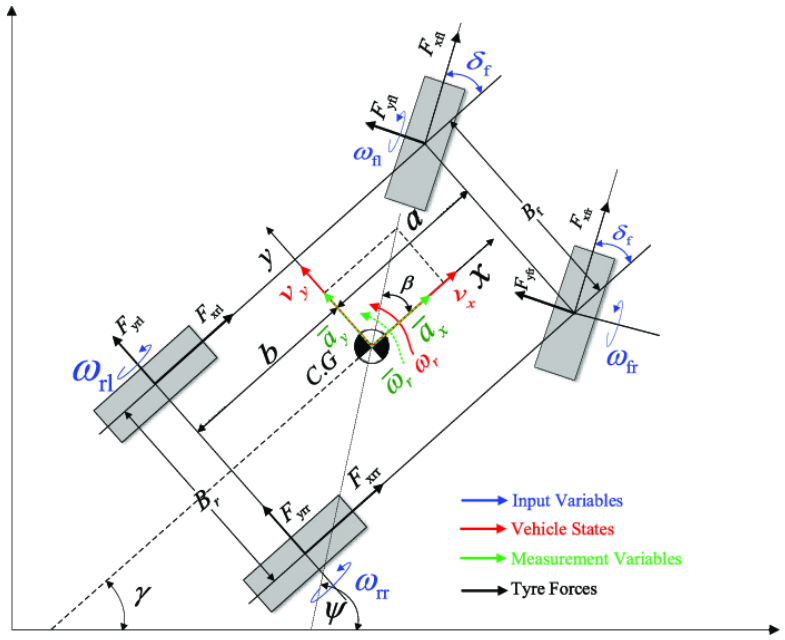

Figure 3.24: 3DOF vehicle body model with four wheels.

Tire-ground interaction

One of the main features of the model is that of having an integrated non-linear tire model; specifically, to deal with the nonlinear characteristics of tires, the front and rear tire longitudinal force F_x and lateral force F_y were expressed in the Pacejka "Magic Formula" as a function of the slip angle α [20].

The following formula was adapted to determine the lateral force F_y.

$$F_y(\alpha) = D \cdot \sin\{C \cdot \tan^{-1}[B \cdot (1 - E) \cdot \alpha + E \cdot \tan^{-1}(B \cdot \alpha)]\}$$

Where B, C, D, and E are constant coefficients that vary with the vertical load, calculated from experimental data [21]:

- B: Stiffness factor (related to the cornering stiffness) can be expressed as $B = S_t/(C \cdot D)$.
- C: Shape factor (typically close to 1, representing how non-linear the force response is) is calculated as $C = 2\left[1 - \left(\frac{1}{\pi}\right)\sin^{-1}R_{sp}\right]$, where R_{sp} describes the relationship between the stiffness of a tire and its peak lateral force capabilities

- D: Peak factor (maximum lateral force) corresponds to $D = F_{y,max}$
- E: Curvature factor (how quickly the lateral force saturates) is determined by

$$E = \frac{(\pm)\tan\left(\frac{\pi}{2C}\right) - B\alpha_{max}}{\tan^{-1(B\alpha_{max})} - B\alpha_{max}}.$$

The cornering stiffness C_α is typically defined as the slope of the lateral force F_y with respect to the slip angle α at small slip angles:

$$C_\alpha = \frac{dF_y}{d\alpha}$$

In the non-linear case, the cornering stiffness changes dynamically with the slip angle α, and the full expression of $C_\alpha(\alpha)$ involves differentiating the lateral force equation with respect to α. The cornering stiffness is maximum at zero slip angle and decreases as the slip angle increases. Another notable observation is the slope at which the curves fall. It drops off sharply, indicating a quicker alignment after the curve. Imagine having a car that has already been turned in, and when it is turned back to its neutral position, this tire would give up lateral grip much more quickly, making it easier to transfer lateral grip to its longitudinal grip, giving the driver the confidence to accelerate.

To make the tire model more realistic, delays have been introduced in the response of the longitudinal and lateral forces. This reflects reality, where forces do not develop instantaneously when the tire deforms but require some time to stabilize. These delays are modeled by a differential equation that describes the dynamic behavior of the forces:

$$\frac{1}{\tau} \cdot \dot{F}_{rit} + F_{rit} = F_{pac}$$

where:

- F_{rit} is the delayed (effective) force acting on the tire.
- F_{pac} is the force calculated with the Magic Formula.
- τ is the time constant characterizing the delay, expressed as the ratio between the longitudinal velocity of the wheel center and the relaxation length.

The relaxation length is the distance traveled by the tire during the transition from the "initial

deformation" state to a steady state when a longitudinal or lateral force is applied. In other words, the relaxation length is the distance needed for the tire to reach its final behavior in response to an external force.

Vehicle Body Roll Model and Normal Forces Calculation

The body roll block is responsible for describing the behavior of the vehicle when subjected to lateral forces. The main input to this block is the lateral acceleration, which generates a rolling moment and leads to a lateral inclination of the vehicle.

Simplifying assumptions used in the model include:

- Suspension roll stiffness (K_{susp}), which is the combined contribution of springs and anti-roll bars:

$$K_{susp} = K_{spring} - roll + K_{ARB}$$

- Suspension roll damping (C_{roll}), due to shock absorbers.
- Roll stiffnesses and damping are calculated separately for the front and rear axles:

$$K_{roll} = K_{susp}, F + K_{susp}, R$$

$$C_{roll} = C_{damp}, F + C_{damp}, R$$

- The roll axis is fixed, which further simplifies the model.

Additionally, two percentages of contribution to roll stiffness are defined:

- Percentage of anti-roll bar stiffness compared to total suspension stiffness:

$$\%ARB = \left(\frac{K_{ARB}}{K_{sosp}}\right) \cdot 100$$

- Percentage of front bar stiffness compared to total anti-roll bar stiffness:

$$\%ARB_{front} = \left(K_{ARB}, \frac{F}{K_{ARB}}, tot\right) \cdot 100$$

To calculate the distribution of vertical forces F_z on the front and rear wheels, the combined effects of longitudinal acceleration a_x, lateral acceleration a_y and roll angle θ are considered.

Longitudinal load transfer occurs due to vehicle acceleration or deceleration and causes a change in normal forces between the front and rear wheels. The normal forces on the front left F_{zFL} and right F_{zFR} wheels, as well as those on the rear left F_{zRL} and right F_{zRR} wheels, are calculated as:

$$F_{zFL} = \frac{m \cdot g \cdot b - m \cdot a_x \cdot H_g}{2L}$$

$$F_{zFR} = \frac{m \cdot g \cdot b - m \cdot a_x \cdot H_g}{2L}$$

$$F_{zRL} = \frac{m \cdot g \cdot a + m \cdot a_x \cdot H_g}{2L}$$

$$F_{zRR} = \frac{m \cdot g \cdot a + m \cdot a_x \cdot H_g}{2L}$$

Where $L = a + b$ is the wheelbase of the vehicle.

In addition to the longitudinal transfer, the lateral load transfer is affected by the lateral acceleration a_y, which causes a change in the forces between the left and right wheels of each axle. This transfer is calculated with the following formula:

$$\Delta F_{zL} = \frac{m \cdot a_y \cdot H_g}{T_w}$$

Where T_w is the track width (width between the left and right wheels). This effect acts by reducing the force on the left wheels and increasing that on the right wheels. Therefore, the updated normal forces become:

$$F_{zFL} = \frac{m \cdot g \cdot b - m \cdot a_x \cdot H_g}{2L} - \frac{m \cdot a_y \cdot H_g}{2T_w}$$

$$F_{zFR} = \frac{m \cdot g \cdot b - m \cdot a_x \cdot H_g}{2L} + \frac{m \cdot a_y \cdot H_g}{2T_w}$$

$$F_{zRL} = \frac{m \cdot g \cdot a + m \cdot a_x \cdot H_g}{2L} - \frac{m \cdot a_y \cdot H_g}{2T_w}$$

$$F_{zRR} = \frac{m \cdot g \cdot a + m \cdot a_x \cdot H_g}{2L} + \frac{m \cdot a_y \cdot H_g}{2T_w}$$

Furthermore, the effect of the roll is integrated into the calculation of the normal forces. The roll angle is determined by the roll stiffness K_{roll} and the lateral acceleration, and is given by:

$$\theta = \frac{m \cdot a_y \cdot H_g}{K_{roll}}$$

Rotational Wheel Dynamics

The angular velocity of the wheels ω_r is determined by the dynamic equilibrium between the applied torques (driving and braking), the longitudinal forces and the moment of inertia of the wheel. The differential relationship that governs the variation of the angular velocity is the following:

$$\dot{\omega}_r = \frac{C_m - C_f - F_x \cdot R_r}{J_{wheel}}$$

Where R_r is the reduced radius of the wheel obtained after introducing the hypothesis of radial deformation of the tire and J_{wheel} is the moment of inertia of the wheel.

The reduced radius depends on the radial stiffness of the tire, which describes how much the tire deforms under load. The relationship used to represent this deformation is as follows:

$$R_r = R_0 - \frac{F_z}{K_r}$$

Where K_r is the radial stiffness of the tire, which represents the resistance of the tire to vertical compression under load.

Pack Actor

The pack actor is a function block that “packs” the vehicle position, velocity, orientation (yaw), and yaw rate information, along with a unique identifier (egoActorID). The output is a “bus” of type BusVehiclePose, which is then used to interface with scene readers or other model components that require a compact representation of the vehicle state.

Limits of the bicycle model

The core of the original vehicle model was a single track model also known as the bicycle model, which is one of the most common simplifications used to describe the lateral and longitudinal dynamics of a vehicle, since it assumes that the vehicle can be represented as a system of two virtual wheels, one on the front axle and one on the rear axle.

It was later decided to implement the vehicle model because the bicycle model has several limitations that reduce its accuracy in complex scenarios or in particular driving conditions.

1. Neglect of the vehicle aerodynamics
   In real world scenarios, aerodynamic forces and more precisely, the interaction between the wind and the vehicle, have significant effects on stability and a significant role in the overall dynamics of the vehicle. Especially in situations of strong crosswinds or at high speeds, the latter can generate unwanted forces that alter the trajectory of the vehicle, requiring corrective interventions by the driver or the vehicle dynamic control system.
2. Limit during high lateral accelerations
   The single-track model is accurate up to about 0.4g of lateral acceleration, beyond which it fails to correctly represent the dynamic behavior of the vehicle. Under these conditions, the lateral forces no longer follow a linear relationship with the slip angle, leading to a loss of accuracy in the model predictions.
3. Roll and pitch negligence
   One of the main limitations of the bicycle model is the exclusion of the degrees of

freedom associated with the roll and pitch of the vehicle. Since the model only considers longitudinal, lateral, and yaw motion, the degrees of freedom associated with roll and pitch are not covered. During rapid maneuvers, sudden braking, or heavy acceleration, roll and pitch affect the load distribution on the tires, changing traction and stability. For example, during high-speed cornering scenarios, the vehicle tends to lean sideways, which then affects lateral forces and can cause a loss of traction. On the other hand, pitch occurs during braking and acceleration, when the load transfer between the axles changes the available traction, causing an imbalance that the bicycle model does not accurately predict.

4. Lack of nonlinear tire modeling

   The bicycle model assumes that the forces generated by the tires are proportional to the slip angles and that the dynamics are linear. In reality, especially in extreme riding conditions, tires operate nonlinearly. It is therefore possible to define a limit of 4 m/s^2 for lateral acceleration in the bicycle model related to the fact that the lateral force generated by the tires no longer increases linearly with the slip angle due to the tire saturation limit.

5. Neglecting tire deformations and suspension behavior

   The bicycle model assumes that tires are rigid and does not account for tires deformation or suspension behavior under load, which generate effects that influence lateral and longitudinal forces.

   Furthermore, the lack of consideration for vertical dynamics, the asymmetric wheel forces the behavior of the suspension impacting the load transfer between the wheels during braking, acceleration and cornering, reduce the accuracy of the model in high-performance contexts.

6. Simplified road surface interaction

   The bicycle model does not take into account the complex interaction between the vehicle and the road surface, especially when there is reduced grip such as wet, icy or gravel roads. The scenarios described above lead to an alteration of traction and stability that this simplified model cannot cope with because it does not take into

account the effects that changes in road friction have on lateral and longitudinal forces, limiting its effectiveness in real-world riding conditions where such variations are common.

7. Simplified approach to distributing forces between wheels

   The model reduces the vehicle to two virtual wheels, neglecting the distribution of forces between the left and right wheels of the vehicle. In reality, the forces on the wheels vary significantly during maneuvers, as stated previously, affecting the behavior of the vehicle. Lateral load transfer is not captured by the bicycle model, which assumes a uniform distribution of forces on the axles. This limits its ability to predict phenomena such as oversteer or understeer.

3.2 Simulation Scenario

The primary input of the previously described model is the simulation scenario. During the development of this thesis, the model has been tested with different scenario variants, which have been appropriately described below.

The use of the Driving Scenario Designer Tool simplified the creation of scenarios thanks to the graphical interface and the intuitiveness of the MATLAB tool. This tool generates an output with all the information of the scenario in a Scenario file or, alternatively, the data can be inserted in a function that generates the Scenario file. In the model used, the latter option was adopted.

3.2.1 Driving Scenario Designer

As anticipated, the Driving Scenario Designer is an application of the MATLAB software that allows the user to design driving scenarios to test their autonomous driving systems [12]. With this tool it is possible to create a scenario from scratch using a special drag-and-drop interface.

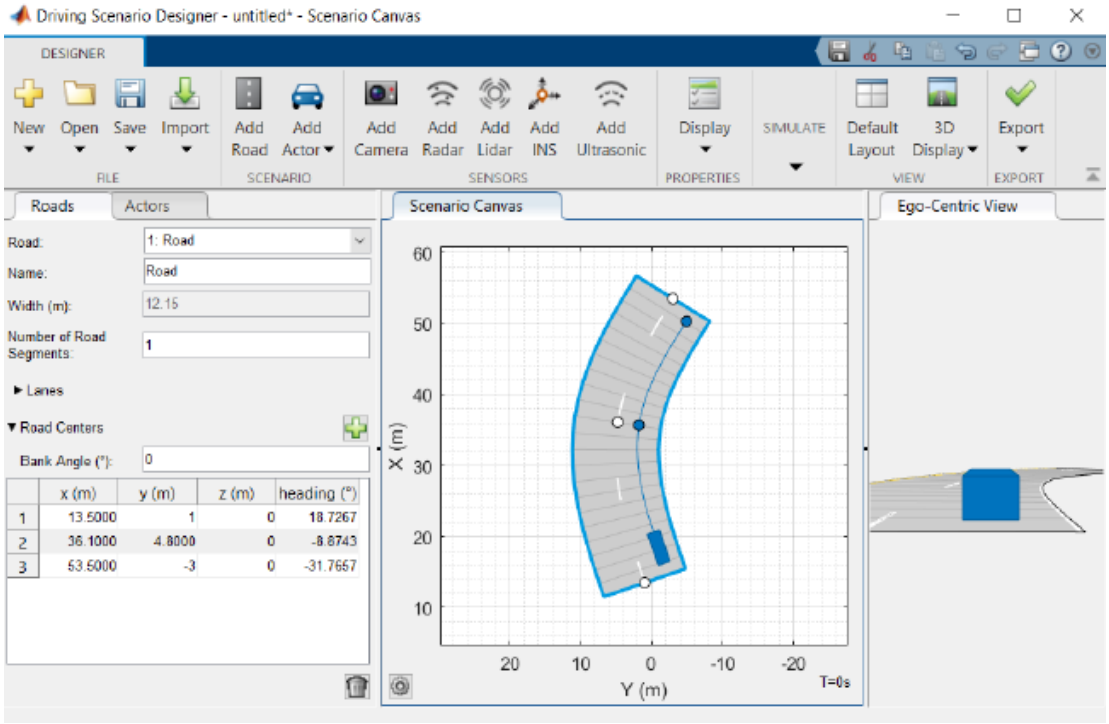

Figure 3.25: Driving Scenario Designer Interface.

In Figure 3.25, a simple example of the creation of a curvilinear trajectory is presented, with a single vehicle (the ego vehicle) traveling along one of the two lanes. The essential elements for the creation of a scenario intended for this type of use are the road and the actors, as well as cars, trucks, pedestrians, bicycles, barriers, guardrails and others.

The first step to create a scenario, even a simple one like the one shown in the Figure, is the creation of the road surface. Through the Add Road tab it is made possible to access the configuration sheet of the road element, visible on the left. The application allows specifying, among various parameters, the number of segments to create, the number of lanes, their width and the inclination angle of the surface. The primary input is the list of road centers, the points on the centerline of the road surface; these can be inserted manually via a table in the configuration sheet or via a drag-and-drop interface. This last methodology allows to position the points graphically in the Scenario Canvas. After positioning them, the points can be dragged to other positions, modifying the centerline of the roadway. The road centers table is automatically filled in, but can be manually modified to correct any inaccuracies.

Once the road centers have been defined, the software automatically generates the centerline and the road surface with the specified number of lanes, visible in the Scenario Canvas.

The process of inserting actors follows a procedure similar to that for creating the road. If a single actor is inserted, as in the example in Figure 3.25, this is identified as the ego vehicle; otherwise, it can be specified which of the actors is the ego vehicle. Using the appropriate tab, it is possible to change the vehicle parameters such as size, type, type of 3D representation, etc. The main advantage is the ability to define a trajectory for a vehicle by inserting points in succession, as for the center line. The points can be inserted via table or graphically, and are automatically interpolated. The vehicle speed can be set manually for each point, allowing the user to calibrate accelerations and decelerations.

In the scenarios created to test the automatic overtaking system, the trajectory of the ego vehicle overlaps at some points with that of another vehicle. This is intentional, since the assistance system must recognize this overlap and correct the trajectory of the ego vehicle or change its speed. In the Driving Scenario Designer simulation, however, this does not represent an error, since the bodies are interpenetrable.

3.3 Planner Strategy Analysis Signals

It is particularly useful to analyze the logic of the system regarding the safety assessments of the ego vehicle with respect to the other vehicles present in the scenario performed for each time step and for each vehicle identified as MIO vehicle.

These signals are those that the planner evaluates instant by instant and their comparison with threshold values leads the system to choose the best solution in terms of driving mode adopted; that is, the CC, LCF and LC modes.

The output of this logical process informs the system about the safety of each MIO vehicle; where MIO safe (Safe) means a vehicle that respects the limits imposed by the system regarding the time to collision (TTC) and the so-called Safe Ranges, that is, the minimum safety distances to be maintained between the ego vehicle and the other vehicles in the scenario.

In the analyzed model, the possibility of differentiating the limit on the relative distance imposed for the vehicles that follow and that precede the ego vehicle has been implemented. Consequently, a comparison is made between the s coordinates of the two vehicles, with s being a curvilinear coordinate calculated with respect to a Frenet reference system. It should be noted that the trajectories are planned by considering the vehicles as point-like, then making this point coincide with the center of the rear axle of the given vehicle.

The process then begins with the evaluation of whether the distance from the following vehicle is greater than that from the preceding vehicle. This initial evaluation is critical because it establishes the directive according to which the system must calculate the safety distance, weighing whether attention must be directed more towards the preceding or the following vehicle.

If the outcome is affirmative, indicating that the ego vehicle has more space behind it than the preceding one, the system deduces that the safety distance must be configured in relation to the following vehicle. This distance is called the "Rear Safety Gap", the safety margin necessary to prevent collisions in situations of sudden braking or other emergencies.

Alternatively, a negative response ($S_{ego} \leq S_{MIO}$) implies that the autonomous vehicle is closer to the vehicle in front than to the one following. In this situation, it becomes imperative to establish an adequate safety distance with the vehicle in front, referred to as the "Front Safety Gap". This safety margin is vital, as vehicles in front may slow down abruptly or stop, requiring a prompt reaction from the autonomous vehicle to maintain safety.

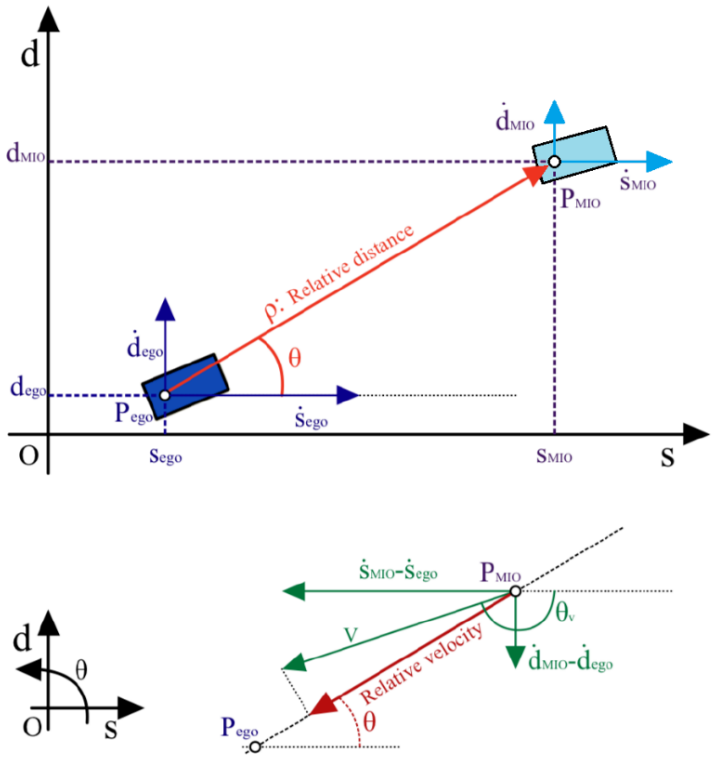

Figure 3.26: Relative distance between ego and MIO vehicles.

The second evaluation is performed by monitoring the time to collision (TTC) between the two vehicles, ego and MIO, given their relative speed and their distance. In Figure 3.26, there is a graphical confirmation of the quantities useful for calculating the TTC. The position of the two vehicles, respectively the ego vehicle, represented in blue, and the MIO vehicle, represented in light blue, is identified by the points Pego and PMIO. Note that the reference system used is a Frenet system, described by the curvilinear coordinate s and the distance in the direction normal to the reference curve d.

The formulas used to calculate the quantities represented are reported below:

$$\text{Relative distance} = \sqrt{\left(s_{MIO} - s_{ego}\right)^2 + \left(d_{MIO} - d_{ego}\right)^2}$$

$$V = \sqrt{\left(\dot{s}_{MIO} - \dot{s}_{ego}\right)^2 + \left(\dot{d}_{MIO} - \dot{d}_{ego}\right)^2}$$

$$\text{Relative velocity} = V\cos(\theta - \theta_V)$$

The angle θ_V is than calculated as follows:

$$\theta_V = \text{atan2}\left(\frac{\dot{d}_{MIO} - \dot{d}_{ego}}{\dot{s}_{MIO} - \dot{s}_{ego}}\right)$$

The relative velocity between the two vehicles is defined as the projection of the velocity vector V along the line joining the centers of the rear axles of the two vehicles. The velocity vector V is in turn defined by the two relative velocity components, respectively along s and along d. Therefore, it is a parameter that expresses the speed with which the two vehicles approach or move away along the direction that joins them. If the value of the Relative velocity parameter is negative, this would indicate an approach of the two vehicles; vice versa, a moving away.

After that, it is possible to calculate the time necessary for the hypothetical collision between the two vehicles, ego and MIO, expressed in seconds in the following way:

$$TTC = \frac{\text{Relative distance}}{\text{Relative velocity}}$$

For values of Relative velocity that are too small, very high TTC values are obtained in modulus. To remedy this problem, a filter has been inserted into the model that, in the case of values lower than the threshold, replaces them with the value of the threshold itself. A threshold of 0.05 m/s has been imposed in the model. Furthermore, examining this formula, it can be stated that a negative value of TTC implies a negative value of Relative velocity between the two vehicles, which indicates a progressive approach between the two, or a reduction of the Relative distance parameter.

A MIO vehicle is defined as Unsafe if it meets at least one of the following conditions:

1. The relative distance between the MIO vehicle and the ego vehicle is less than the Rear Safety Gap/Front Safety Gap safety distance.
2. The TTC assumes a negative value and at the same time is less than the safety limit in modulus.

The safety limits cited and imposed within the model are reported in the following Table 3.3:

Parameter	Limit	Unit
Front Safety Gap	30	m
Rear Safety Gap	10	m
TTC limit	4	s

Table 3.3: Default limits on TTC and safety distance.

Determining whether a MIO vehicle is defined as Safe or Unsafe is necessary to evaluate the lane to which the ego vehicle should be moved or whether to maintain the current lane. The planner evaluates the MIO vehicles present in the ego vehicle lane; if there is a risk of collision (i.e., if an MIO vehicle defined as Unsafe were present in the same lane), the same analysis would also be carried out on the adjacent lanes. If there is no MIO Unsafe vehicle in the adjacent lanes, the planner evaluates, based on a given cost function, whether to maintain its lane (enabling the CC or LCF mode), or whether to enable the lane change strategy. It should be noted that the presence of an Unsafe vehicle in the ego vehicle lane does not necessarily imply the immediate enabling of the lane change mode.

The Safe/Unsafe definitions influence the preferential lane index, but any preferential lane change does not necessarily determine a lane change of the vehicle; this depends on assessments related to feasibility, collision risk and the associated cost parameter.

However, the change of preferential lane necessarily implies the activation of the trajectory planning flag; an input signal is given to the model, so that at that given moment the set of available alternatives is evaluated, each corresponding to a given mode (CC, LCF, LC), and the best one is identified.

Case Studies and Results Obtained

In order to provide a complete overview of the study of an automatic overtaking system, the most critical aspects of the maneuver were analyzed and subsequently tested through appropriate simulations. To carry out the tests and to develop appropriate methodologies capable of limiting the problems highlighted, the tools analyzed in the previous chapters were used, made available for the MATLAB & Simulink simulation environment. What was actually produced was a set of scenarios, each of which was developed with the aim of highlighting a specific aspect or a specific problem of the healthcare system; which then led to theoretical considerations, the development of solutions to be implemented in the simulation environment adopted and the presentation of the same through graphical and numerical results.

4.1 Overview of the Case Studies

The study carried out in chapters 2 and 3 has allowed us to understand the main characteristics required to provide a system dedicated to the design of an overtaking trajectory and to acquire detailed knowledge on the construction of a model intended for this purpose. In this chapter, some of the issues related to the topic have been explored in depth, all analyzed with the aid of the simulator presented in chapter 3.

All the discussion related to the case studies has been produced using the model containing only the trajectory planner, excluding the controller, with the aim of avoiding alterations of the signals produced and, therefore, calibrating the implemented strategies with greater accuracy.

The themes analyzed in this chapter are the following:

- **Case 1 - Single lane change:**

  To clarify the logic with which the individual cases were studied, it was considered appropriate to analyze the behavior of the model applied to a trivial case. Specifically, a scenario was chosen with a single vehicle to overtake, through a single lane change and in the total absence of other actors as shown in Figure 4.1.

  In this way, it was possible to show the functioning of the model for simulations without critical issues and carry out an examination of the type of results obtainable.

Figure 4.1: "Single Lane Change" scenario.

- **Case 2 - Overtaking with Oncoming Vehicle:**

  A further simulation was carried out on a scenario introduced in chapter 2 with the name of "Overtaking with a vehicle in the opposite direction". This is a scenario used in the previous discussion to facilitate the understanding, through practical feedback, of some of the functions or strategies illustrated. Consequently, it was decided to expand the range of simulations reported to include this scenario, both to obtain a complete treatment of the case study and to develop a strategy to deal with the criticality represented by the arrival of vehicles in the opposite direction. In this way it was possible to optimize the assistance system for scenarios not exclusively on motorways but with two-way carriageways. This case is graphically shown in Figure 4.2.

Figure 4.2: “Overtaking with Oncoming Vehicle” scenario.

- **Case 3 - Overtaking a line of vehicles:**

  A rather critical driving situation, both for a driver and for the simulator, is when the vehicles to be overtaken are multiple and very close to each other, as shown in Figure 4.3.

  Specifically, this case study was chosen to analyze the model responses in the case of scenarios in which the positioning of vehicles not initially identified as MIO is not irrelevant for the planner; with the ultimate goal of developing the correct strategies to implement to improve the performance of the assistance system.

Figure 4.3: “Overtaking a Line of Vehicles” scenario.

4.2 Case 1 - Single Lane Change

The scenario of this case study was designed with the aim of providing a simple example from which to obtain and analyze the results of a simulation carried out using the model presented in chapter 3. It consists of an ego vehicle and a target vehicle that travel along the same lane of a straight, two-lane road 250 meters long. This value was chosen to emphasize the overtaking maneuver, ensuring that it is completed within the 250 meters defined by the route.

The target vehicle was positioned so as to precede the ego vehicle, thus assuming the name of lead vehicle. Furthermore, in order to create the need to evaluate whether or not to perform the overtaking maneuver, the two vehicles were characterized by two constant and different speeds, that of the lead vehicle equal to 15 m/s, lower than that of the ego vehicle, corresponding to 20 m/s. Finally, both actors were associated with the same geometry.

In Figure 4.4, two graphs are shown that are useful for understanding the construction of the scenario.

The graph at the top shows the initial state of the two vehicles, whose speeds were represented by two vectors, with different magnitudes but the same direction and orientation.

The graph at the bottom shows the evolution of the scenario in the absence of an intervention of the assistance system, therefore, without a change in the trajectory and/or speed of the ego vehicle. In particular, a collision between the two vehicles is recorded after 6.6 seconds.

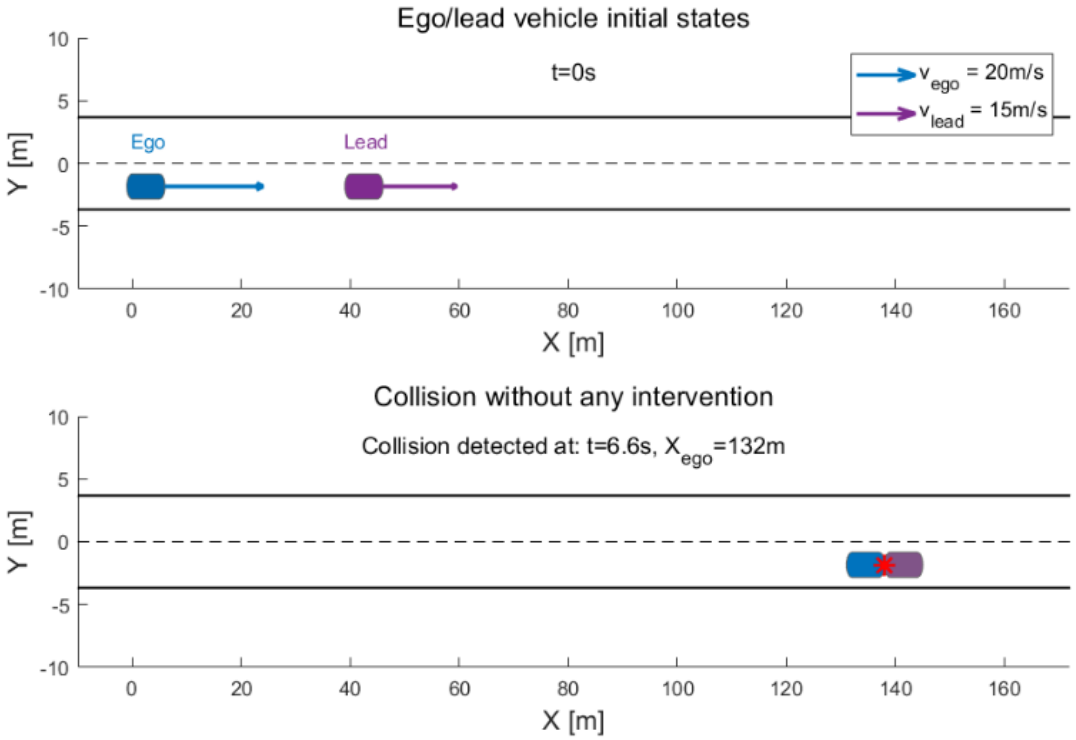

Figure 4.4: Ego/lead vehicle initial states and collision without any intervention in the "Single Lane Change" scenario.

In order to test the automatic overtaking model with this specific scenario, it was necessary to reproduce it using the Driving Scenario Designer application, in order to obtain the correct input. After that, the function produced by the designer, containing the set of actor and road data, was loaded into the model and the simulation was performed. It was successful: the ego vehicle executed the overtaking maneuver, and no errors were recorded during the simulation. It is specified that the simulation was performed maintaining all the default planner configuration parameters described in Table 3.1. The analysis of the results is reported below.

4.2.1 Outcomes of the Simulation - Case 1

As anticipated, no errors were found during the simulation and the strategy used by the planner to avoid the collision with the lead vehicle was to enable the lane change function. Figure 4.5 shows the trajectory of the ego vehicle.

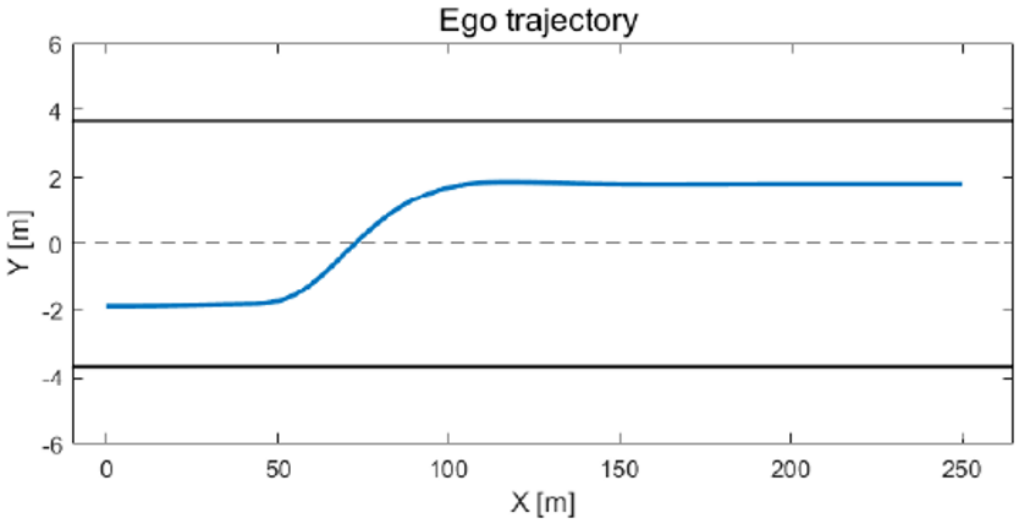

Figure 4.5: Ego vehicle trajectory in the "Single Lane Change" scenario.

It should be noted that in all the graphs reported in this chapter containing trajectories or trajectory segments, it is necessary to pay attention to the difference in scale between the abscissa and ordinate axes. Due to several orders of magnitude of difference between the length and width of the road section examined, the trajectories reported appear to be "squashed"; therefore, it is not necessary to rely on the curvature of the represented trajectories, as in Figure 4.5, or on the shape of the vehicles in the capsule representation, as in Figure 4.7.

In Figure 4.6, in the first graph, there is a representation of the trend of the indices related to the lane of the ego vehicle. Specifically, it can be seen that at the instant t = 2.1 s the planner varies the preferential lane, identifying it with the left lane; a lane that is then reached at the instant t = 3.7 s. The graph below represents the operating mode of the model instant by instant. Following the considerations made regarding the potential collision detected, the model evaluates the optimal terminal state as that which involves a Lane Change (LC). The

preference switches to the Cruise Control (CC) mode only when the vehicle travels along the left lane.

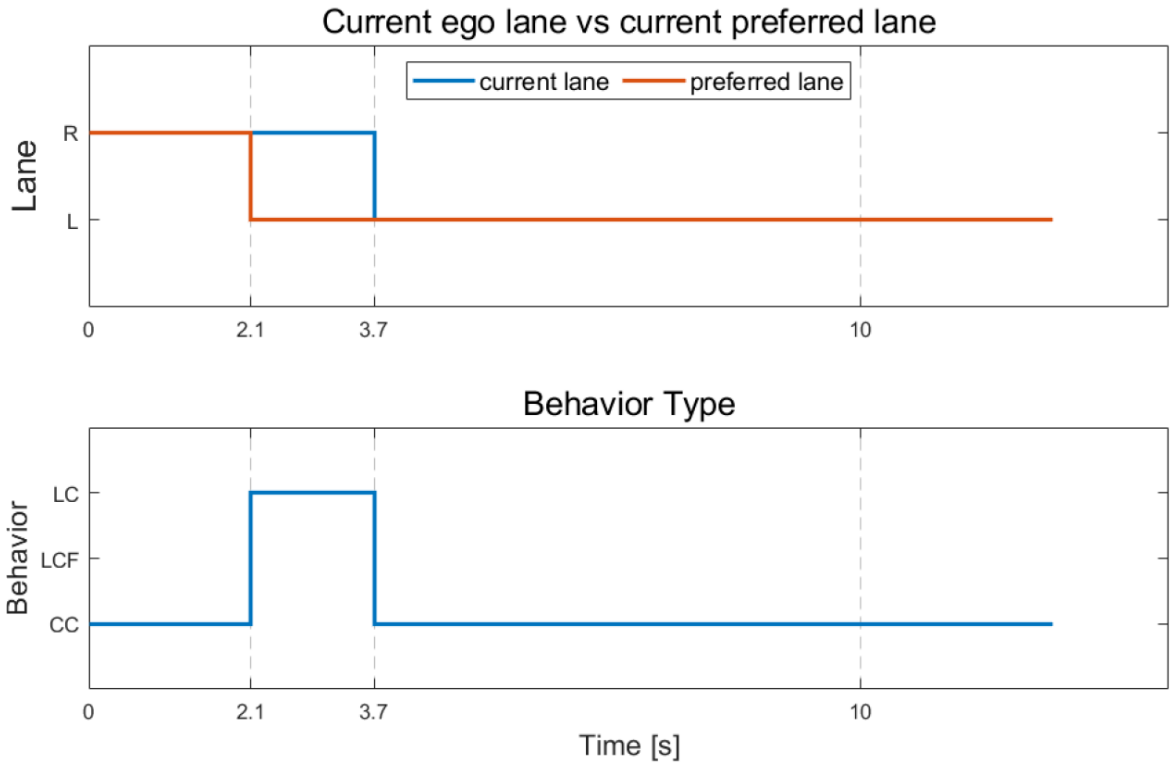

Figure 4.6: Ego vehicle lane in the "Single Lane Change" scenario.

In Figure 4.7, some steps of the trajectory traveled by the ego vehicle have been reported, through a capsule representation of the vehicles. Specifically, the following were represented: the initial instant (t = 0 s), the instant of the start of the lane change (t = 2.1 s), the instant in which the ego vehicle overtakes the lead vehicle (t = 8.1 s) and, finally, the instant of the end of the simulation (t = 12.5 s).

In the representation corresponding to the start of the lane change (t = 2.1 s), there is a segment of the future trajectory of the ego vehicle. This segment was represented because the instant in question is the one in which the change of preferential lane occurs and a consequent planning of the segment of the future trajectory occurs. Going into more detail about the planning mechanisms, the change of preferential lane leads, following an appropriate comparison between all the various alternatives, to the identification of the LC

mode as optimal. This occurs because, by construction of the model, every time the preferred lane changes, the set of terminal states is updated and each of these is evaluated with the aim of identifying a new trajectory segment, if there is a valid one among the available alternatives.

Consequently, the green section represents the trajectory that the ego vehicle will have to follow up to the given time horizon, the instant in which it will be placed in the center of the left lane, since the mode with which this trajectory segment was planned is the LC.

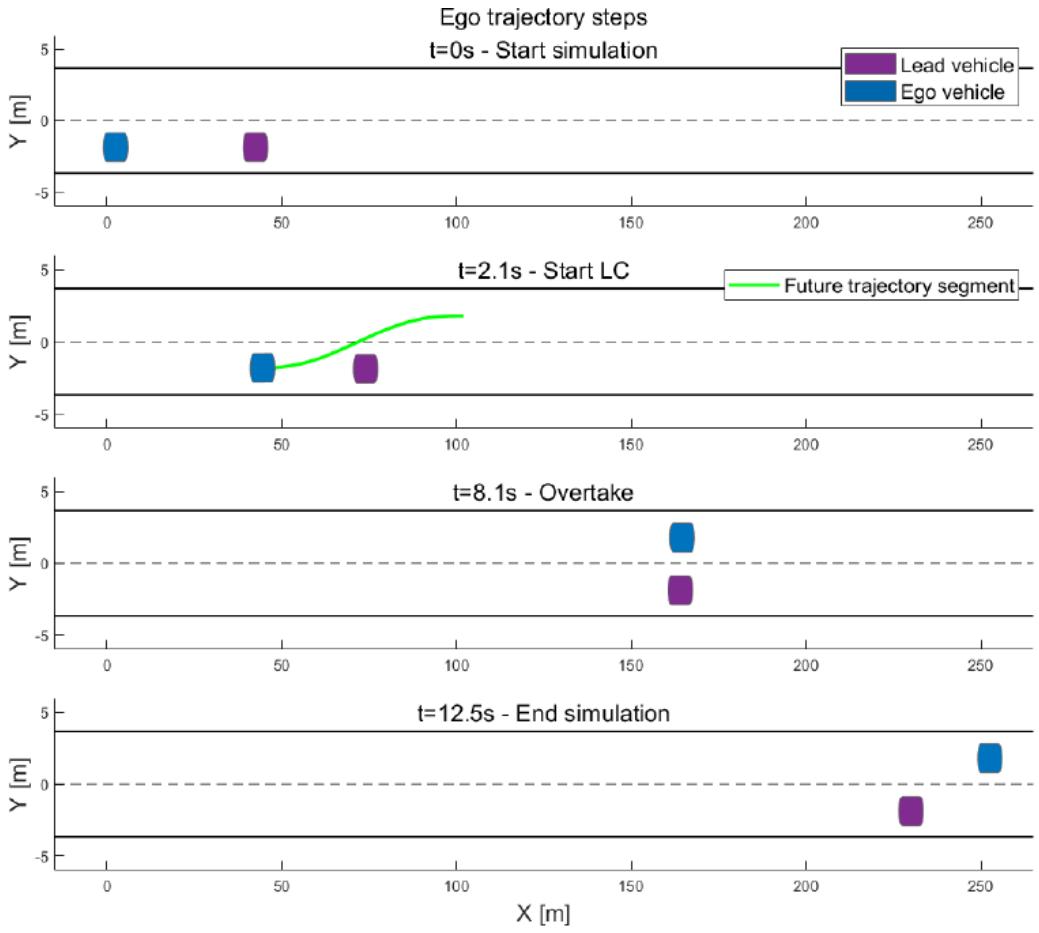

Figure 4.7: Ego vehicle trajectory step in the "Single Lane Change" scenario.

In order to illustrate in detail the mechanisms for selecting the future trajectory segment, in Figure 4.8 the trajectories corresponding to all the terminal states present in the set of alternatives at the instant t = 2.1 s have been graphically reported. In the third graph the future position of the lead vehicle has been reported for each of the three time horizons, so

that the absence of collisions is highlighted by means of a comparison with the corresponding terminal point of one of the represented trajectory segments and calculated with the same time horizon.

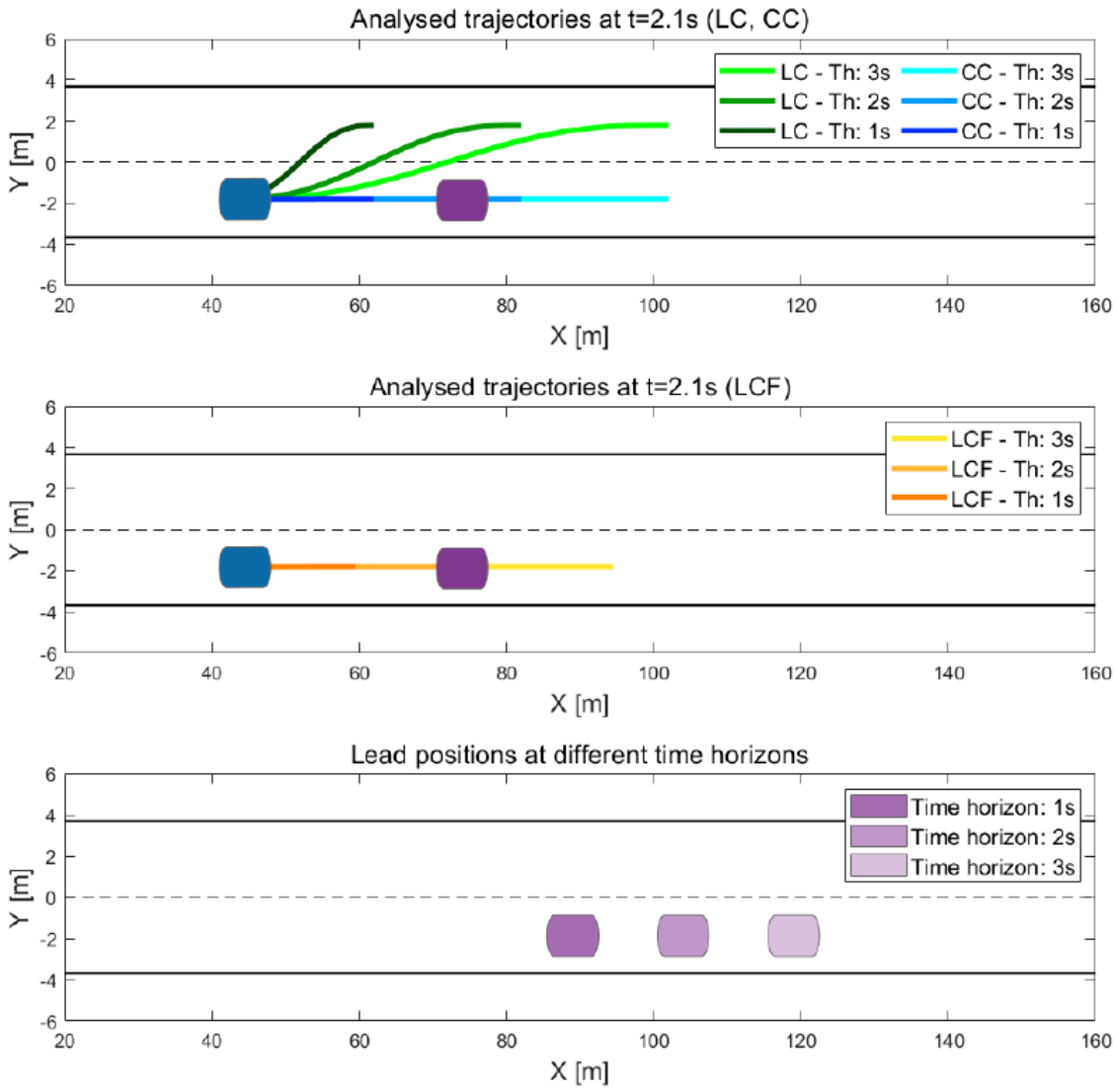

Figure 4.8: Analysis of future trajectories t = 2.1 s in the "Single Lane Change" scenario.

The available alternatives are the following:

- LC mode: for each of the three time horizons (1 s, 2 s and 3 s), they have been represented with different shades of green. Each of which places the ego vehicle in its terminal point in the center of the left lane with a maneuver performed while

maintaining a constant speed.

- CC Mode: for each of the three time horizons (1 s, 2 s and 3 s), they have been represented with different shades of blue. Each of which places the ego vehicle in its terminal point in the center of the right lane with a maneuver performed while maintaining a constant speed.
- LCF Mode: for each of the three time horizons (1 s, 2 s and 3 s), they have been represented with different shades of orange. Each of which places the ego vehicle in its terminal point in the center of the right lane with a maneuver carried out by progressively reducing the speed of the ego vehicle until it equals that of the lead vehicle. Note how, due to the reduction in speed, the terminal point is placed at a lower abscissa than the trajectory segments created with the two different modes but corresponding to the same time horizon.

mode	ṡ [m/s]	d [m]	time hzn.[s]	lat. cost	time cost	speed cost	cost
LC	20	1.80	3	0	-3	0	-3
LC	20	1.80	2	0	-2	0	-2
LC	20	1.80	1	0	-1	0	-1
CC	20	-1.80	3	3.6	-3	0	0.6
CC	20	-1.80	2	3.6	-2	0	1.6
CC	20	-1.80	1	3.6	-1	0	2.6
LCF	15	-1.80	3	3.6	-3	0	5.6
LCF	15	-1.80	2	3.6	-2	0	6.6
LCF	15	-1.80	1	3.6	-1	0	7.6

Table 4.1: Cost parameter set of alternative trajectories (t = 2.1s) in the “Single Lane Change” scenario.

Table 4.1 reports the set of terminal states related to the instant t = 2.1s. Each of them has been described by the mode, the speed in Frenét coordinates, the distance from the center line, the time horizon, the contributions of the cost parameter associated with the lateral deviation, time and speed, and finally the total cost parameter. The states have been ordered based on this last value. As can be seen, the solution identified, the one with the lowest cost parameter, corresponds to the trajectory segment calculated in LC mode and with a time horizon equal to 3s. The cost parameter is calculated with the following equation:

$$\text{cost} = \text{lat. cost} + \text{timecost} + \text{speed cost}$$

Where it is recalled that the individual contributions represent:

- lat. cost: the distance between the preferential lane and the lane in which the ego vehicle is placed at the given time horizon.
- time cost: the time interval that separates the current instant from that in which the vehicle will be at the terminal point of the calculated trajectory segment, i.e. the time horizon. This data is then multiplied by a coefficient equal to -1, so as to favor long-term planning.
- speed cost: the difference in speed between that at the time horizon and that established during the scenario description phase.

Figures 4.9 and 4.10 show the trends of the main quantities that describe the motion of the ego vehicle during the simulation. A rather significant peak in the yaw speed can be noted, however well below the limits on the feasibility of the trajectory defined. The lateral velocity gives a rather small contribution to the absolute speed of the vehicle, which, as can be seen from the third graph, faithfully copies the trend of the longitudinal velocity.

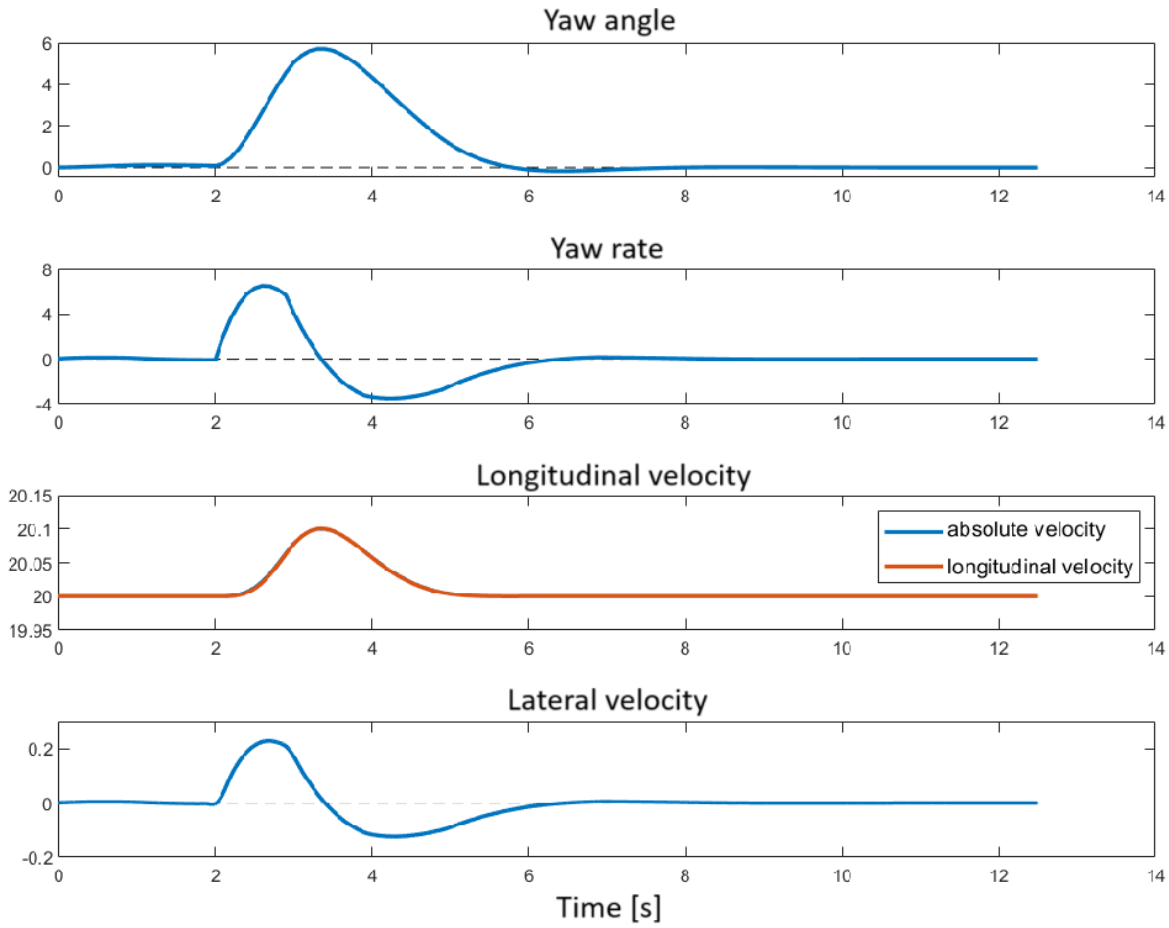

Figure 4.9: Trends of the motion quantities of the ego vehicle in the "Single Lane Change" scenario. Units: [Yaw angle: °], [Yaw rate: °/s], [Longitudinal velocity: m/s], [Lateral velocity: m/s].

As can be seen from the acceleration trend in Figure 4.10, the longitudinal acceleration, assumes very small values of deviation from zero compared to the lateral acceleration, during the lane change maneuver.

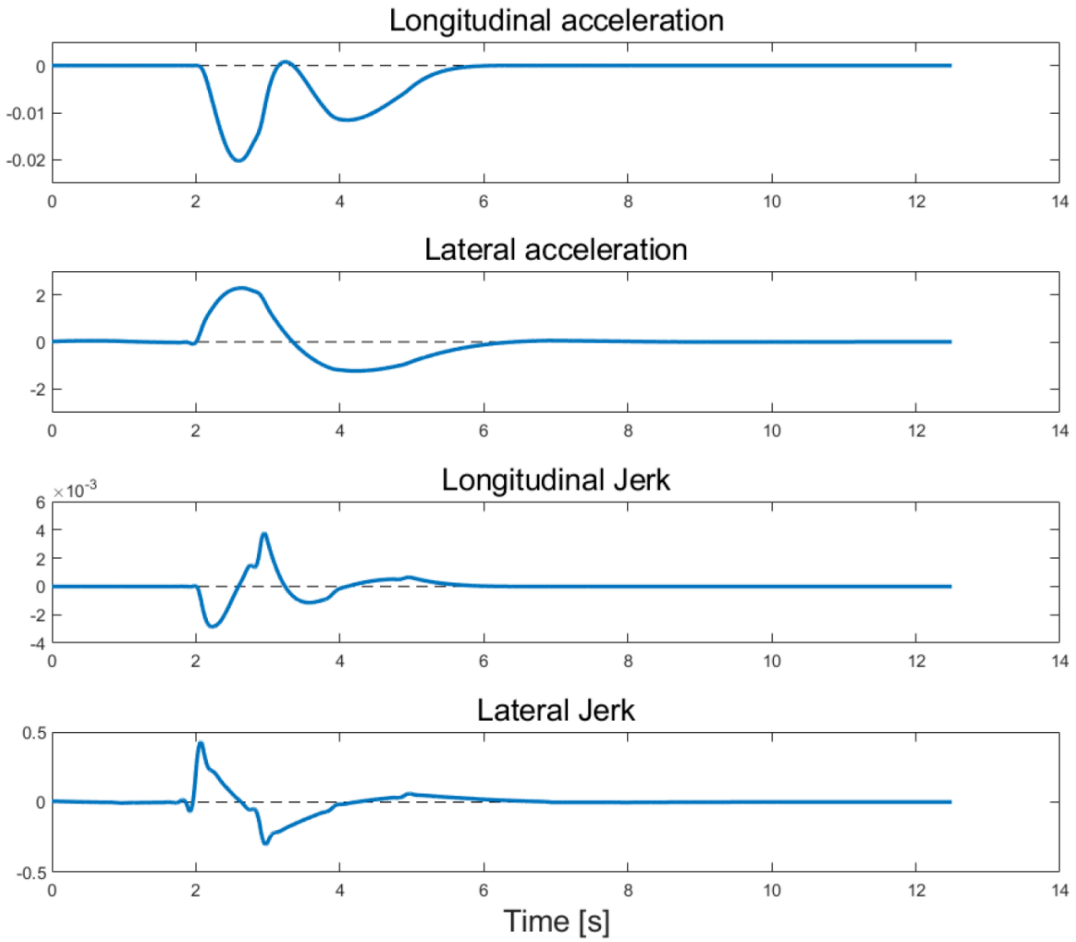

Figure 4.10: Trends of the motion quantities of the ego vehicle (2) in the "Single Lane Change" scenario. Units: [Longitudinal acceleration: m/s²], [Lateral acceleration: m/s²], [Longitudinal jerk: m/s³], [Lateral jerk: m/s³].

As expected, also in the case of the Jerk, the lateral and longitudinal components of the analyzed quantity record values with different orders of magnitude. The contribution of the lateral component of the Jerk is the greater of the two; however, it remains well below the maximum threshold imposed.

The following Table 4.2 shows the peak values, calculated in modulus, of the reported quantities.

Parameter	Max	Limit	Unit
Yaw rate	6.52	20	°/s
Long. Acceleration	$2.02 \cdot 10^{-2}$	5	m/s^2
Lat. Acceleration	2.29	5	m/s^2
Long. Jerk	$3.78 \cdot 10^{-3}$	5	m/s^3
Lat. Jerk	0.43	5	m/s^3

Table 4.2: Peak values of the quantities of the ego vehicle in the “Single Lane Change” scenario.

As initially stated, by comparing each parameter with the relative limit value, no exceedance of the threshold was recorded; consequently, the simulation was successful and the identified trajectory is valid.

After having examined the values of the quantities that characterize the motion of the vehicle, it is particularly interesting, for the purpose of a greater understanding of the logic of a driver assistance system such as the one under examination, to analyze the signals that are evaluated by the system to record the need to perform a lane change.

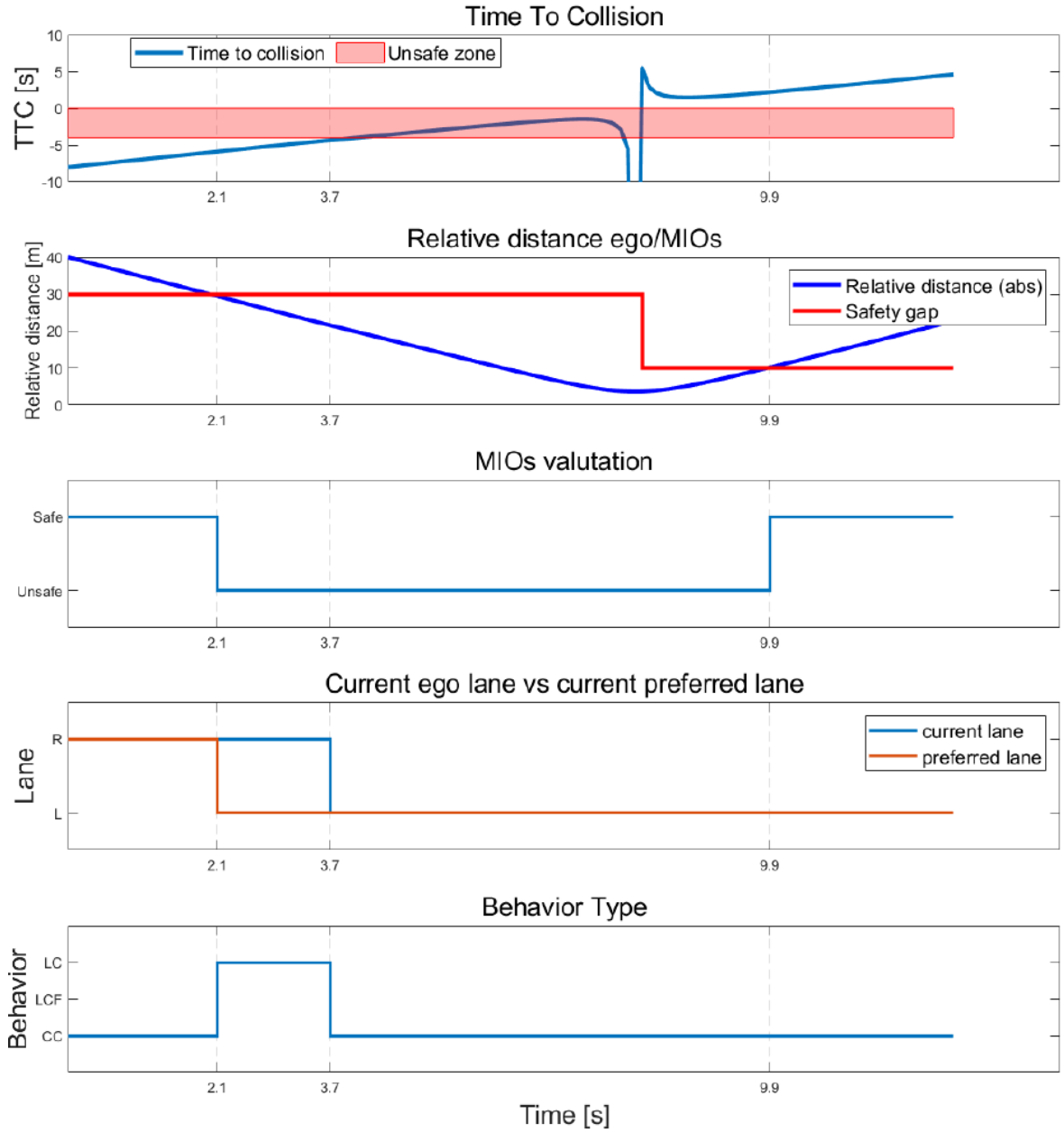

Figure 4.11: MIO vehicle evaluation in the "Single Lane Change" scenario.

Examining the graphs shown in Figure 4.11 it emerges that:

- The MIO vehicle is recognized as Unsafe as soon as the threshold previously defined as safety distance is exceeded, indicated in the second graph as Safety gap and represented in red. As can be seen, the red line assumes two different values; this is due to the different limit between the vehicles preceding the ego vehicle and those

following it. The red line was built to represent the correct limit, instant by instant, taking into account the relative position between the ego vehicle and the MIO vehicle. The logic for assigning the correct limit has been schematized in the flow chart in Figure 3.64. The values used as a limit are equal to the default ones, defined in Table 3.3.

- At the same time that the relative distance value between the two vehicles in the scenario exceeds the threshold value, shown in red, the MIO vehicle is declared Unsafe, as shown in the third graph (MIOs evaluation).
- Following this evaluation of the MIO vehicle, also called lead vehicle or Obstacle 1 vehicle, the planner varies the preferred lane index of the ego vehicle. The related signal is shown in the fourth graph.
- By modifying the preferred lane index, the planner modifies the behavior of the ego vehicle, where behavior change means switching from cruise control (CC) mode to Lane change (LC) mode, as shown in Figure 4.6. This is followed by the execution of the lane change maneuver as can be seen in the graph showing the trajectory in Figure 4.5.
- At this point the ego vehicle is in the left lane following the lane change maneuver. This was possible because there are no MIO vehicles declared Unsafe in the left lane. This is because, by definition of the scenario, there is only one other vehicle besides the ego vehicle, and it maintains the right lane for the entire duration of the simulation.
- It can be seen that, in the graph above showing the trend of the TTC, as long as the two vehicles are in the same lane, the imposed threshold is never exceeded, the numerical value of which is equal to the one assigned by default by the model and reported in Table 3.8. This is exceeded only when the two vehicles travel in different lanes.
- Analyzing the graph, it can be seen that to identify the MIO vehicle as Unsafe by evaluating the TTC, the value of this parameter must reside within the area

represented in red. To declare a MIO vehicle Unsafe, the TTC must be lower in modulus than the threshold and also negative; since, being negative, it implies the presence of a progressive approach of the two vehicles, essential to obtain a collision.

- It is noted that when the ego vehicle has passed the MIO vehicle, an instant that graphically coincides with the change of threshold on the distance shown in red in the second graph for the reasons explained previously, both the relative distance and the TTC assume an increasing trend.
- Finally, examining the third graph, it emerges that in the terminal instants of the simulation, the MIO vehicle assumes the Safe status again, since both parameters, relative distance and TTC, respect the imposed conditions. This does not interest the planner while maintaining the CC mode, since the vehicles travel in different lanes, but it could be useful information if a lane return strategy after having performed the overtaking maneuver were implemented within the model.

Comparison between the different CC/LCF/LC modes

To highlight the characteristics of the assistance system, a comparison was made between the results produced by simulations carried out with the same scenario, progressively eliminating the possibility of enabling the implemented modes. Three different cases were examined: the first allows the assistance system to be able to freely modify the vehicle behavior between the three modes; the second inhibits the activation of the lane change; and the last, instead, allows the ego vehicle to travel its trajectory exclusively in CC mode, i.e. the intervention of the assistance system is completely prevented.

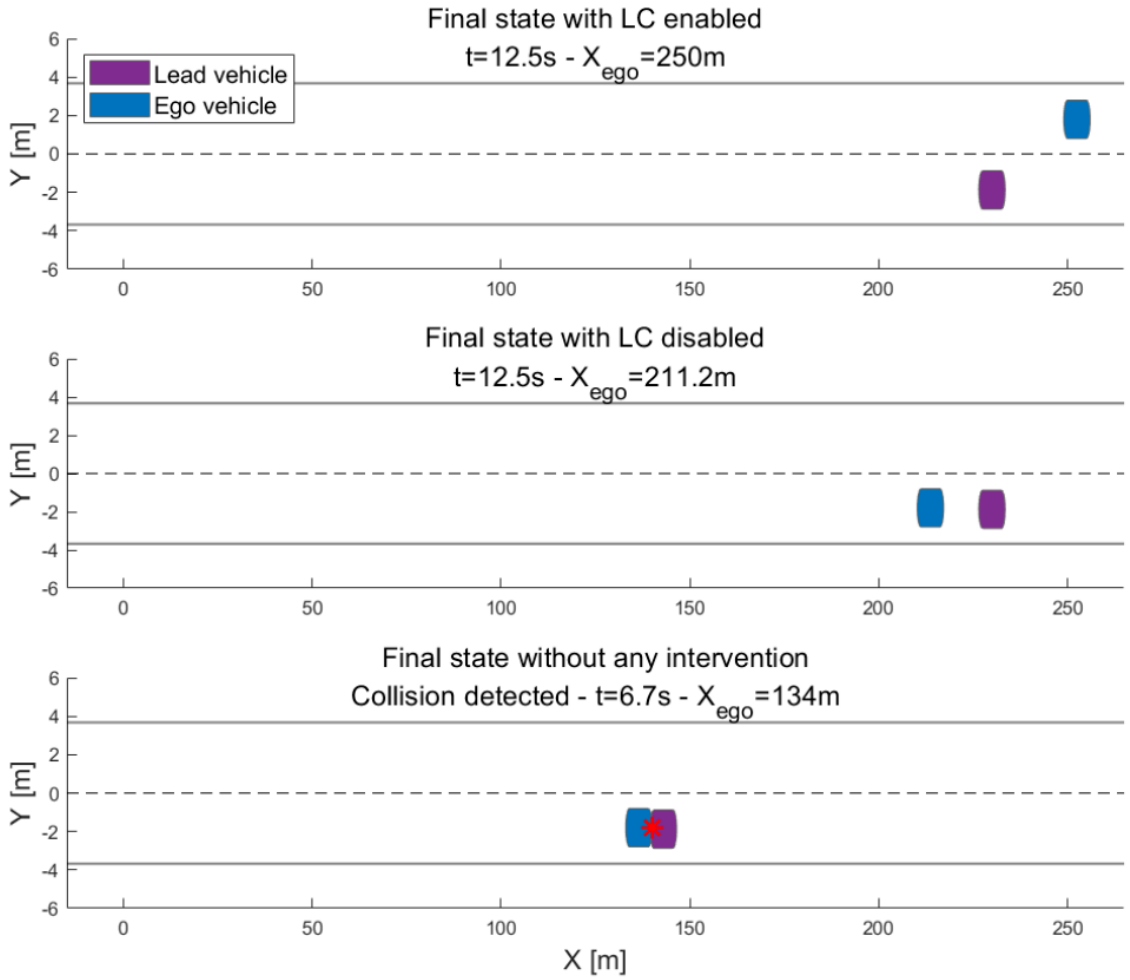

Figure 4.12: Comparison between terminal states in the “Single Lane Change” scenario.

In Figure 4.12 the scenario is shown at the final instant of the simulation for each of the three cases analyzed. From this comparison it emerges that:

- The first graph shows the position of the vehicles in the case in which all modes can be enabled; it has been extensively analyzed previously in this paragraph. The ego vehicle travels a greater distance than the other two cases and no anomalies or collisions are recorded.
- The second graph shows the position of the vehicles in the case in which a lane change is not allowed. Note that the position of the lead vehicle does not vary, since it is defined with the scenario, unlike that of the ego vehicle; which, due to the intervention of the assistance system that enables the LCF mode, reduces the speed

of the ego vehicle until it equals that of the lead vehicle.

- The third graph is relative to the case in which the ego vehicle is allowed to proceed in its motion as described in the scenario definition phase; that is, any type of intervention by the assistance system is inhibited. The simulation is not completed (up to t = 12.5s), as a collision occurs between the two vehicles.

Figure 4.13 shows the time trend of the planner operating mode for each of the three cases analyzed. The comments relating to the first case have already been made in the previous discussion. As regards the second case, note that the LCF mode is enabled at a time after t = 2.1s; where it is recalled that t = 2.1s is the time when the distance between the two vehicles is less than the Front safety gap threshold. This occurs because, unlike the first case, at this step the CC mode is the one with the lowest cost parameter. As evidence of what has been said, the data reported in Table 4.1 can be analyzed, which are valid for all three cases, as up to the time t = 2.1s the ego vehicle behaves in a similar way in the three cases.

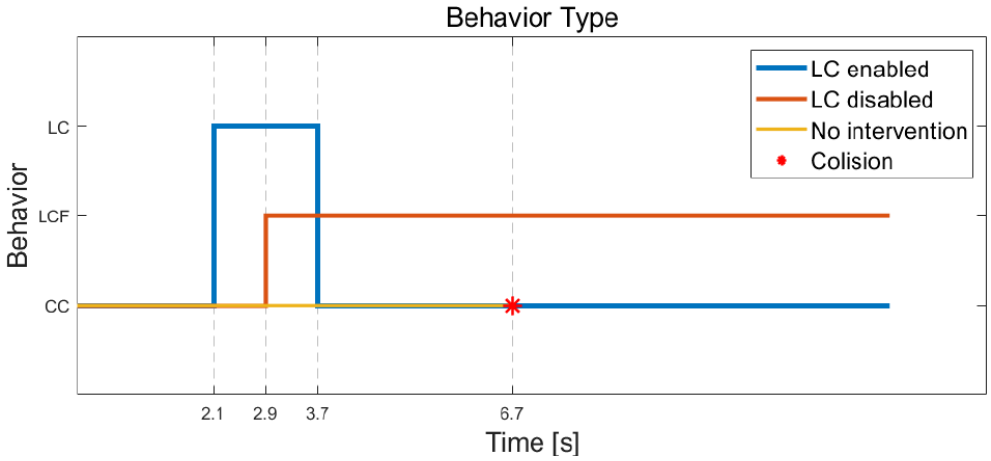

Figure 4.13: Mode comparison in the "Single Lane Change" scenario.

The following Table 4.3 shows the comparison between the peak values of the quantities that describe the motion of the ego vehicle for the two cases that did not encounter collisions.

Parameter	LC disabled	LC enabled	Unit
Max yaw rate	0.00	6.52	°/s
Max long. acceleration	$9.43 \cdot 10^{-2}$	$2.03 \cdot 10^{-2}$	m/s^2
Max lat. acceleration	0.00	2.29	m/s^2
Max long. jerk	$8.98 \cdot 10^{-3}$	$3.78 \cdot 10^{-3}$	m/s^3
Max lat. jerk	0.00	0.43	m/s^3

Table 4.3: Comparison of peak values in the " Single Lane Change " scenario.

It can be easily noted that the configuration that inhibits the lane change leads to a worse performance in terms of distance traveled to the advantage of comfort; since the benefits in terms of angular velocity, lateral acceleration and jerk are not compensated by the worsening of the longitudinal components of acceleration and jerk. However, the lane change allows to cover a greater distance while remaining below the imposed limit thresholds.

4.3 Case 2 - Overtaking with Oncoming Vehicle

The scenario called "Overtaking with Oncoming Vehicle" was introduced and analyzed in the previous chapters with the aim of providing a case study on which to test the introduced functions and algorithms. The road in question, with a straight configuration, is 300 meters long and has two lanes. This value was chosen to emphasize the overtaking maneuver, ensuring that it is completed within the 300 meters defined by the route.

In this scenario, in addition to the ego vehicle, there are two target vehicles. The first, identified as obstacle vehicle 1 or lead vehicle, travels along the road at a constant speed along the same lane as the ego vehicle, as in Case 1. The second vehicle, on the other hand, travels along the road, also at a constant speed, but along the lane with the opposite direction of travel and has been named obstacle vehicle 2 or oncoming vehicle. This name, defined at the initial moment, will be maintained throughout the simulation and for the analysis of the results.

ego vehicle lane	2
lead vehicle lane	2
oncoming vehicle lane	1
$s_{lead} - s_{ego}$	40 m
$s_{oncoming} - s_{ego}$	200 m
v_{ego}	20 m/s
v_{lead}	15 m/s
$v_{oncoming}$	10 m/s

Table 4.4: Vehicle data of the "Overtaking with Oncoming Vehicle" scenario.

Examining the data reported in Table 4.4 it is necessary to specify that, assuming the point of view of the ego vehicle, the lane with index 2 is the right one, where the lead vehicle passes and initially also the ego vehicle; instead, the lane with opposite direction of travel,

the lane belonging to the obstacle vehicle 2, has been identified with index 1. Furthermore; s_{ego}, $s_{oncoming}$ and s_{lead} represent the curvilinear coordinate of the vehicles at the initial instant. Where curvilinear coordinate means the length of the arc calculated with respect to a reference curve established for the reference system adopted; in this case, the reference curve considered was the one representing the center of the roadway. Expressing everything in global coordinates [X,Y] in the XY plane, the reference curve begins at the point [0, 0] and ends at the point [300, 0].

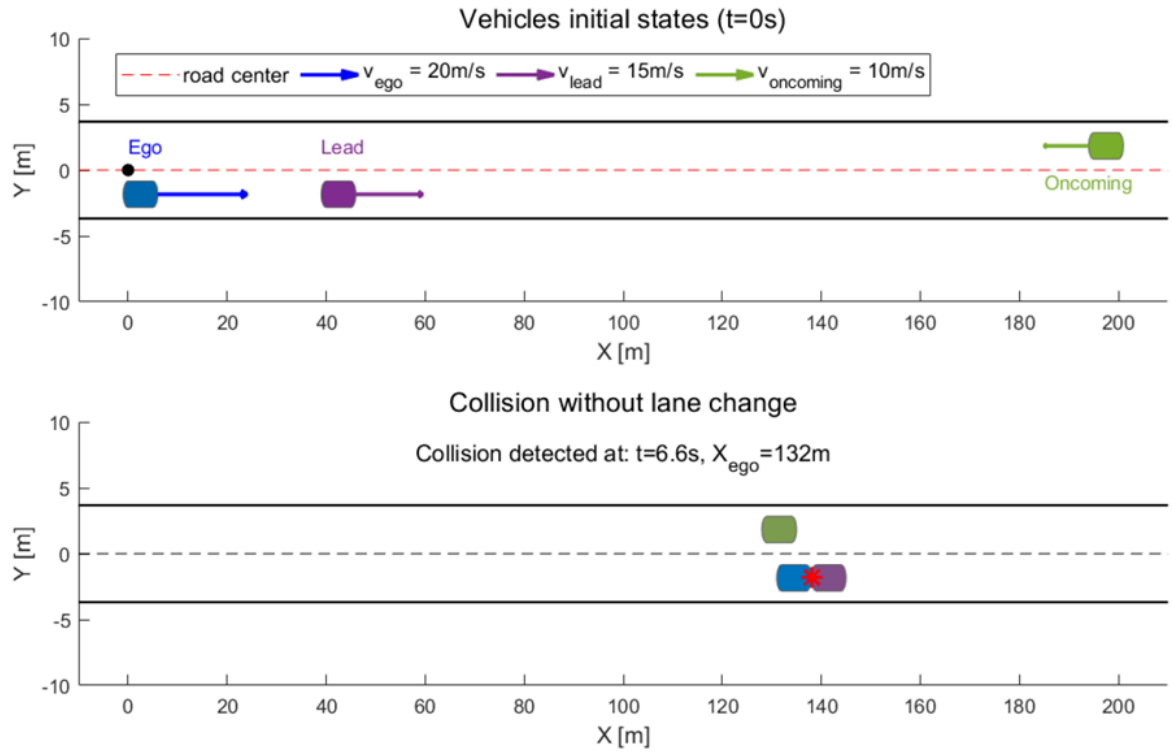

Figure 4.14: Initial states t = 0 s of the "Overtaking with vehicle in opposite direction" scenario.

Figure 4.14 shows a graphic representation of the scenario at the initial instant. The limits of the two lanes have been represented with black continuous lines; the reference curve has been shown as a dashed red line, whose waypoints have been indicated by black dots.

The graph below shows a representation of the collision that would occur in the absence of intervention on the trajectory of the ego vehicle and/or on its speed.

This scenario was designed with the intent of reproducing a highly critical situation to perform an overtaking maneuver. The criticality is inherent in the arrival of a second obstacle vehicle (the oncoming vehicle), which travels in the lane that the system would use as an overtaking lane but with opposite direction of travel.

The scenario was then reproduced within the Driving Scenario Designer application, in order to obtain the correct input to provide to the model. The simulation was completed without recording any collisions and the planner considered that the best alternative for the ego vehicle, in relation to the inserted scenario, was to overtake the lead vehicle. However, the identified trajectory does not respect some of the imposed limits; consequently, it cannot be considered a valid solution. The simulation results were examined in the following paragraph; after these analyses, the critical issues that emerged were analyzed, finally proposing a solution capable of identifying a valid trajectory.

4.3.1 Outcomes of the Simulation - Case 2

As anticipated, the simulation was completed in the total absence of collisions between the ego vehicle and the other two vehicles in the scenario. Following the detection of an imminent collision, in the event of maintaining the trajectory described within the scenario, the planner had to intervene on the states of the ego vehicle by modifying its trajectory. The trajectory traveled by the vehicle during the simulation is shown in Figure 4.15.

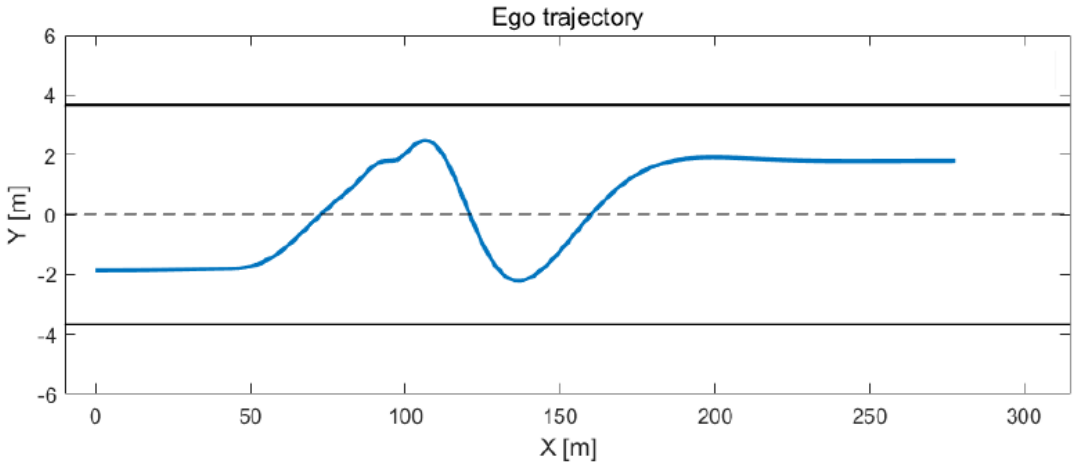

Figure 4.15: Ego vehicle trajectory in the "Overtaking with vehicle in opposite direction" scenario.

It is immediately noticeable that the identified trajectory is not optimal, as it requires a triple lane change to be achieved; which is in no way necessary, since the ego vehicle moves into the overtaking lane and then returns to the previous lane, without actually having performed the overtaking maneuver to the detriment of the lead vehicle.

For a better understanding of the simulation results, the scenario representations by means of capsules at the most relevant instants have been superimposed in Figure 4.16. The trajectory segments produced by the planner have been represented in green.

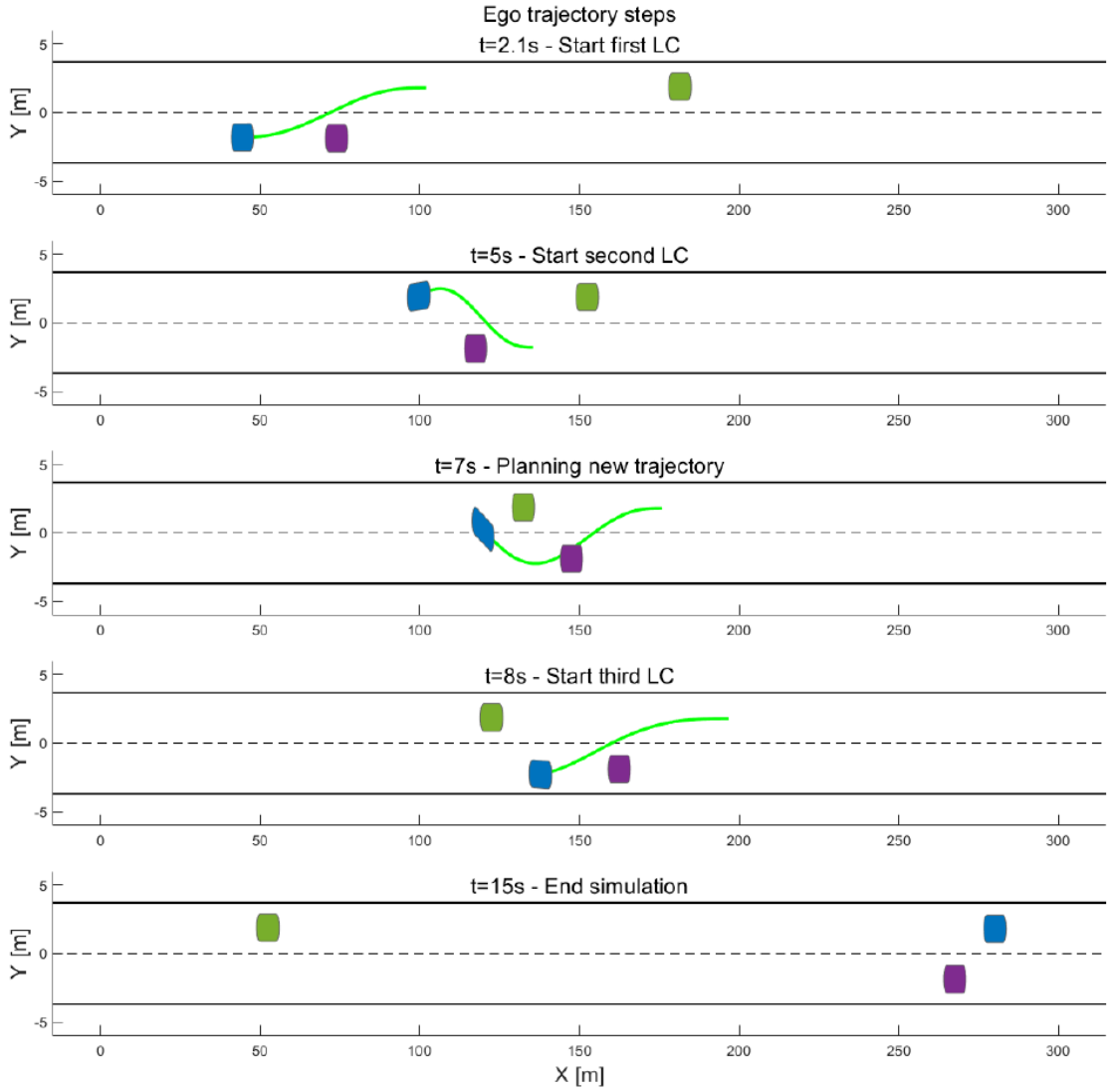

Figure 4.16: Ego vehicle trajectory frame in the "Overtaking with Oncoming Vehicle" scenario.

Furthermore, this trajectory is not actually drivable by the vehicle because the imposed feasibility limits are exceeded; a numerical and graphical confirmation of the quantities that characterize the vehicle motion is available in Table 4.5 and in figures 4.17 and 4.18. Specifically, the limits regarding the yaw rate and the lateral acceleration are exceeded at two different moments of the simulation, around 4.9 s and around 7.7 s.

Parameter	Max	Limit	Unit
Yaw rate	26.99	20	°/s
Long. Acceleration	0.83	5	m/s^2
Lat. Acceleration	7.84	5	m/s^2
Long. Jerk	0.14	5	m/s^3
Lat. Jerk	2.53	5	m/s^3

Table 4.5: Peak values of vehicle and ego quantities of the "Overtaking with Oncoming Vehicle" scenario.

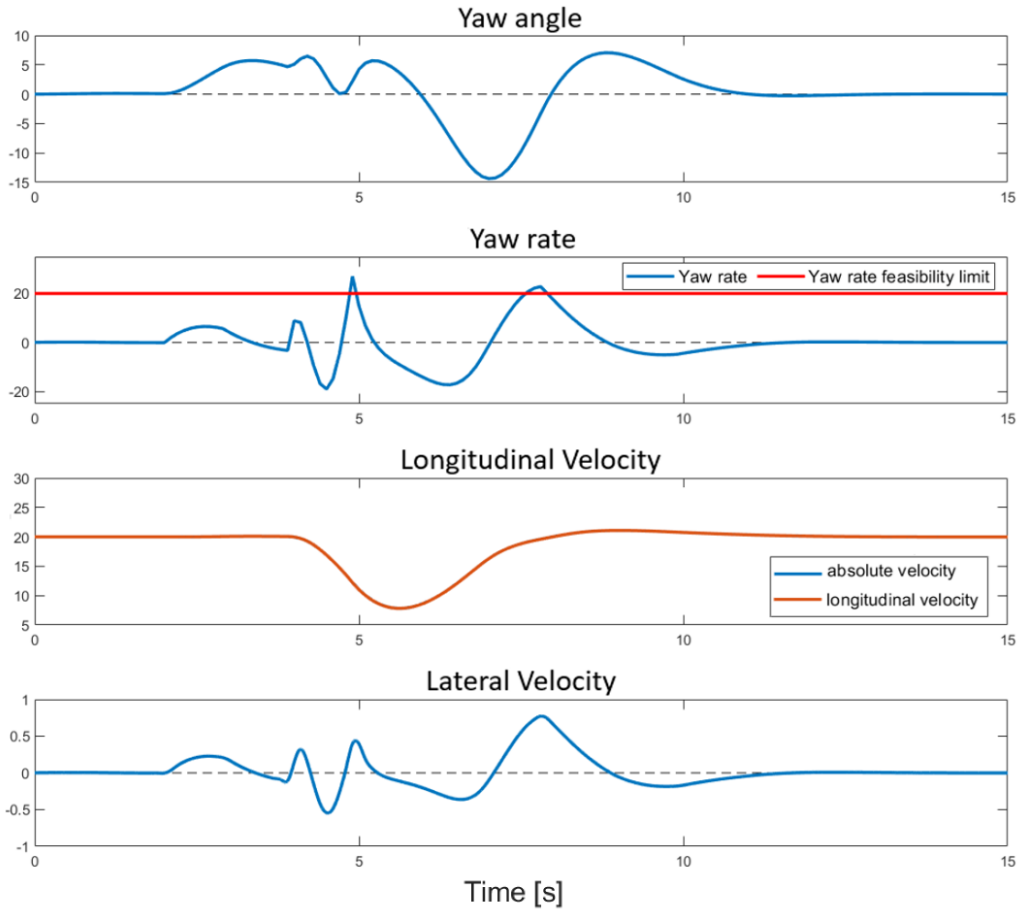

Figure 4.17: Trends of the motion quantities of the ego vehicle in the “Overtaking with Oncoming Vehicle" scenario. Units: [Yaw angle: °], [Yaw rate: °/s], [Longitudinal velocity: m/s], [Lateral velocity: m/s]

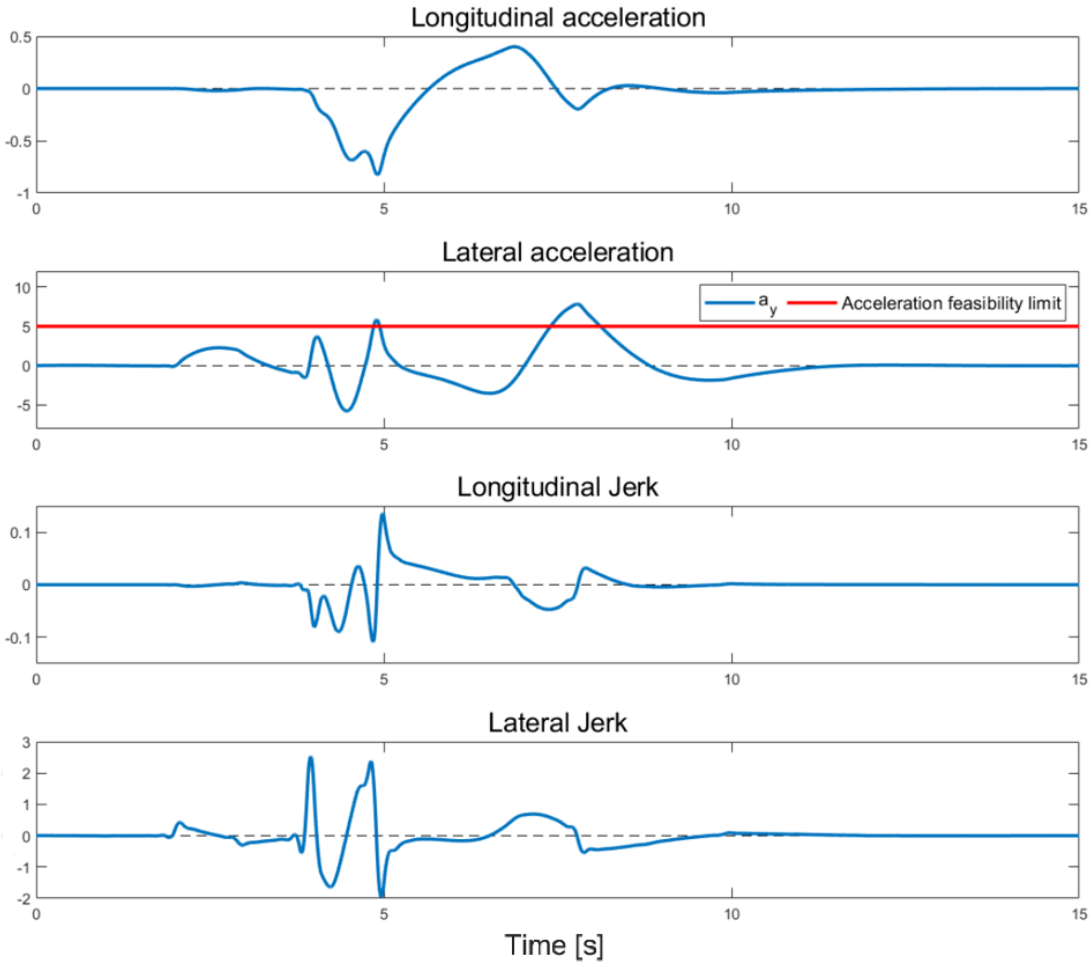

Figure 4.18: Trends of the motion quantities of the ego vehicle (2) in the “Overtaking with Oncoming Vehicle" scenario. Units: [Longitudinal acceleration: m/s²], [Lateral acceleration: m/s²], [Longitudinal jerk: m/s³], [Lateral jerk: m/s³].

Analyzing the reported results in more detail, it emerges that the two areas of the trajectory that do not respect the limits imposed on lateral acceleration and yaw velocity correspond to the sudden changes in trajectory necessary to avoid collisions with the other two vehicles present in the scenario. To make the explanation more effective, the "invalid" points according to the imposed feasibility parameters have been graphically reported on the trajectory in Figure 4.19.

In the proximity of the 100 m, it can be seen how the ego vehicle, now having completed the lane change maneuver, having therefore reached the center of the same with zero yaw angle, is forced to perform an emergency maneuver to avoid the collision with the vehicle

previously called oncoming.

During this maneuver, the change in yaw angle is so sudden that it exceeds the imposed limit and weighs on the calculation of the lateral acceleration a_y, also bringing the value of the latter beyond the threshold.

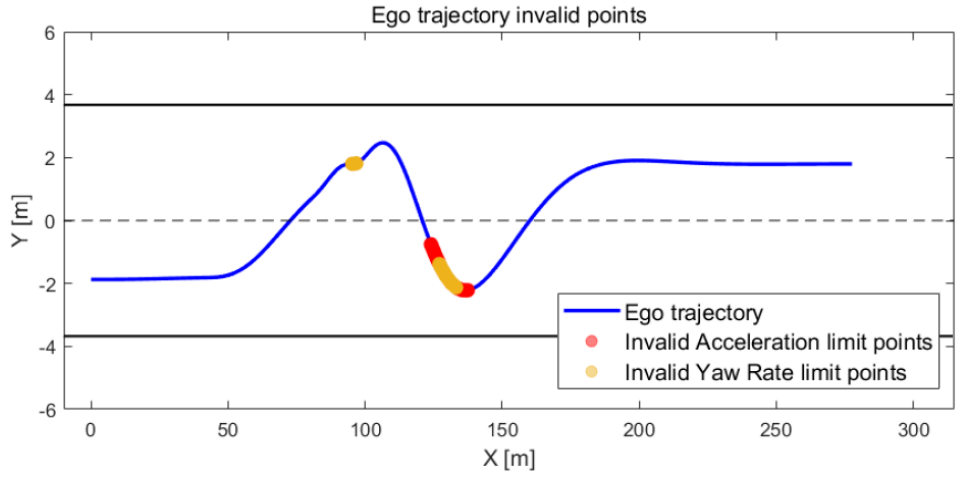

Figure 4.19: Trajectory points invalidating the imposed limits in the "Overtaking with Oncoming Vehicle" scenario.

Looking at the yaw angle and yaw velocity graphs in Figure 4.17, we note that:

- After performing this first evasive maneuver, the ego vehicle heads toward the lane of its own direction of travel, the lane on which the lead vehicle is traveling, reaching a minimum in the yaw angle trend at the instant in which 7.7 s have passed since the start of the simulation, to which a value equal to -14.37° is associated.
- After only 1.1 s, at the simulation instant equal to 8.8 s, the ego vehicle has a yaw angle equal to 7.05°, as well as the maximum trend of the quantity examined.
- This data implies that the vehicle, at the instant equal to 8.8 s, is involved in its third lane change, with the ultimate goal of completing the overtaking to the detriment of the lead vehicle.
- Achieving the goal in the subsequent instants of the simulation, brings to a rapid change of direction leading to invalidate once again the limits regarding the yaw rate and the related lateral acceleration a_y, as depicted in Figure 4.19.

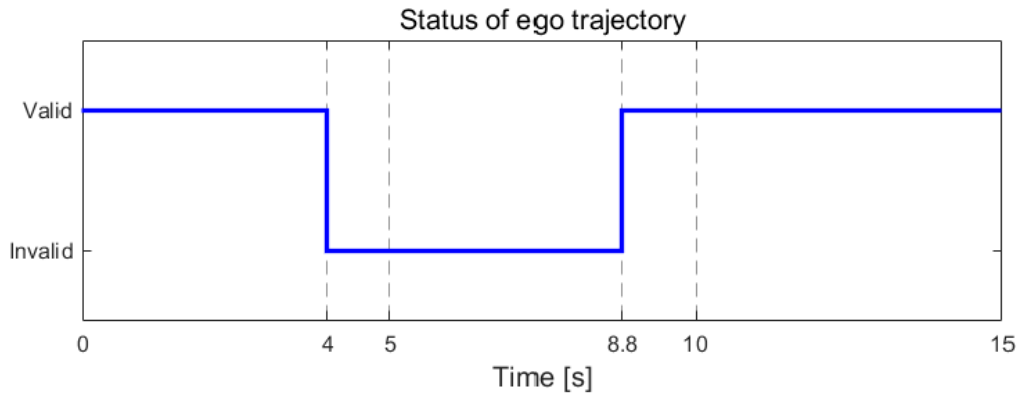

Figure 4.20: Ego vehicle status in the "Overtaking with Oncoming Vehicle" scenario.

Following the considerations introduced, we can comment on the graph in Figure 4.20. It reports in graphic form the signal of the ego vehicle status (Valid/Invalid). To obtain a simulation with a positive outcome, therefore a trajectory that can actually be traveled by the ego vehicle, it is necessary that the reported signal is equal to the abscissa Valid for each instant of time. However, for the reasons explained previously, this is not the case in question. However, it can be noted that the start and end instants of the Invalid status are respectively prior and subsequent to the two time instants obtained previously, equal to 4.9 s and 7.7 s, approximately the actual exceeding of the limits linked to the descriptive quantities of the vehicle motion. This is due to the construction logic of the planner, as it elaborates a trajectory to be followed by the ego vehicle starting from the current instant up to a subsequent time horizon, and then recalculates it if necessary or if this time horizon is reached. Consequently, the evaluation of the set of alternatives is carried out at the time in which the next trajectory section is planned; if a point of the best trajectory identified is invalid, then the vehicle is declared Invalid from the very moment of planning, until a Valid trajectory to follow is identified.

Figure 4.21 shows a graphic representation of the optimal and ideal lane index, as far as the trajectory of the ego vehicle is concerned; the graph below shows the trend of the system operating mode during the simulation.

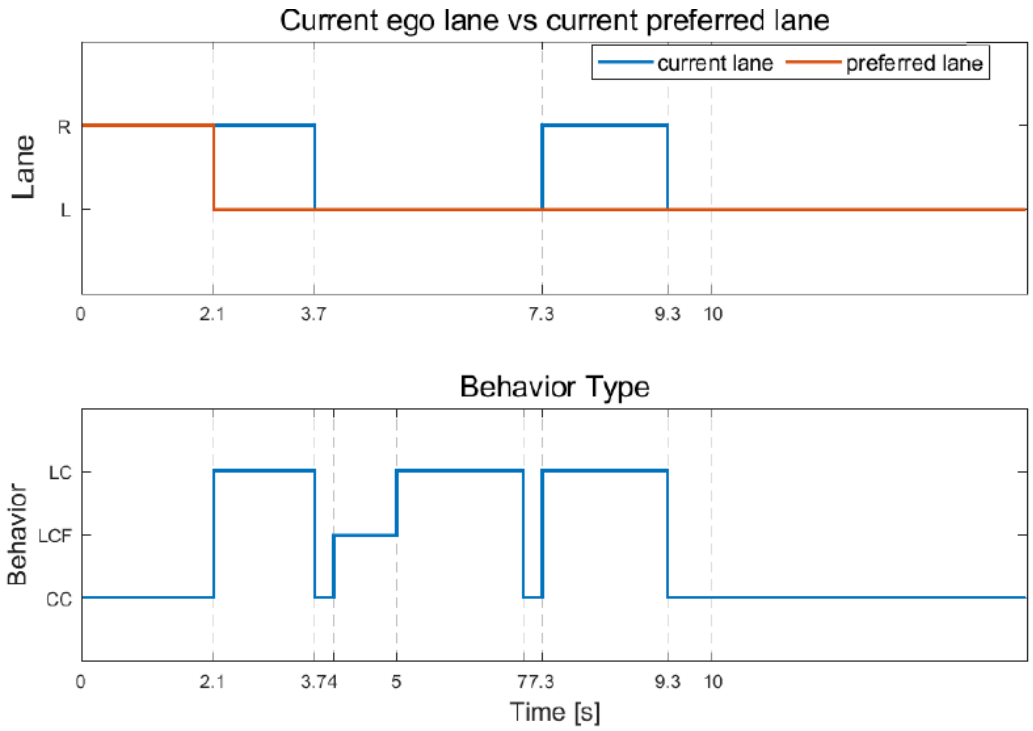

Figure 4.21: Ego vehicle lane in the "Overtaking with Oncoming Vehicle" scenario.

As can be seen from the graph above in Figure 4.21, from the instant equal to 2.1 s the preferred lane is the overtaking lane, the left lane, following the detection of an imminent collision with the lead vehicle in the absence of a variation in the motion of the ego vehicle compared to what is described in the scenario. This is due to two coinciding trajectories between the two vehicles and to a higher speed of the ego vehicle compared to that of the lead vehicle. The preferred lane is kept the same for the rest of the simulation, unlike the current lane, as the ego vehicle is forced to return to the right lane again to avoid a collision with the oncoming vehicle.

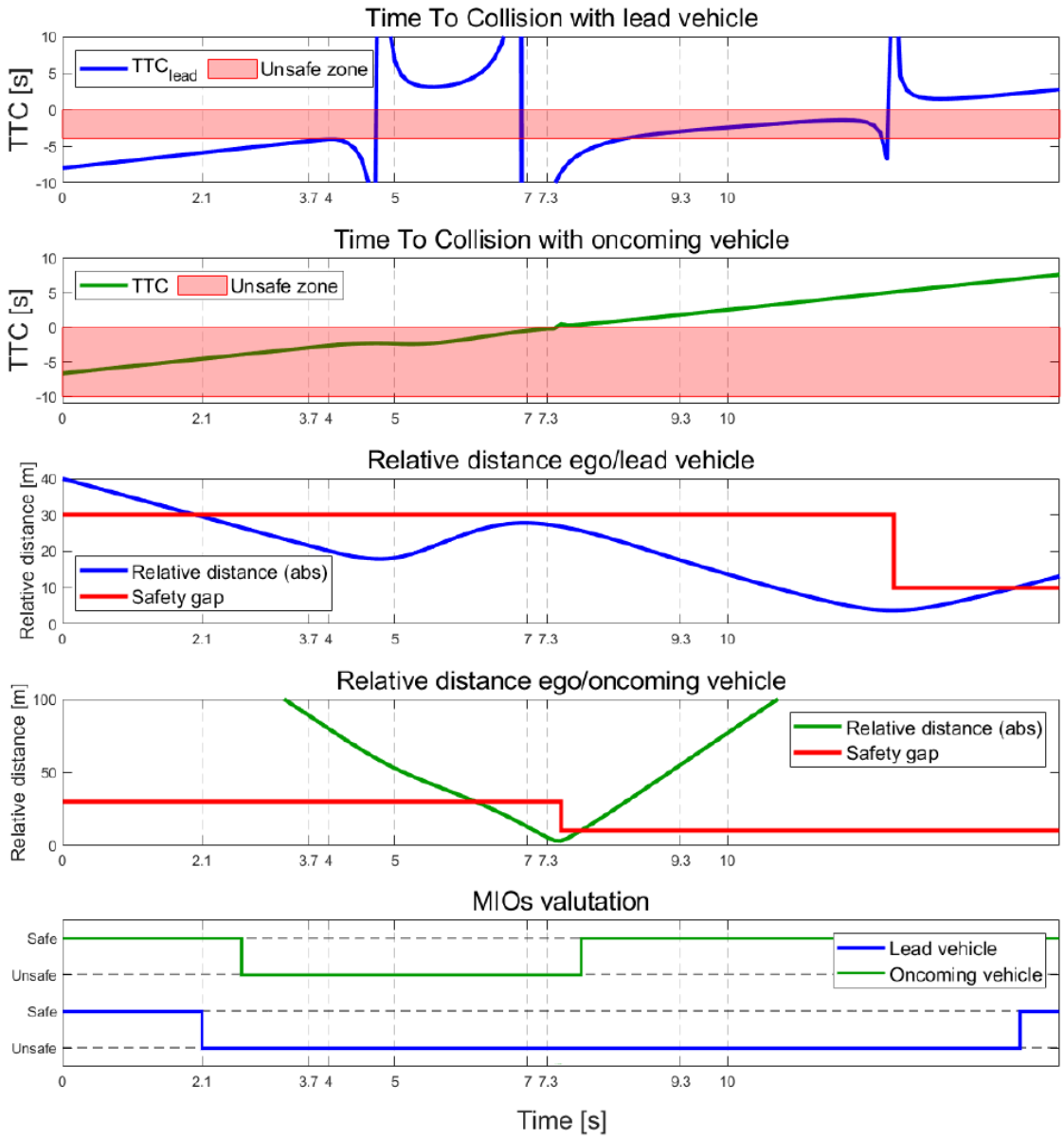

Figure 4.22: MIO vehicle evaluation in the "Overtaking with Oncoming Vehicle" scenario.

The following is an analysis of the trends of the main quantities used by the planner during the simulation phase, reported graphically in Figure 4.22. The analysis will be performed in chronological order, commenting on the most significant moments. In order to make this analysis more understandable, the operating modes of the planner have been reported on the x-axis; the related trend can also be analyzed using the graphic representation in Figure 4.21.

- The first section is the one in CC mode, as the vehicle follows the trajectory described within the scenario.
- After 2.1s, the vehicle switches to LC mode, as the lead vehicle is declared as Unsafe. This happens because the limit about the Relative Distance is exceeded, as can be appreciated in the third graph in Figure 4.22. As can be seen from the last graph, called "MIOs evaluation", the oncoming vehicle, travelling along the overtaking lane, is considered Safe at that moment. The planned trajectory segment can be seen in the first graph in Figure 4.17.
- The overtaking is performed, and once the ego vehicle is in the left lane, the CC function is enabled, useful for overtaking the lead vehicle along the curvilinear reference coordinate s.
- During operation in LC mode, therefore during the lane change manoeuvre, the oncoming vehicle progressively reduces its distance from the ego vehicle; since, as can be seen from the second graph in Figure 4.23, the TTC is negative. The TTC progressively decreases until the threshold, represented in red and with the name of Unsafe zone, is exceeded and the oncoming vehicle is declared Unsafe. This happens at the instant equal to 2.6 s, that is, before the completion of the overtaking maneuver.
- At the moment in which the oncoming vehicle is declared Unsafe, both vehicles, as can be seen from the last graph, are not Safe. The lane change maneuver started at the instant t=2.1s is completed and the CC mode is enabled.
- The one presented in the previous point turns out to be a disabling criticality; since, if the oncoming vehicle had been recognized as unsafe before the start of the lane change maneuver, the latter would not have been carried out and the subsequent evasive maneuvers described previously would not have been necessary.
- At the instant equal to 4 s the LCF function is enabled; however, at the same instant the trajectory is declared Invalid, as can be seen from the graph in Figure 4.20. When the trajectory to follow is evaluated as invalid, this implies that all the terminal states lead to the development of trajectories considered invalid. Consequently, this mode

(LCF) is identified as the one to follow since the CC mode leads first to the collision, since the speed of the ego vehicle is greater than that of the oncoming vehicle. In fact, a progressive reduction in speed can be seen in the graph in Figure 4.18. It should be remembered that in CC mode the ego vehicle adopts a speed equal to that entered during the description of the scenario, 20 m/s in this case; instead, in LCF mode, at the final instant of the planned trajectory it adopts a speed equal to that of the vehicle preceding it along the same lane, in this case the oncoming vehicle, which has a speed equal to 10 m/s and negative according to the given reference system. As anticipated, a lower ego vehicle speed implies a lower Relative Velocity, therefore a higher TTC all other conditions being equal; this is therefore the reason why the LCF mode is enabled.

- As can be seen from the description given in the previous point, the assistance system enables the LCF mode by recognizing the oncoming vehicle, which has the opposite direction of travel to the first, as the vehicle to follow, simply because it precedes the ego vehicle in the same lane. This is incorrect and is a problem solved with the solution adopted later and presented in the next subparagraph.
- At the instant t=5s, the LC mode is switched again to avoid the collision with the oncoming vehicle, until the right lane is reached and the CC mode is activated again. However, it is necessary to underline again that this lane change is performed before having passed the ego vehicle with respect to the curvilinear reference coordinate s; therefore, when the lane re-entry is completed, the lead vehicle continues to precede the ego vehicle, effectively making the maneuvers carried out previously useless. This turns out to be a system problem on which it is necessary to intervene. The planned trajectory segment can be viewed in the second graph in Figure 4.17.
- At the instant t=7s a new trajectory segment is planned in CC mode; this is because, as can be seen from the third graph in Figure 4.16, the first and last points of the fifth-order polynomial are both in the same lane. However, at the instant t=7.3s the vehicle crosses the center line, thus switching to LC mode, as the current lane and the terminal state lane of the segment are different.

- At the instant t=8s the ego vehicle is in the right lane, a new trajectory is planned in LC mode to avoid a collision with the lead vehicle. The planned trajectory segment can be viewed in the fourth graph in Figure 4.16.
- In the subsequent instants of the simulation the overtaking is performed and the CC mode is enabled again. Possible because the oncoming vehicle has passed, therefore no longer approaching and considered Safe. Both vehicles are declared Safe only starting from instant 14.3 s, that is when the ego vehicle overtakes the lead vehicle with respect to the curvilinear reference coordinate s.

4.3.2 System Adaptation to Presented Critical Issues - Case 2

Following the critical issues that emerged from the simulation results, appropriately analyzed in the previous subsection, a strategy was developed that is able to adapt the response of the advanced driver assistance system to scenarios similar to the one described in this case study.

Explanation of the problem

The previous analysis led to the identification of the problem that leads to the impossibility of identifying a valid and optimal trajectory for this case study; which develops with the concomitance of the scenarios described in the following points:

- The advanced driver assistance system with which the ego vehicle is equipped defines the oncoming vehicle as Unsafe after having already performed the first lane change maneuver; that is, only when the ego vehicle is already traveling along the overtaking lane, the left one.
- The limits on the time to collision TTC and on the relative distance Relative Distance, on the basis of which the safety assessment of a given Safe/Unsafe vehicle is carried out, are incompatible with the return to the lane and a concomitant completion of the overtaking maneuver to the detriment of the lead vehicle. The incompatibility arises from the fact that these limits are calibrated for the comparison between the trajectory of the ego vehicle and that of other vehicles traveling on the roadway in the same direction of travel.

The two characteristics presented must coexist in order to incur a criticality that leads to invalid results, such as those in the case in question. This is necessary because overtaking can actually be completed even if a vehicle in the same lane and in the opposite direction of travel were to arrive during the maneuver, provided that the distances are compatible with overtaking the lead vehicle and subsequently returning to the lane.

However, to allow this, the second condition would have to be missing; that is, it would be

necessary to adopt higher limits regarding the TTC and the Relative Distance, in order to communicate to a higher Relative Distance and a TTC the need to modify the trajectory. Remembering that this need arises from the presence of an Unsafe vehicle in the same lane as the ego vehicle and that the vehicles in the scenario are declared Unsafe if they exceed the TTC and Relative Distance limits.

In such way, even if the definition of Unsafe vehicle regarding the oncoming vehicle were to occur already when overtaking had started, as in the case in question, there would be the time and distance necessary to complete the maneuver, as the planner would vary the trajectory of the ego vehicle with sufficient advance notice. However, what may apparently seem like a solution, i.e. modifying the parameters mentioned, does not appear to be an effective path for the reasons explained below.

Adopted solution

To solve the problem presented, the objective was to allow the system to recognize the presence of such a criticality, providing a response that is always solid, i.e. not dependent on the numerical calibration of the configuration parameters on a case-by-case basis. Consequently, a strategy was developed based on the principles set out below.

For a better understanding of what follows, it is recalled that the trajectory to follow until the next time horizon is chosen from a set of alternatives based on a given cost function. Furthermore, this set of alternatives is calculated on the basis of the so-called "terminal states"; that is, starting from the current position of the ego vehicle, the point where it is wanted to be placed at the given time horizon for each specific alternative. Where time horizon means the instant t at which the planned trajectory segment ends; instant t calculated by adding a Δt, expressed in seconds, to the "current time". For each of these terminal states, which differ both in time horizon and in operating mode (CC, LCF or LC), a trajectory segment will be calculated.

Based on what has been said, the strategy has been developed in order to:

1. Recognize the imminent criticality by identifying any oncoming vehicles. This was

possible by implementing an analysis of the quantities that describe the motion of the vehicles present in the scenario and defined MIO.

2. Base the process referred to in the previous point on dedicated parameters, therefore not useful for other functions of the assistance system. This was sought because the problem could have been solved by increasing the numerical value of the limits regarding the TTC and the Relative Distance; however, this solution would have compromised the planning of optimal trajectories in common scenarios such as the one presented in "Case 1". Therefore, it was chosen to differentiate the limits relating to oncoming vehicles from those of the lead vehicle, in order to avoid that the overtaking maneuver to the detriment of the latter begins too early, needlessly increasing the permanence of the ego vehicle in the lane with the opposite direction of travel.
3. Establish a range in which the maneuver can still be performed safely if there is a vehicle in the same lane, with the opposite direction of travel and approaching; that is, when the latter has a distance or speed compatible with the maneuver to be performed safely.
4. Act upstream of the entire planning process; that is, avoiding that the terminal states that foresee the presence of the ego vehicle in a lane in which an "oncoming" vehicle is passing, which does not fall within the parameters imposed to declare the safe maneuver, participate in the subsequent phases of trajectory planning, therefore crossing them out from the set of alternatives.

To recognize the possible presence of critical issues such as that of the case presented, it is first necessary to verify the presence of vehicles with opposite direction of travel. In this sense, the signals containing the states of the vehicles present in the scenario in Frenét coordinates were exploited. Delving further into the specifics, the speed of the examined vehicles $\dot{s}$ was compared with that of the ego vehicle; if the sign of the analyzed value was opposite to that of the respective parameter of the ego vehicle, this would imply an opposite direction of travel.

Subsequently it was introduced an assessment necessary to exclude all those vehicles that are progressively moving away from the ego vehicle, and not approaching it. In this context, the existing TTC signal, was cleverly exploited; remembering that a TTC less than zero implies a condition of progressive approach between two vehicles, as demonstrated previously.

Among the vehicles that meet both of the previous conditions, it was necessary to exclude vehicles that do not imply a risk for the overtaking maneuver even if approaching and traveling in the same lane but in the opposite direction. This exclusion is essential to allow the system to operate lane changes in conditions in which the oncoming vehicle is sufficiently "slow" or "distant". This evaluation was carried out on the parameter relating to the time to collision TTC; as it is a value that takes into account both the relative distance and the relative speed between the two vehicles. In this sense, the threshold value was called $\mathrm{TTC}_{\mathrm{OncomingVehicle}}$ and is a parameter expressed in seconds. The numerical value adopted is equal to 10 s; it was chosen arbitrarily but is compatible with the times necessary for the system to perform a lane change.

As anticipated, in case of detection of such criticality it was chosen to act upstream of the entire planning system. Specifically, all terminal states that imply the presence of the ego vehicle in the same lane as all those vehicles that respect all the conditions previously exposed are eliminated.

The input signals of the function created to perform the exposed task are the terminal states relating to the CC, LCF and LC modes, the signal containing the information relating to the road, the states of the ego and MIO vehicles, and finally the $\mathrm{TTC}_{\mathrm{OncomingVehicle}}$ parameter. The logical steps performed within the function called CheckOncomingVehicle are reported below.

1. The output structures CheckCruiseControlStates, CheckLeadCarFollowingTerminalStates, CheckLaneChangeStates are initialized, which will contain the terminal states that have passed the check for oncoming vehicles. They are initialized with the initStruct structure, which is defined by calling

the HelperLCPlannerDefaultData helper function. By initializing these structures, a zero number of terminal states is associated.

2. If MIO vehicles are present in the scenario, for each of them the relative TTC with the ego vehicle is stored in the MIOInfoTTC vector; this is obtained using the helper function helperCalculateTTC.
3. If the vehicle complies with the three conditions previously defined, the lane of the MIO vehicle is stored in the OncomingVehicleLane vector. This in turn is obtained using the helperDetectLaneNumber helper function.
4. The input terminal states are analyzed and the lane of the ego vehicle is obtained through the helper function helperDetectLaneNumber. If the lane index of a given terminal state coincides with an element present in the previously obtained OncomingVehicleLane vector, for that given modality the initStruct vector would be provided in output, containing a null number of terminal states. It should be remembered that the terminal states of a given modality by construction, all have the same d coordinate; that is, for a given modality the ego vehicle travels along the same lane.
5. If no coinciding indices are obtained in output, the structure relating to a given operating modality of the vehicle that is in input is provided.

4.3.3 Outcomes of the Simulation - Case 2 with Resolved Critical Issues

After introducing the function presented in the previous subsection into the model, in order to make it capable of overcoming critical issues such as those that led to the production of invalid results, set out in subsection 4.3.1, a simulation was performed using the scenario presented as case 2 as input. The simulation was successful. The system was able to produce a perfectly valid result in all its points and no collisions were found. The ego vehicle overtook the lead vehicle exclusively after the passage of the oncoming vehicle; this is due to the fact that the system was able to recognize the imminent criticality, evaluating the presence of the oncoming vehicle as not compatible with the immediate overtaking maneuver to the detriment of the lead vehicle.

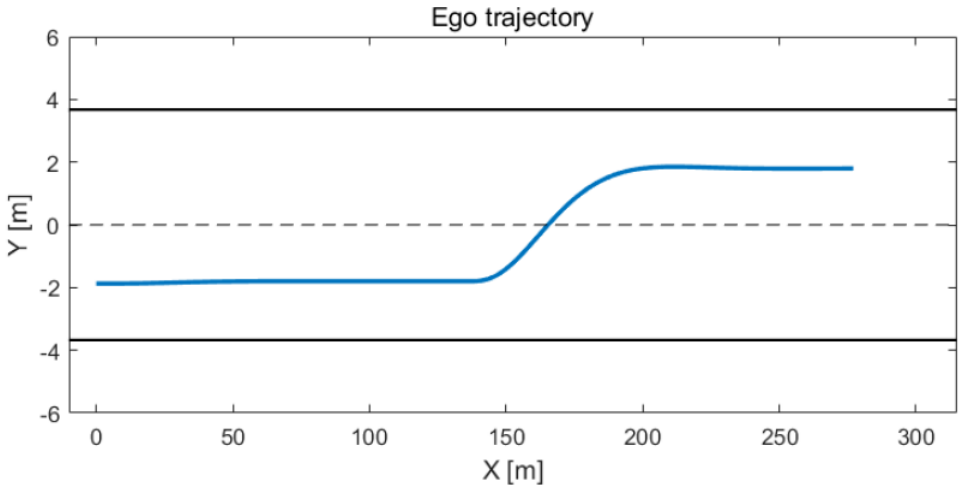

Figure 4.23: Ego vehicle trajectory in the "Overtaking with vehicle in opposite direction" scenario.

Figure 4.23 shows the trajectory actually traveled by the ego vehicle. It is immediately noticeable that the triple lane change obtained by the simulation in the absence of the CheckOncomingVehicle function is absent.

For a better understanding of the simulation results, the scenario representations at the most relevant moments have been shown in Figure 4.24.

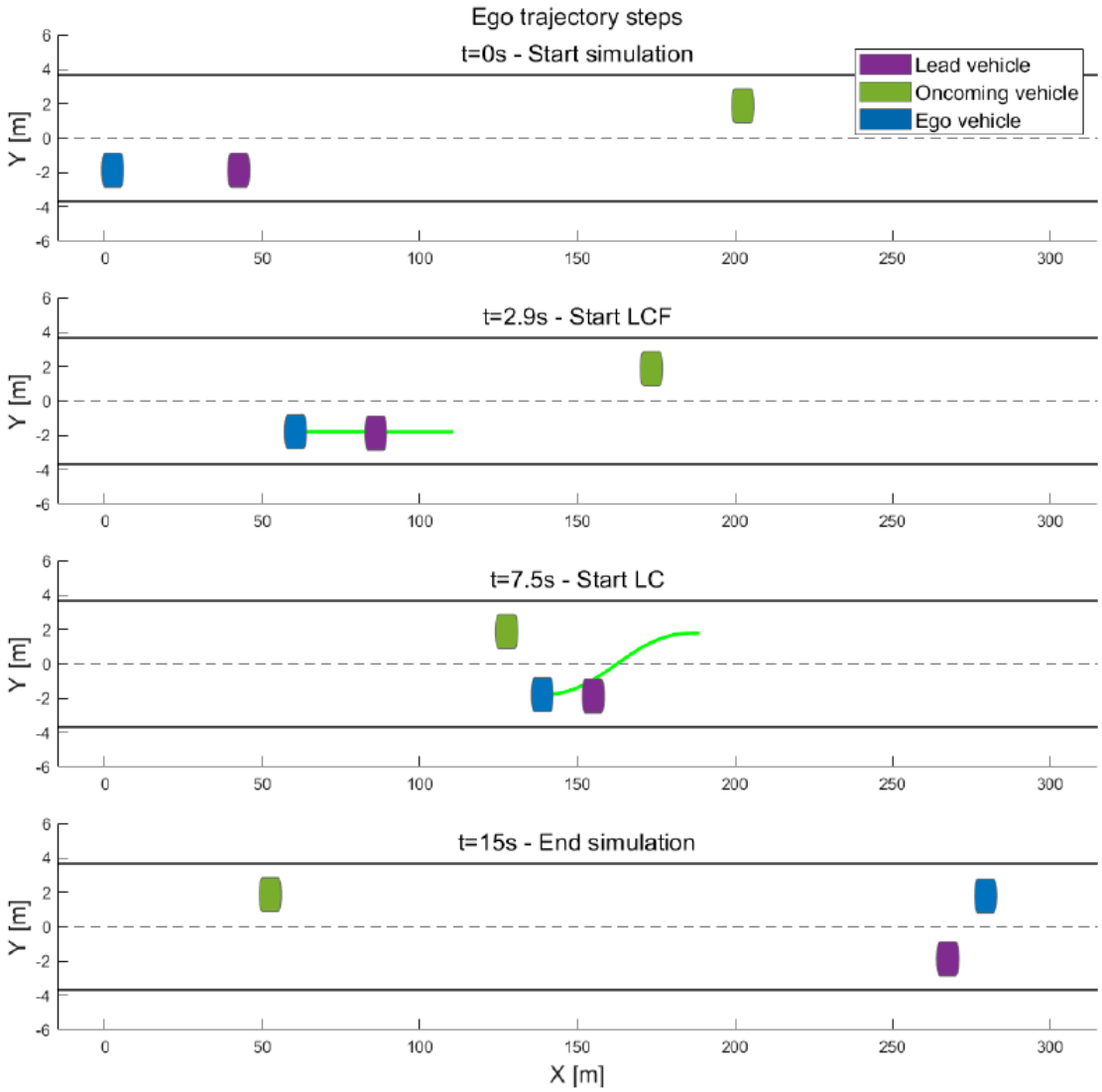

Figure 4.24: Ego vehicle trajectory frame in the "Overtaking with Oncoming Vehicle" scenario.

Table 4.6 shows the peak parameters, relating to the quantities that describe the vehicle motion. It is noticeable that no limit parameter has been exceeded.

Parameter	Max	Limit	Unit
Yaw rate	6.07	20	°/s
Long. Acceleration	$9.41 \cdot 10^{-2}$	5	m/s^2
Lat. Acceleration	1.64	5	m/s^2
Long. Jerk	$8.53 \cdot 10^{-3}$	5	m/s^3
Lat. Jerk	0.28	5	m/s^3

Table 4.6: Peak values of vehicle and ego quantities of the "Overtaking with Oncoming.

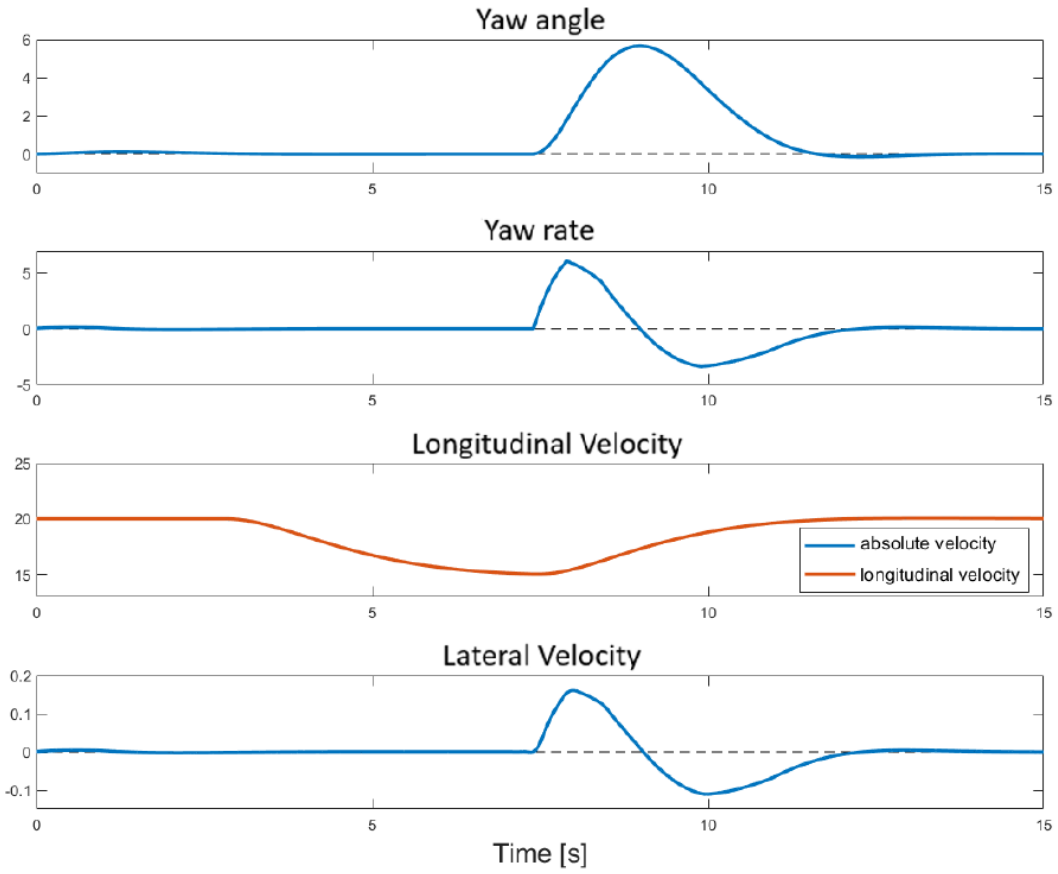

Figure 4.25: Trends of the motion quantities of the ego vehicle in the “Overtaking with Oncoming Vehicle" scenario. Units: [Yaw angle: °], [Yaw rate: °/s], [Longitudinal velocity: m/s], [Lateral velocity: m/s].

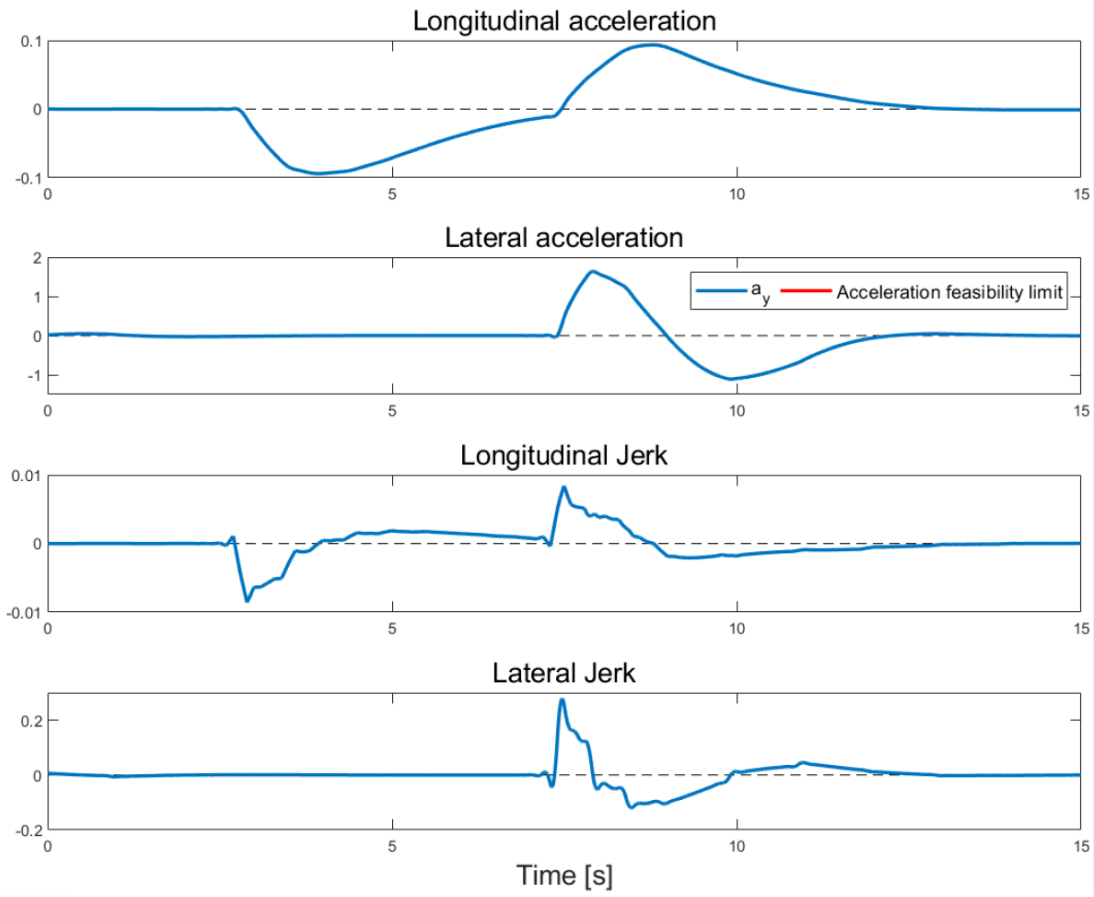

Figure 4.26: Trends of the motion quantities of the ego vehicle (2) in the “Overtaking with Oncoming Vehicle" scenario. Units: [Longitudinal acceleration: m/s²], [Lateral acceleration: m/s²], [Longitudinal jerk: m/s³], [Lateral jerk: m/s³].

Figures 4.25 and 4.26 show the graphs of the main quantities describing the ego vehicle motion.

To avoid repetitions, please refer to paragraph 4.2 for comments on these graphs, since they are partly similar to those obtained from the simulation of the "Single lane change" scenario. Since, after the introduction of the CheckOncomingVehicle function, the ego vehicle follows the trajectory of a simple overtaking, similar to that followed by the ego vehicle in case 1.

However, a difference between the trends reported for case 1 is about the absolute speed. A significant deceleration can be noted, due to the impossibility of overtaking the lead vehicle and a consequent change in operating mode, from CC to LCF; as can be seen in Figure 4.27.

Consequently, compared to case 1, there are different trends for the quantities related to the absolute speed; in this case the longitudinal components of speed, acceleration and jerk.

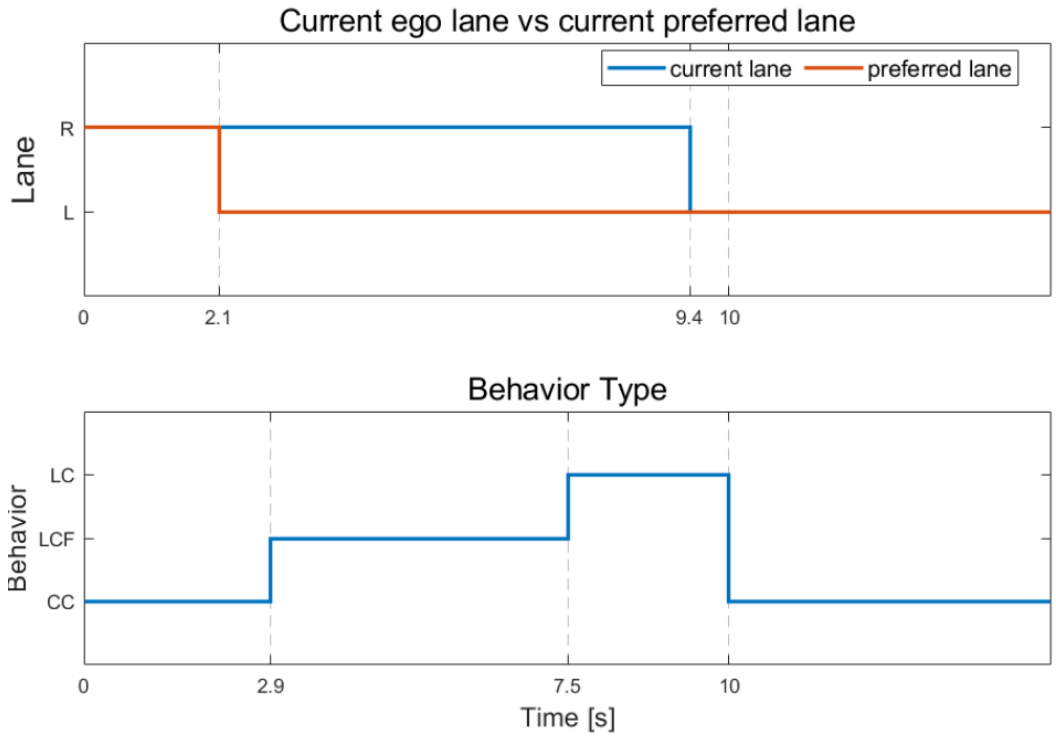

Figure 4.27: Ego vehicle lane in the "Overtaking with Oncoming Vehicle" scenario.

From Figure 4.27 it can be seen how the ego vehicle preferred lane is, after 2.1 s of simulation, the left lane, therefore different from the vehicle current lane; this is due to the detection of a possible collision between the ego vehicle and the lead vehicle while maintaining the trajectory and/or speed described in the scenario.

However, since the CheckOncomingVehicle function prevents the system from enabling the LC function, the only option that does not lead to a collision is to enable the LCF function; which adjusts the speed of the ego vehicle to that of the lead vehicle. The introduced function inhibits the LC mode since in this case the vehicle would find itself traveling in the same lane as the oncoming vehicle; which is approaching, with the opposite direction of travel and with a TTC lower than the threshold, i.e. 10 s, as can be seen from the second graph in Figure 4.28.

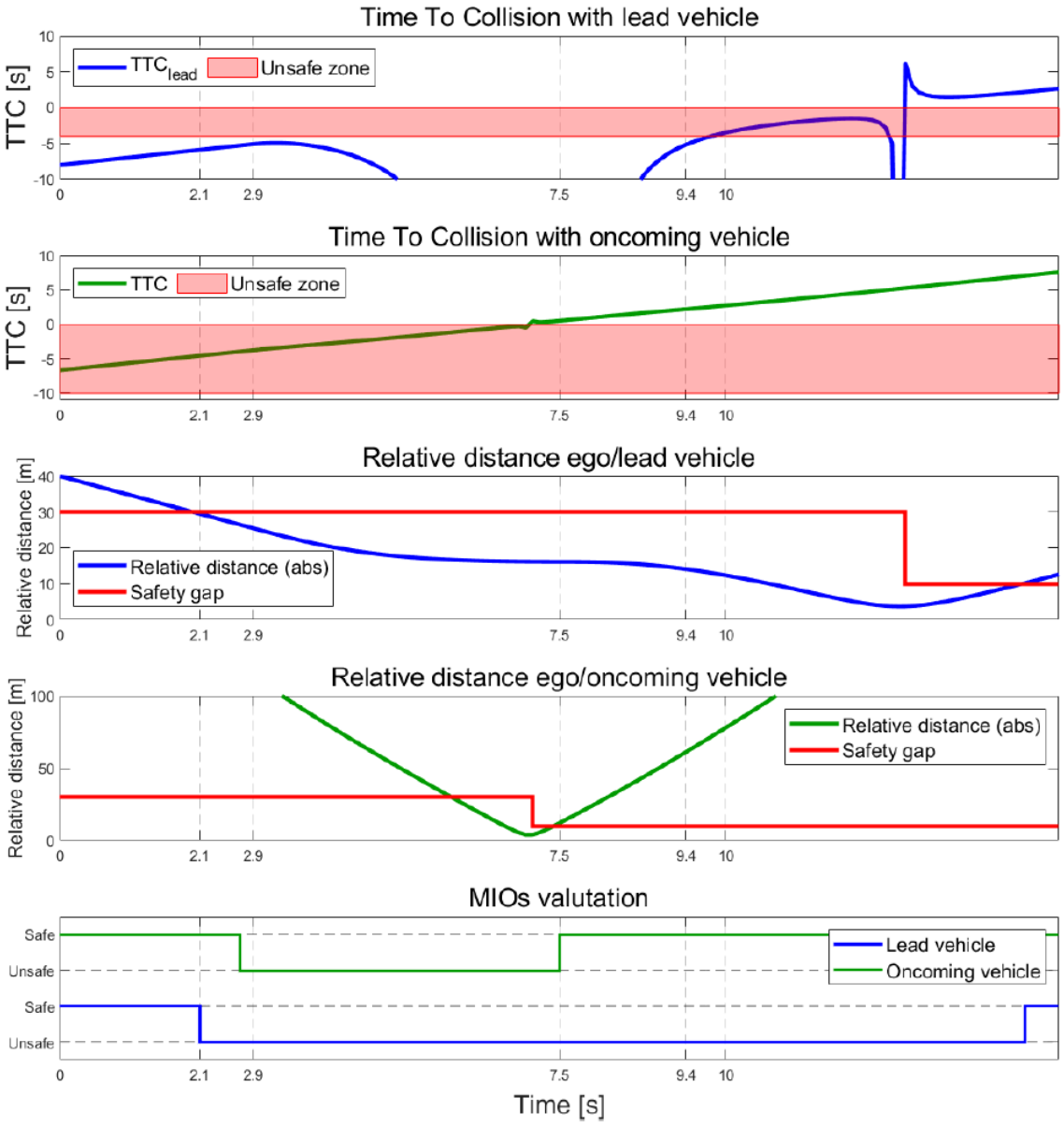

Figure 4.28: MIO vehicle evaluation in the "Overtaking with Oncoming Vehicle" scenario.

A detailed analysis of the graphs in Figure 4.28 shows that:

- From t=0 s to t=2.0 s the vehicle proceeds undisturbed in CC mode.
- At the instant t=2.1 s the Front safety gap threshold calculated between the ego vehicle and the lead vehicle is exceeded. However; this does not imply an immediate switch to LCF mode, since it corresponds to a higher cost parameter than that

associated with the states calculated using CC mode. Although the current lane of the ego vehicle is defined as Unsafe, as a vehicle declared as such is travelling in it, it is not possible to enable LC mode. This happens because, as can be seen from the second graph in Figure 4.28, the TTC calculated between the ego vehicle and the oncoming vehicle from the first moments of the simulation is lower than the imposed threshold; therefore the introduced function inhibits the possibility of a lane change.

- At the instant t=2.9 s, the LCF mode is enabled, since it corresponds to a lower cost parameter than the CC mode. This results in a progressive slowing down of the ego vehicle, with the aim of being able to adjust its speed to that of the lead vehicle. Consequently, the TTC calculated between the two mentioned vehicles increases in modulus and the relative distance is kept constant after an initial transient.
- From the fourth graph, it is possible to obtain the instant in which the Relative Distance parameter calculated between the ego vehicle and the oncoming vehicle respects the imposed threshold again (Rear safety gap). This instant is equal to 7.5 s. Therefore, it can be noted from the fifth graph that for t=7.5 s the oncoming vehicle is declared Safe and the concomitant positive TTC value allows the system to obtain in the set of alternatives also the terminal states that involve the LC mode and to enable it.
- In the subsequent instants, the lane change maneuver is regularly performed and the ego vehicle proceeds along the left lane until the end of the simulation.

$\mathbf{TTC_{OncomingVehicle}}$ parameter study

At the end of this case study, an analysis was performed on the set $TTC_{OncomingVehicle}$ threshold value. As anticipated in the previous subparagraph, the value $TTC_{OncomingVehicle} = 10$ s was associated to it in an arbitrary manner. However, as can be seen from the second graph in Figure 4.28, the TTC value calculated with respect to the oncoming vehicle is well below the threshold imposed from the first moments of the simulation; this may be due to excessive conservatism.

The study presented was performed by varying the threshold parameter and the speed of the ego vehicle; furthermore, the tested scenario is similar to the one presented in this case study, except for the initial position of the ego vehicle.

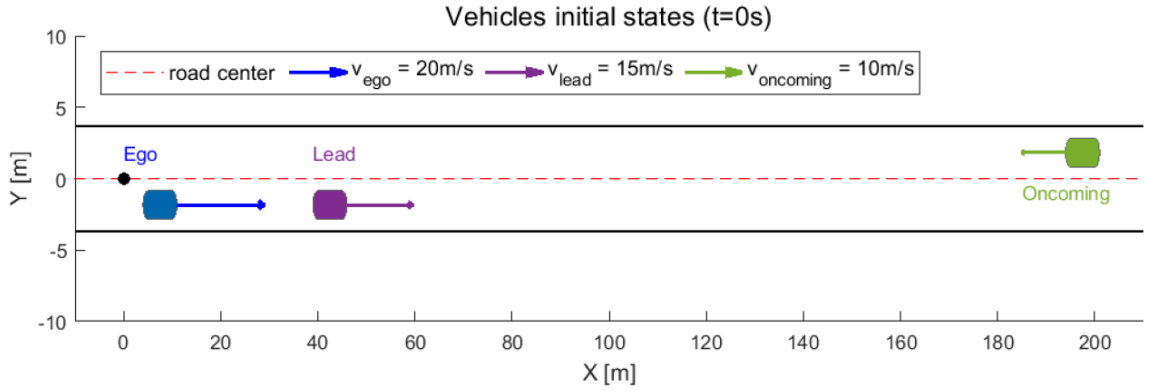

Figure 4.29: Initial states t = 0 s of the "Overtaking with Oncoming Vehicle" scenario.

As can be seen from Figure 4.29, it has been moved forward by 5m. This change was made with the aim of decreasing the distance from the lead vehicle, making the need for overtaking more imminent. If the overtaking maneuver is anticipated, keeping all the other descriptive parameters of the scenario unchanged, this implies a higher value of the TTC calculated with respect to the oncoming vehicle when the LC mode is activated. Consequently, it is preferable to carry out the study in this condition, as it guarantees greater sensitivity to the $TTC_{OncomingVehicle}$ parameter.

Furthermore, the initial value of the Time-to-Collision, indicated with TTC_0, between the ego vehicle and the oncoming vehicle was determined, corresponding to the different speed

values assigned to the ego vehicle. The results obtained from the simulations are reported in Table 4.7.

		TTC_0	$TTC_{OncomingVehicle}$			
			4s	5s	6s	8s
v_{ego}	60 km/h	7.3 s	Triple LC	Activated	Activated	Activated
	70 km/h	6.62 s	Stop	Triple LC	Activated	Activated
	80 km/h	6.05 s	Colliding	Colliding	Activated	Activated
	90 km/h	5.57 s	Deactivated	Deactivated	Activated	Activated

Table 4.7: $TTC_{OncomingVehicle}$ sensitivity study in the "Overtaking with Oncoming Vehicle" scenario.

It can be noted that five different system behaviors were recorded; they must be interpreted in the following way:

- Activated: The $TTC_{OncomingVehicle}$ threshold was exceeded, consequently the LC mode was inhibited and the ego vehicle overtook the lead vehicle only after the oncoming vehicle passed to its left, in a manner similar to what was seen in this paragraph.
- Triple LC: The $TTC_{OncomingVehicle}$ threshold was exceeded after the lane change had already begun; however, since a short time had elapsed since the start of the maneuver, it was possible to safely return to the lane.
- Stop: The system stopped the motion of the ego vehicle because it had not identified a safe and passable trajectory. The nature of this error will be explored in the next paragraph.
- Colliding: The ego vehicle performs the overtaking maneuver; however, since it has no possibility of returning to the lane, it collides with the oncoming vehicle.
- Deactivated: The $TTC_{OncomingVehicle}$ threshold is not exceeded and the ego vehicle

regularly performs the overtaking maneuver, not colliding with the other vehicles in the scenario.

Based on what is shown in Table 4.7, it can be stated that the value initially associated with the threshold (10s) is very conservative; since assuming a value of $TTC_{OncomingVehicle}$ = 6 s, no error is recorded. However, in order to make the implemented strategy work correctly even for the most critical scenarios, it is advisable to adopt a higher value, in order to increase the conservativeness of the assistance system.

4.4 Case 3 - Overtaking a Line of Vehicles

The "Overtaking a Line of Vehicles" scenario was introduced with the aim of testing and improving the assistance system ability in planning the correct maneuver when faced with a group of vehicles to be overtaken, with a large overall size. Defining a "line of vehicles" as a group of vehicles traveling on the same trajectory and with narrow distance constraints. The main problem brought by the scenario is given by the impossibility for the ego vehicle to return to the lane between one vehicle and another in the line, due to a space between the vehicles that is not compatible with such a maneuver.

The objective of the reported analysis is to build a trajectory planning mechanism that is able to evaluate eventualities such as the one in question; in order to evaluate the maneuver taking into account a larger overall dimensions compared to the lead vehicle alone.

Going into more detail, this scenario consists of six different vehicles traveling along a 700 m straight road with two lanes, one for each direction of travel.

The vehicles that make up the scenario are the following:

- an ego vehicle;
- a lead vehicle, traveling along the same lane as the ego vehicle and with a lower speed than the latter;
- three obstacle vehicles, specifically called obstacle 1, obstacle 2 and obstacle 3, which precede the lead vehicle, proceed at the same speed as the latter and distance each other by an equal amount;
- a vehicle that proceeds in the opposite direction and in the opposite lane to that in which the previously described vehicles travel; which has been called the oncoming vehicle.

Table 4.8 shows the scenario configuration parameters; furthermore, it is specified that all the vehicles present have the same geometry.

ego vehicle lane	2
lead vehicle lane	2
obstacle vehicles lane	2
oncoming vehicle lane	1
$s_{lead} - s_{ego}$	35 m
$s_{oncoming} - s_{ego}$	480 m
$distance_{obstacle}$	13 m
v_{ego}	20 m/s
v_{lead}	15 m/s
$v_{obstacles}$	15 m/s
$v_{oncoming}$	10 m/s

Table 4.8: Vehicle data of the "Overtaking a Line of Vehicles" scenario.

Examining the data reported in Table 4.8 it is necessary to specify that, assuming the point of view of the ego vehicle, the lane with index 2 is the right one, where the lead vehicle passes and initially also the ego vehicle; instead, the lane with opposite direction of travel, the lane belonging to the obstacle vehicle 2, has been identified with index 1. Furthermore s_{ego}, $s_{oncoming}$ and s_{lead} represent the curvilinear coordinate of the vehicles at the initial instant. It is also worth mentioning that the term $v_{obstacles}$ identifies the speed of the components of the line of vehicles and the term $distance_{obstacle}$ refers to the distance between the vehicles belonging to the line, calculated with respect to the relative rear axle.

Figure 4.30 shows a graphic representation of the scenario at the initial moment of the simulation. The limits of the roadway have been reproduced by means of two continuous black lines and the line that delimits the two lanes, therefore the two directions of travel, has been created with a dashed line. The latter coincides with the reference curve of the Frenét

coordinates, or center line and the way points with which it was created are reported in graphic form by means of black dots and are: [0, 0] and [700, 0].

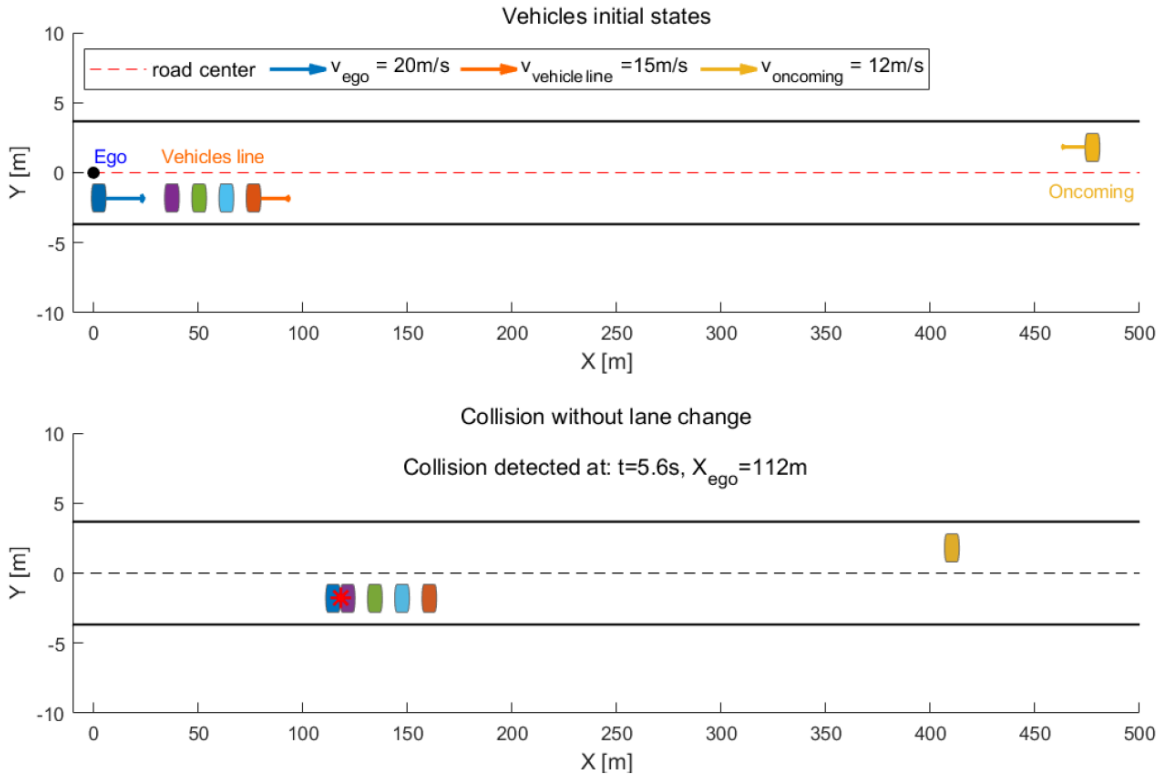

Figure 4.30: Initial states t = 0 s of the "Overtaking a Line of Vehicles" scenario.

In Figure 4.31 an enlargement of the scenario at the initial instant has been reported; thanks to which it was possible to graphically identify the line of vehicles, the individual components and the relative distances. For simplicity, a single red speed vector $v_{obstacles}$ has been represented, valid for all vehicles belonging to the line.

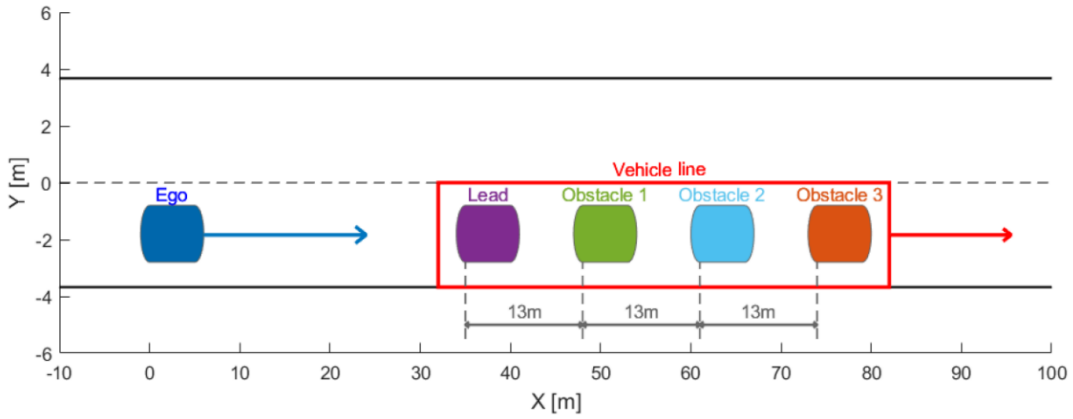

Figure 4.31Line of vehicles of the "Overtaking a Line of Vehicles" scenario.

Also in this case the scenario was constructed in such a way as to require the intervention of the assistance system; this is due to an incompatibility between the line of vehicles and the ego vehicle regarding the trajectories and speeds. This incompatibility, in the absence of interventions on the trajectory of the ego vehicle and/or on its speed, leads to a collision after 5.6 s from the initial instant of the simulation, as can be seen from the second graph shown in Figure 4.30.

It can be easily noted, given the reduced distance between the vehicles that is kept constant due to the equality between the speeds of the line components, that the ego vehicle cannot re-enter between one vehicle and the other in the event of overtaking but can complete the maneuver only after having passed the front vehicle (vehicle at the head of the line). In the case in question, this denomination is associated with the obstacle vehicle 3, represented in orange in the graph in Figure 4.31.

This scenario was then reproduced within the Driving scenario designer application, in order to obtain the correct input for the model used. The results obtained from the simulation are examined in the following sub-paragraph.

4.4.1 Outcomes of the Simulation - Case 3

The simulation examined below was performed by keeping active the model modifications that were introduced following the analysis of the results of the previous case study, "Overtaking with Oncoming Vehicles" in paragraph 4.3. The analogy with the previous case lies in the progressive approach of an oncoming vehicle in the lane with the opposite direction of travel; instead, the difference is linked to the extension of the time spent in the same lane as the oncoming vehicle, since to complete the overtaking maneuver the ego vehicle must pass the entire line of vehicles.

The simulation was not completed successfully, because, for a certain interval of time, the model did not identify a valid trajectory among the alternatives of CC, LCF and LC. During this interval of time the vehicle appears to be stationary and waiting for one of the three operating modes to be considered valid and physically drivable. The analysis of the results reported below will focus on the causes of this anomaly and will not delve into what happens during the subsequent moments, as all the recorded values are considered invalid. In the following Figure 4.32, the trajectory actually followed by the ego vehicle during the simulation has been reported on a Cartesian plane.

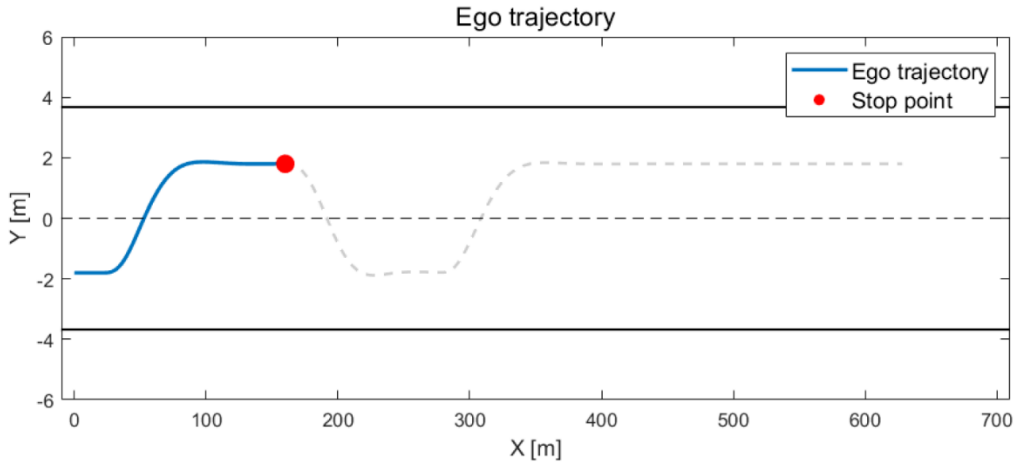

Figure 4.32: Ego vehicle trajectory in the " Overtaking a Line of Vehicles" scenario.

To better understand the reasons why the result of the simulation in question cannot be considered acceptable, it is useful not to limit oneself to the trajectory traveled by the ego

vehicle and to analyze the evolution of its state over time. For this purpose, a graph has been produced, reported in Figure 4.33, which shows the positioning of the vehicles within the scenario at specific moments in time.

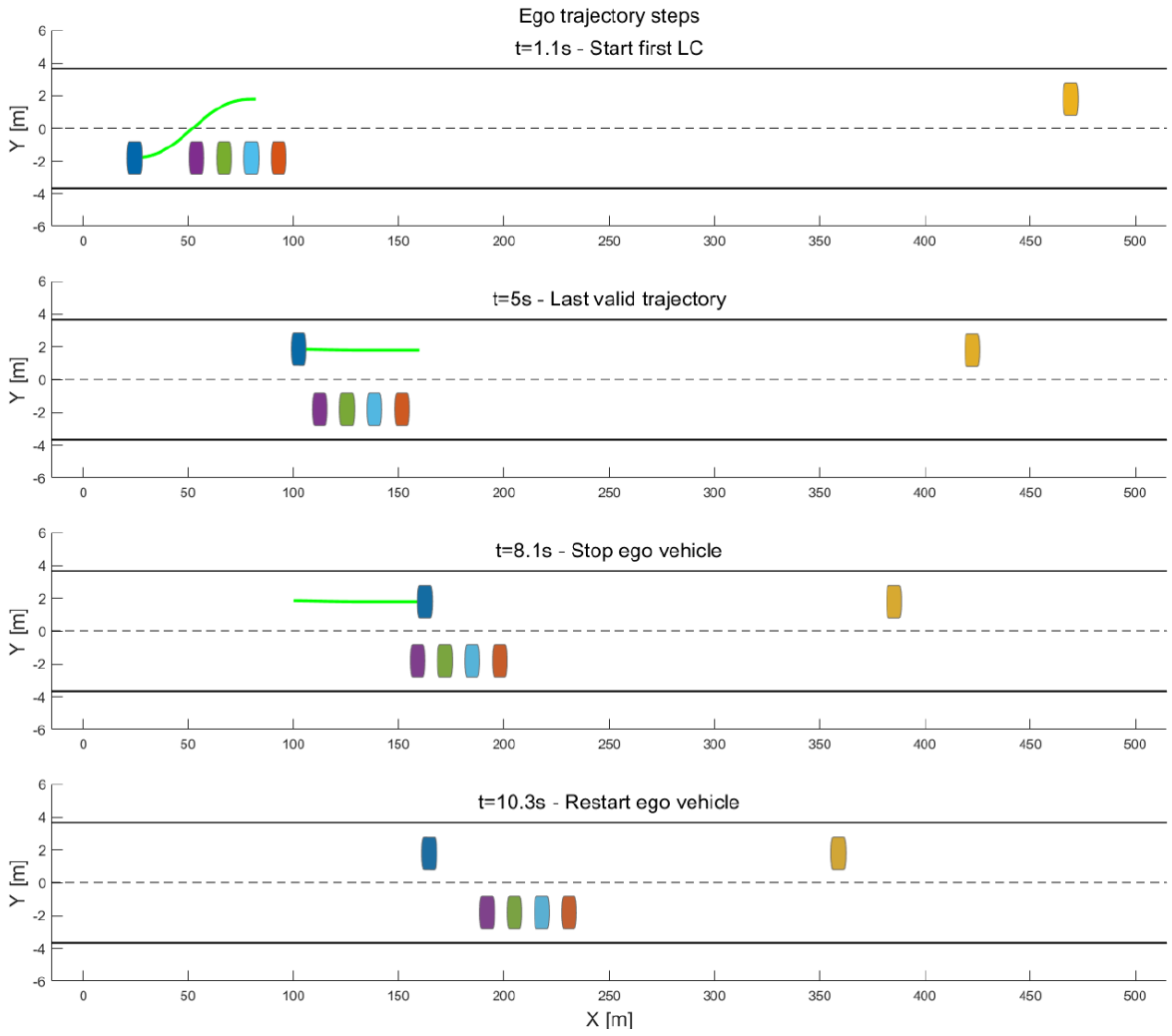

Figure 4.33: Ego vehicle trajectory frame of the " Overtaking a Line of Vehicles" scenario.

The analysis of each of the graphs inserted in figures 4.32 and 4.33 is shown below:

- t = 1.1 s: The ego vehicle reaches such a distance from the first vehicle of the line, called vehicle lead and shown in purple, to enable the lane change mode.
- t = 5 s: the planner identifies a straight trajectory as the reference trajectory that keeps the ego vehicle in the center of the left lane. This segment has a time horizon of 3 s

and is shown in green in the second graph in Figure 4.35.

- From t = 5.1 s to t = 8 s: The ego vehicle, after completing the lane change maneuver, has already started to overtake part of the line of vehicles; However, the imminent approach of the oncoming vehicle (represented in yellow) along the same lane and in the opposite direction of travel leads to an incompatibility with the current configuration of the model. At the instant t = 5.1 s the TTC calculated between the ego vehicle and the oncoming vehicle drops in modulus below the imposed $TTC_{OncomingVehicle}$ threshold. The consequence is the elimination of the modes that provide for the planning of trajectories implying the presence of the ego vehicle in the same lane as the oncoming vehicle, in this case those related to the CC and LCF. Consequently, the only mode that can be used is the LC; but even in this case it is not possible to plan the relative trajectories, since, due to the size of the line of vehicles, the ego vehicle cannot return to the lane without impacting with one of its components. In this time interval the ego vehicle continues to follow the trajectory segment identified in the previous point.
- t = 8.1 s: Not having a trajectory to follow, the model freezes the position of the ego vehicle, stopping its motion at the last planned point until a new trajectory to follow is identified.
- t = 10.3 s: During the stop of the ego vehicle, the line of vehicles continues to advance along its own trajectory, until it overtakes the ego vehicle with respect to the curvilinear coordinate s. At the instant equal to 10.3 s there is a sufficient distance between the line of vehicles and the ego to allow the latter to re-enter the lane. Therefore, the ego vehicle instantly returns to motion to follow the new planned trajectory.

Thanks to this general overview of the behavior of the ego vehicle during the simulation, the two main problems that led to considering the obtained result invalid were identified:

- The ineffective planning strategy led to a lane change to overtake the line of vehicles, in the absence of suitable conditions to complete the maneuver.

- The simulation involves a vehicle stop from the instant t = 8.1 s to t = 10.3 s. As anticipated, this is due to the absence of a viable solution among the set of alternatives; a set from which the solutions that include the CC and LCF modes have been eliminated. This is not an actual stop of the vehicle, such as an emergency braking due to the criticality of the situation, but an error in the simulation. In other words, at the instant in which the stop ends, the vehicle resumes driving with exactly the same values of the parameters characterizing its motion, as if the stop had never occurred.

At the end of the stop, the simulation continues, the ego vehicle returns to the lane and subsequently, after having let the oncoming vehicle pass, passes the line of vehicles. The trajectory described after the stop is shown in gray in Figure 4.32. However, as anticipated, it was chosen not to delve into what happens after the stop period, as the data recorded during this criticality have no physical validity; therefore, they completely invalidate the simulation.

The deduction regarding the nature of the vehicle stop emerges from the analysis of the descriptive parameters of the ego vehicle motion. In particular, the acquisitions show a freezing of all the characterizing values, such as: position, yaw angle, speed and acceleration; which remain constant during the interval mentioned (from t = 8.1 s to t = 10.3 s). This occurrence can only be interpreted as a system error, since it would be impossible to have non-zero speed values while keeping the vehicle still in the same position.

Consequently, the model makes the ego vehicle resume motion with a state coinciding with that of the first instant of stop, as if no irregularity had occurred. This implies that the nature of the error is attributable exclusively to the model planning logic; therefore, with the exception of the interval in which all the motion quantities remain constant, no anomalies are found in their trends. For this reason, it was considered unprofitable for the purposes of analyzing the problem to report the related graphs, but for completeness in the following Table 4.9 the peak values of the main quantities are reported; they respect all the imposed limits.

Parameter	Max	Limit	Unit
Yaw rate	6.59	20	°/s
Long. Acceleration	0.11	5	m/s^2
Lat. Acceleration	2.31	5	m/s^2
Long. Jerk	$1.09 \cdot 10^{-2}$	5	m/s^3
Lat. Jerk	0.43	5	m/s^3

Table 4.9: Peak values of ego vehicle quantities in the " Overtaking a Line of Vehicles" scenario.

For a better understanding of the problems highlighted, it is useful to analyze the trend of the signals used by the planner to establish the trajectory of the ego vehicle. Figure 4.34 shows the trend of the preferential lane and the current lane (first graph) and the operating mode of the vehicle over time (second graph).

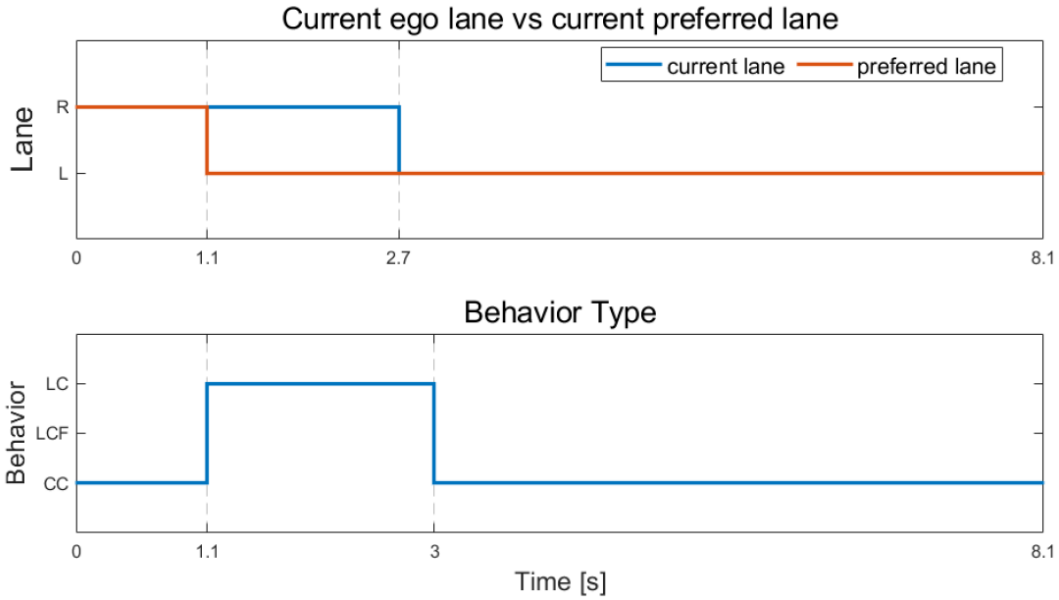

Figure 4.34: Ego vehicle lane in the "Overtaking a Line of Vehicles" scenario.

In Figure 4.35 the trends of the main quantities useful for the trajectory planner assessments regarding the comparison between the ego vehicle and the other vehicles present in the

scenario have been superimposed. Specifically, the following graphs have been reported:

- Relative to the TTC and the Relative distance with the lead vehicle, with which the ego vehicle risks colliding in the absence of a change in the planned trajectory.
- Of the TTC calculated with respect to the oncoming vehicle. A larger Unsafe zone is noted with respect to that relating to the lead vehicle, as in this case it is a vehicle that travels in the opposite direction to that of the ego vehicle. The threshold value is the one established downstream of the study carried out with the analysis of the previous case "Overtaking with Oncoming Vehicle", paragraph 4.3.
- On the safety assessment (Safe/Unsafe) relating to the two vehicles mentioned.
- Finally, it is specified that the most significant time instants and the relative operating mode of the planner have been highlighted on the ordinate axis; with the aim of helping to understand how the various signals have influenced the planning of the ego vehicle.

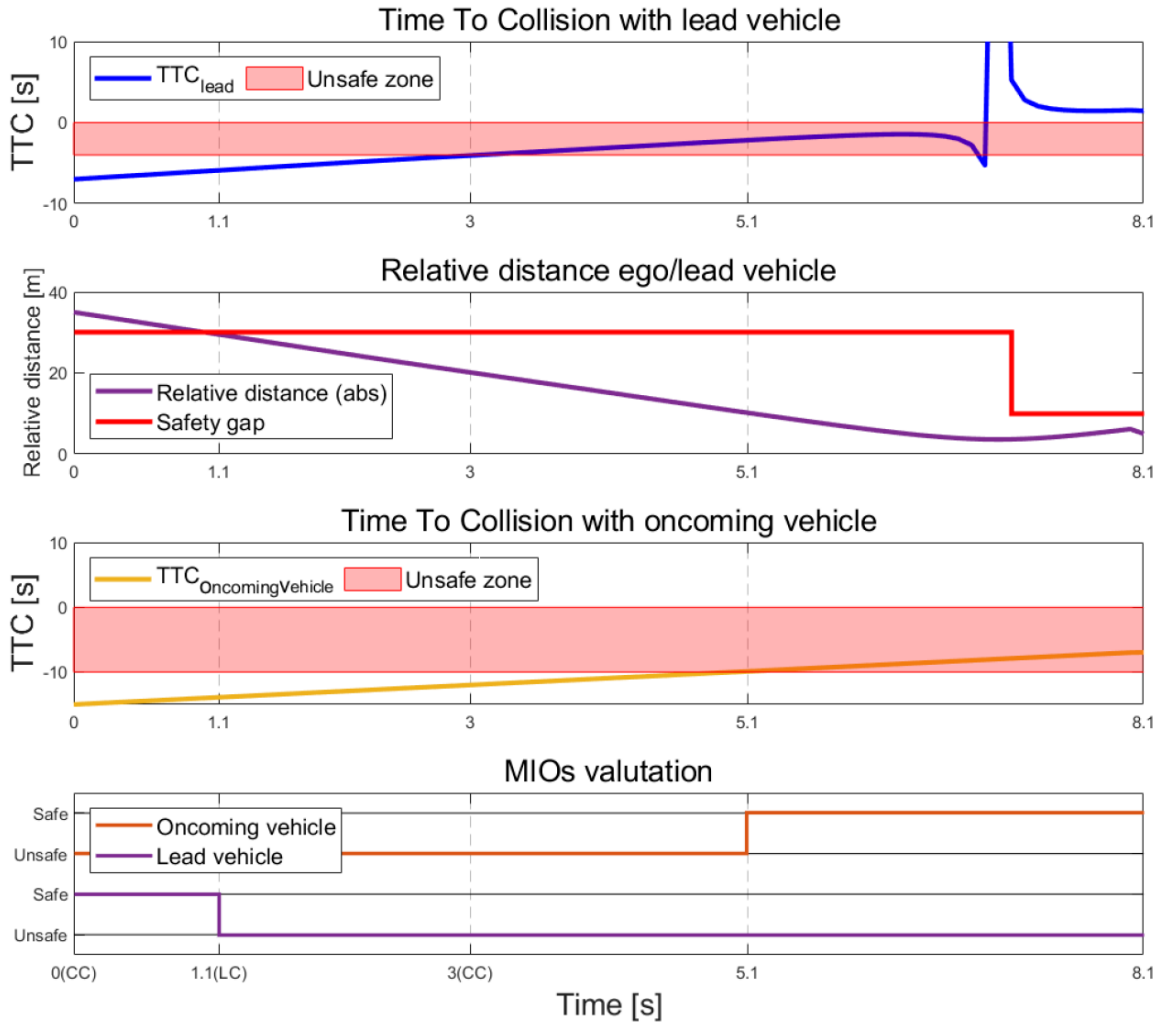

Figure 4.35: MIO vehicle evaluation in the " Overtaking a Line of Vehicles" scenario.

By comparing the graphs presented in figures 4.32, 4.33 and 4.35, it is possible to provide a more detailed interpretation of the causes that led to the two problems previously exposed. The analysis by points is reported below, carried out following the chronological line of events.

- From t = 0 s to t = 1.0 s: The ego vehicle follows the trajectory preset during the scenario construction phase without encountering the need for a new planning.
- t = 1.1 s: The ego vehicle enters LC mode. This occurs because the limit imposed on the Relative distance with the vehicle preceding it, in this case the lead vehicle, is

exceeded (second graph in Figure 4.35). This implies the variation of preferred lane (first graph in Figure 4.34) and due to this variation the planner identifies as a reference the trajectory that leads to the lane change.

- t = 2.7 s: The CC mode is enabled again; because, as can be seen from the graph in Figure 4.36, the ego vehicle has now reached the left lane.
- t = 5.1 s: The TTC, calculated between the ego and oncoming vehicles, is lower in modulus than the imposed threshold ($TTC_{OncomingVehicle}$). This implies the logic chain introduced with the function illustrated through the previous case study, paragraph 4.3 "Overtaking with Oncoming Vehicle"; which leads to the elimination of all terminal states that imply the placement of the ego vehicle in the same lane as the oncoming vehicle at the given time horizon. Consequently, the planner eliminates all terminal states that maintain the current lane occupied by the ego vehicle, i.e. those that involve the CC and LCF modes. From this exclusion, only the terminal states that involve a lane change (LC mode) remain in the set of alternatives. However, as can be seen in Figure 4.35, the ego vehicle does not have enough space available to be able to return to the lane.
- t = 6.1 s: Please note that the model plans a new trajectory every time a flag is activated; the latter assumes the logical value 1 with a periodic cadence of 1 s unless there are further alarms coming from the processed signals, for example a change in preferred lane. In the time between t = 5.1 s and t = 6.1 s there are no signal variations that lead to the evaluation of the set of available alternatives; consequently, only after the model planning time interval (1s) has passed, i.e. at the instant equal to t = 6.1 s, the terminal states are evaluated. During the evaluation, the impossibility of following one of the selected trajectories emerges, since all the available alternatives lead to a collision with the vehicles in the queue during the possible return to the lane. The ego vehicle, in the absence of new trajectories, continues to follow the last identified trajectory.
- t = 8.1 s: The last trajectory segment followed by the ego vehicle was planned at the

simulation instant t = 5.0 s with the maximum available time horizon (3s). Consequently, the vehicle follows the available trajectory up to the current instant and then stops because it does not have a new segment to follow. This happens because at each instant the available alternatives have been evaluated; but all lead to a collision during the possible re-entry into the lane maneuver without having yet passed the line of vehicles, as can be seen in Figure 4.33.

4.4.2 System Adaptation to Presented Critical Issues - Case 3

The considerations made during the analysis of the results led to the identification of two main critical issues:

- The lack of an optimal planning strategy, since three different lane changes are performed to overtake.
- The presence of a stop in the motion of the ego vehicle, which led the entire simulation to be considered invalid.

The cause of the stop originates from the impossibility of following the trajectories available in the set of alternatives. The latter was reduced due to the presence of the oncoming vehicle, following the introduction of the CheckOncomingVehicle function in the model. However, the available combinations, i.e. those that include the lane change, all lead to a collision with the vehicles in the queue, since the ego vehicle had already started the overtaking maneuver. This analysis leads to considering the two problems stated as related to each other, since it is due to the first lane change that the vehicle begins an overtaking maneuver that it cannot complete.

Adopted solution

The logic with which this criticality was addressed follows the principles adopted in the previous case study. It is specified that this strategy was developed assuming that the data of each vehicle present in the scenario is available; therefore, assuming that the ego vehicle is

equipped with a V2X (vehicle-to-everything) system.

1. Recognizing the imminent criticality. In this case, it is a matter of not limiting the analysis of the scenario to only the MIO vehicles, but of identifying the presence of the line of vehicles and in particular identifying which of its components is the front vehicle.
2. Base the process on dedicated parameters and with a strategy dedicated to this criticality. This is necessary because a valid result would be obtainable by increasing the value of the limit parameter $TTC_{OncomingVehicle}$ used in the CheckOncomingVehicle function, introduced in the case study "Overtaking with Oncoming Vehicle". However, this solution was not adopted because for scenarios where there is a single vehicle preceding the ego vehicle, a value of the $TTC_{OncomingVehicle}$ parameter would prevent the overtaking maneuver at sufficient distances to still perform the maneuver in complete safety.
3. Establish a criterion that allows discerning when the overtaking maneuver can still be performed safely, even if what must be overtaken is an entire line of vehicles.
4. Act upstream of the entire planning process. Specifically, it was chosen to remove from the subsequent phases of trajectory creation and evaluation all those terminal states that do not comply with the imposed criteria, or those that lead to incurring criticalities similar to the one presented.

The function introduced to adapt the model to this criticality as well is called CheckLeadExtension and its main purpose is to check the longitudinal size of the vehicles that must be overtaken during the overtaking maneuver.

- The first task of the function is to obtain all the parameters necessary for the evaluations carried out later, such as: lane of the Target vehicles, distance of the same from the ego vehicle and the relative TTC. Note that in this case, working with information relating to vehicles that may have other vehicles between them and the ego vehicle, the information relating to only the MIO vehicles cannot be used, but the data of all the vehicles present in the scenario (Target vehicles) must be verified.

- Subsequently, it is necessary to verify the presence of a lead vehicle in the scenario; otherwise, since there is no vehicle to overtake, this criticality cannot arise.
- If a lead vehicle is present, it is necessary to verify the presence of a line of vehicles; to do this, a threshold distance between the vehicles was chosen to consider them part of a line. The chosen threshold is the following:

$$\text{Threshold distance} = \text{frontSafetyGap} + \text{rearSafetyGap}$$

  The reason why this threshold was chosen is related to the possible impossibility of the ego vehicle to perform a return to the emergency lane between the two vehicles; since in order to perform this maneuver it must have a distance equal to frontSafetyGap from the vehicle preceding it and equal to rearSafetyGap from the vehicle following it.
- Consequently, if the distance between two vehicles was less than the sum of the two safety distances, the ego vehicle could not place itself in the space that separates them; therefore, the two vehicles can be considered part of a line.
- After having identified a possible vehicle that meets the previous condition, the check is carried out by looking for a further vehicle that precedes the one just identified. The loop continues until the vehicle that "opens" the line, called front, is identified.
- An approximate estimate of the time needed to overtake the entire line TTO (time to overtake) is calculated:

$$\text{TTO} = \frac{s_{\text{ego}} - s_{\text{front}}}{\dot{s}_{\text{ego}} - \dot{s}_{\text{front}}} + \text{lineTTC}$$

  The parameters were calculated with respect to the front vehicle, since it is the furthest from the ego vehicle among those belonging to the line. Furthermore, a constant called lineTTC was added, that takes the value of the egoTTC parameter, i.e. the standard threshold on the TTC. The latter has a value equal to 4s, which is considered sufficient for returning to the lane after overtaking the front vehicle, since the planner has a maximum time horizon of 3s.

- The presence of vehicles traveling in the opposite direction and approaching is checked. If this check is positive, the function stores the lane of vehicles that meet the following condition:

$$|\mathrm{TTC}| < \mathrm{TTO}$$

- After storing the lanes of vehicles with a high risk of collision, the function proceeds in a manner completely analogous to that illustrated for the CheckOncomingVehicle function; that is, it eliminates from the set of alternatives all the terminal states that involve the placement of the ego vehicle in one of the lanes identified downstream of the previous check at the given time horizon.

4.4.3 Outcomes of the Simulation - Case 3 with Resolved Critical Issues

After having thoroughly exposed the updates made to the model with the introduction of the CheckLeadExtension function, a new simulation of the scenario presented in this case study was performed. This was useful to evaluate the actual adaptation of the model to the criticality encountered. In figure 4.36 there is a representation on the XY Cartesian plane of the trajectory actually followed by the ego vehicle.

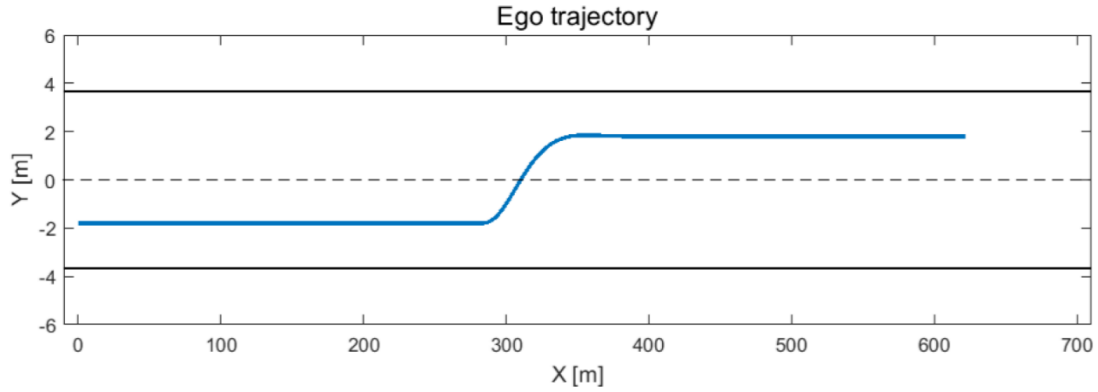

Figure 4.36: Ego vehicle trajectory in the "Overtaking a Line of Vehicles" scenario.

The peak values of the quantities that characterize the vehicle motion are reported below in table 4.10.

Parameter	Max	Limit	Unit
Yaw rate	6.05	20	°/s
Long. Acceleration	$9.41 \cdot 10^{-2}$	5	m/s^2
Lat. Acceleration	1.64	5	m/s^2
Long. Jerk	$8.57 \cdot 10^{-3}$	5	m/s^3
Lat. Jerk	0.28	5	m/s^3

Table 4.10: Peak values of ego vehicle quantities of the "Overtaking a Line of Vehicles" scenario.

The simulation was successful, no stops were found like the one recorded in the absence of the CheckLeadExtension function, the trajectory is not characterized by useless maneuvers like two of the three lane changes in the previous simulation and the descriptive parameters of the motion are well below the imposed threshold.

In figure 4.37 the most important moments of the simulation are represented by capsules. It should be noted that the initial moment has not been included, since it was already presented in the introduction phase of the paragraph.

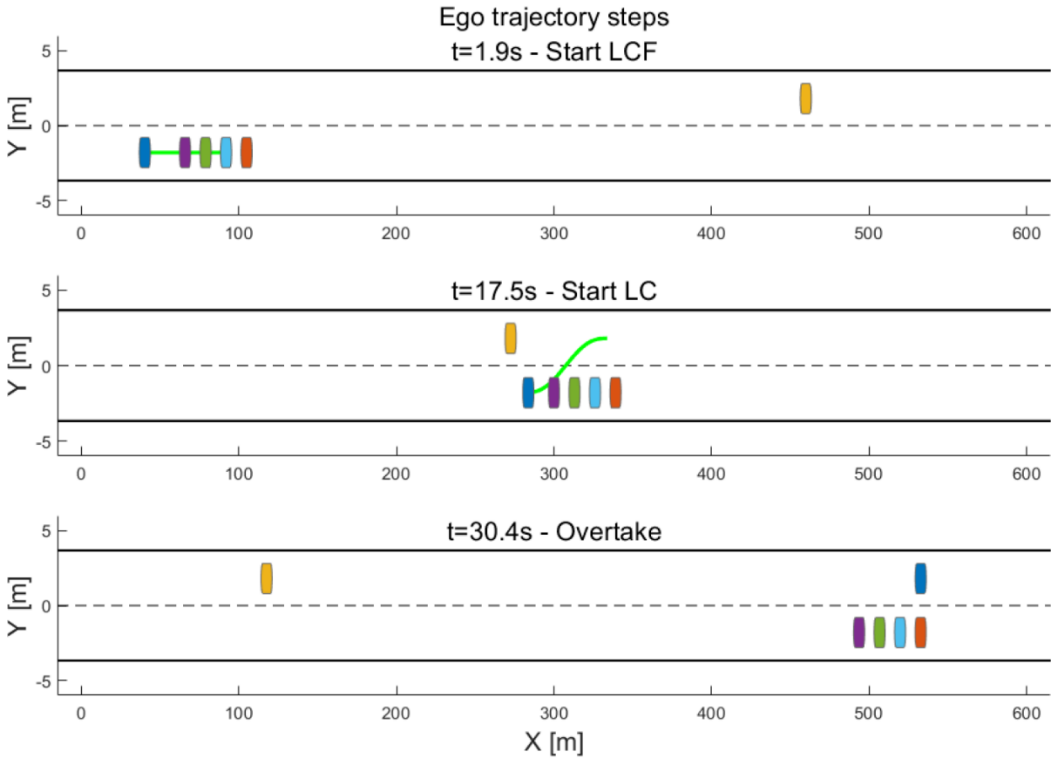

Figure 4.37: Ego vehicle trajectory frame in the "Overtaking a Line of Vehicles" scenario.

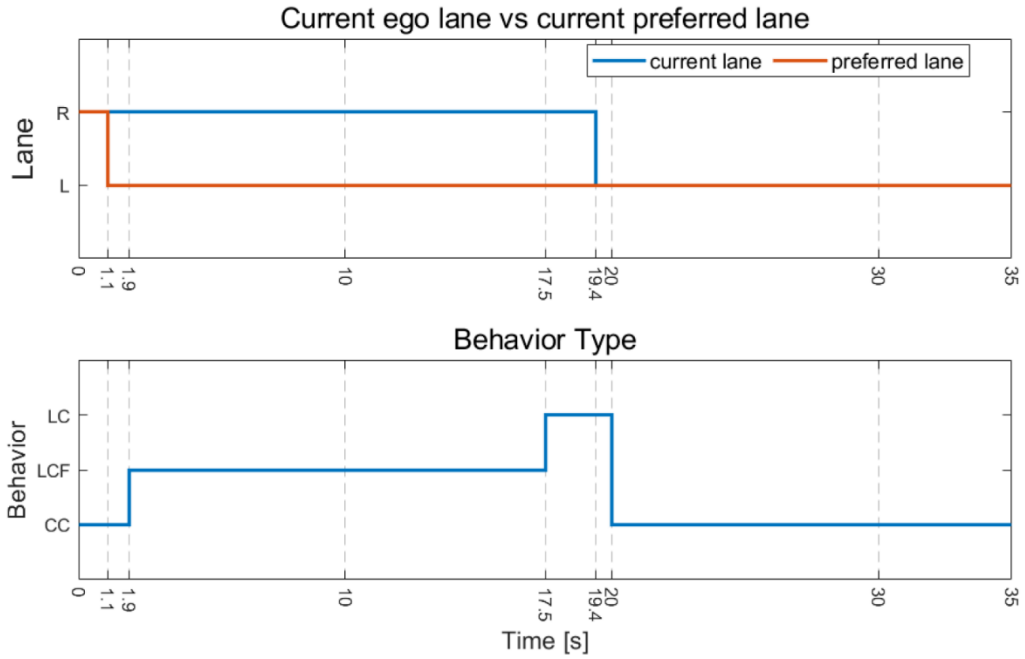

Figure 4.38: Ego vehicle lane in the "Overtaking a Line of Vehicles" scenario.

In figure 4.38 the trends relating to the current and preferential lane of the ego vehicle and to its operating mode over time have been reported. The most relevant moments have been reported in numerical value on the x-axis. From the first graph it can be seen how the lane change function was inhibited in the initial phase of the simulation, since the preferential lane and the current lane appear to be different for a not insignificant interval of time.

Figure 4.39 superimposes the trends of some quantities relevant to the planner. Specifically, the following have been reported: the Relative distance between the ego and lead vehicles, ego and front and ego and oncoming; furthermore, the trend of the TTC relative to the oncoming vehicle has also been reported in comparison with the TTO relative to the row of vehicles.

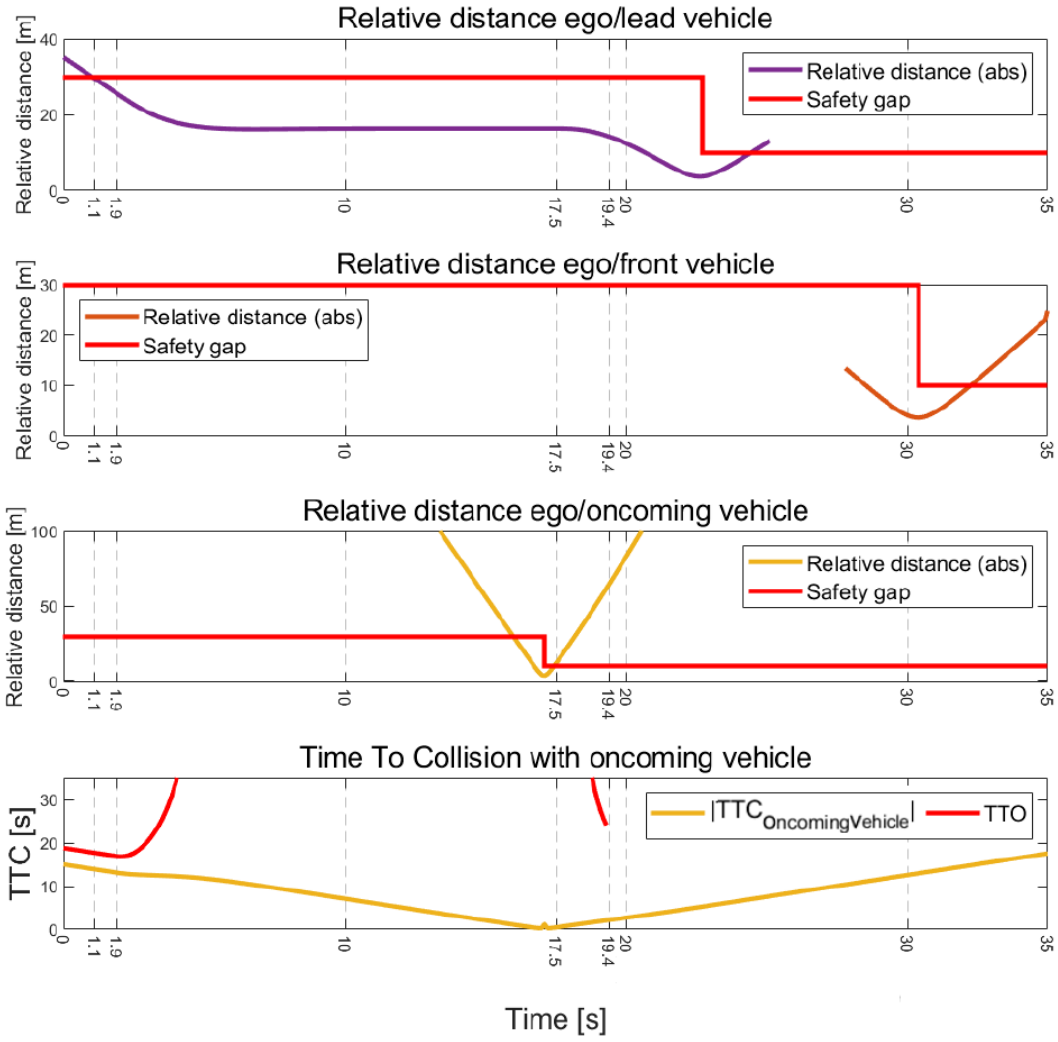

Figure 4.39: MIO vehicle evaluation in the "Overtaking a Line of Vehicles" scenario.

The following is a point-by-point analysis of the results obtained from the simulation, commenting on the most relevant moments in chronological order.

- t = 1.1 s: The ego vehicle approaches the lead vehicle so much that it exceeds the minimum threshold on the Relative distance. Consistent with what has been seen so far, one would expect the activation of the LC mode in the event of a change in a preferential lane; however, the newly introduced CheckLeadExtension function prevents its activation. This happens because the value of the TTC calculated with respect to the oncoming vehicle is lower than the value of the TTO parameter, as shown in the fourth graph in figure 4.39.

- t = 1.9 s: Unable to change its lane and proceeding at a higher speed than the line of vehicles, the ego vehicle switches to LCF mode to avoid a collision. This type of mode change occurs following evaluations carried out on the cost parameter associated with each terminal state.
- t = 17.1 s: The oncoming vehicle passes the ego vehicle with respect to the curvilinear coordinate s; as can be seen from the slope variation of the Relative distance reported in the third graph in figure 4.39. Therefore, the LC mode is introduced in the set of terminal state alternatives.
- t = 17.5 s: The value of the Relative distance calculated with respect to the oncoming vehicle assumes a value higher than the rearSafetyGap threshold, this implies the absence of Unsafe vehicles in the left lane. At the current time, the new trajectory segment is planned, enabling the LC mode; available since the oncoming vehicle has already passed.
- t = 19.4 s: The ego vehicle reaches the left lane, as can be seen from the trend of the current lane in figure 4.38. This coincides with the preferential lane, consequently the vehicle enters CC mode when a new trajectory segment is planned at t = 20.0 s.
- t = 30.4 s: The ego vehicle overtakes the front vehicle with respect to the reference curve.
- t = 35 s: End of simulation.

In conclusion, it is possible to state that the introduction of the CheckLeadExtension function has had a positive response, making the model capable of adapting to scenarios that present critical issues such as those illustrated in this case study.

4.5 Controller and Vehicle Model Test Simulation

In this paragraph, all the simulations conducted during the creation process of the overall model have been reported. The results obtained have been commented at the end of the discussion, with the aim of avoiding presenting related concepts before having introduced all the necessary simulation and data analysis techniques, to facilitate their understanding. The topics covered include:

- Verification of the correct functioning of the overall model in a scenario characterized by a low performance requirement.
- Introduction of the drift stiffness characteristics $C_{\alpha,\mathrm{F/R}}$ in the integrated model within the adaptive MPC controller.

4.5.1 Test 1 - Verification of the overall model functionality in a scenario with low performance requirements

After completing the coupling of all the parts of the complete model, various tests were carried out with low-critical scenarios to verify the actual functioning and the correct exchange of information between the parts. The scenario examined is that of a double lane change, in which the ego vehicle overtakes the lead vehicle. The maneuver is performed at maximum lateral accelerations that are not too high. This last characteristic was sought because proposing a scenario with an impactful criticality could have produced alterations in the signals that would not allow us to understand if there were coupling problems between the parts of the model. Figure 4.40 shows the comparison between the reference trajectory and the one followed by the ego vehicle.

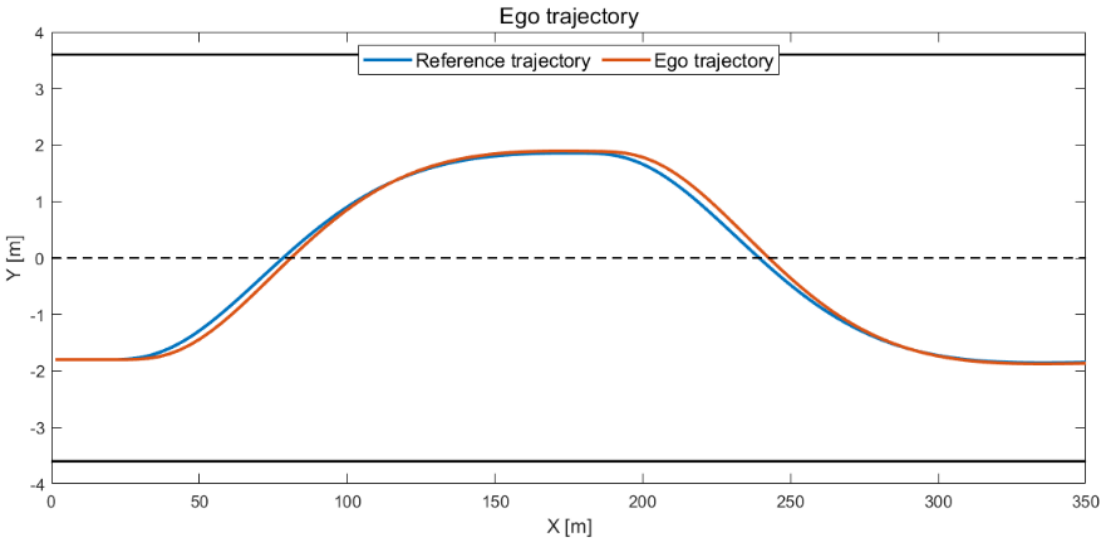

Figure 4.40: Comparison of reference/actual ego vehicle trajectories.

An excellent correspondence between the two trajectories is immediately noticeable. The trend of the position errors, expressed in m, and orientation over time, expressed in degrees, is reported in Figure 4.41.

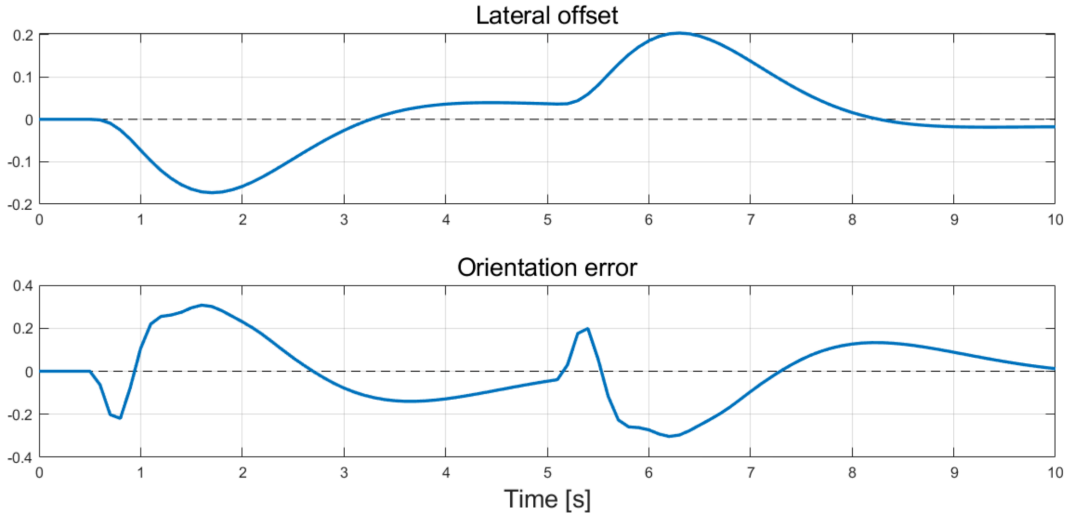

Figure 4.41: Trend of lateral and angular errors of the actual vehicle trajectory ego compared to the reference trajectory.

The following Figure 4.42 shows the outputs produced by the controller during the simulation, including the steering angle, expressed in degrees, and longitudinal acceleration, expressed in $\mathrm{m/s^2}$.

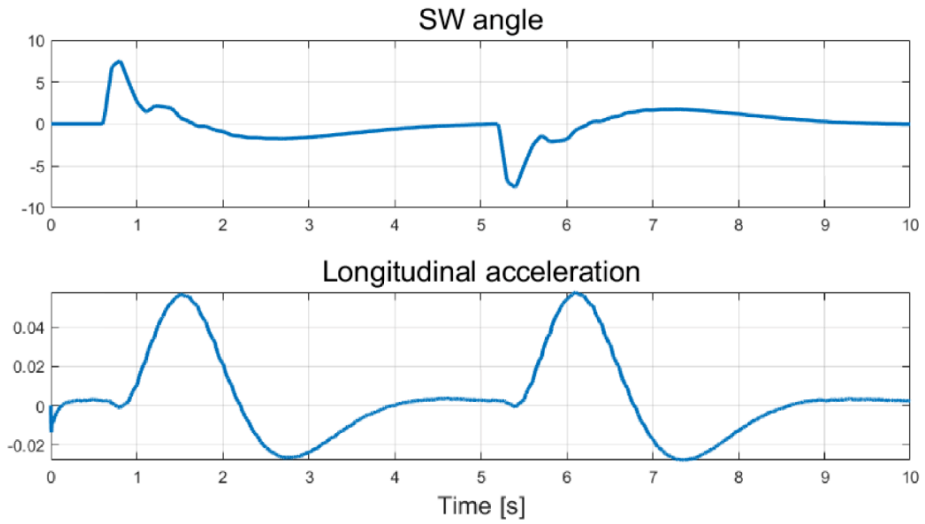

Figure 4.42: Trend of the outputs produced by the controller over time.

The trend of the main quantities describing the motion of the ego vehicle and the respective references are illustrated in Figure 4.43. The low-performance requirement is evident by observing the blue curves in each graph, which correspond to the reference trajectory.

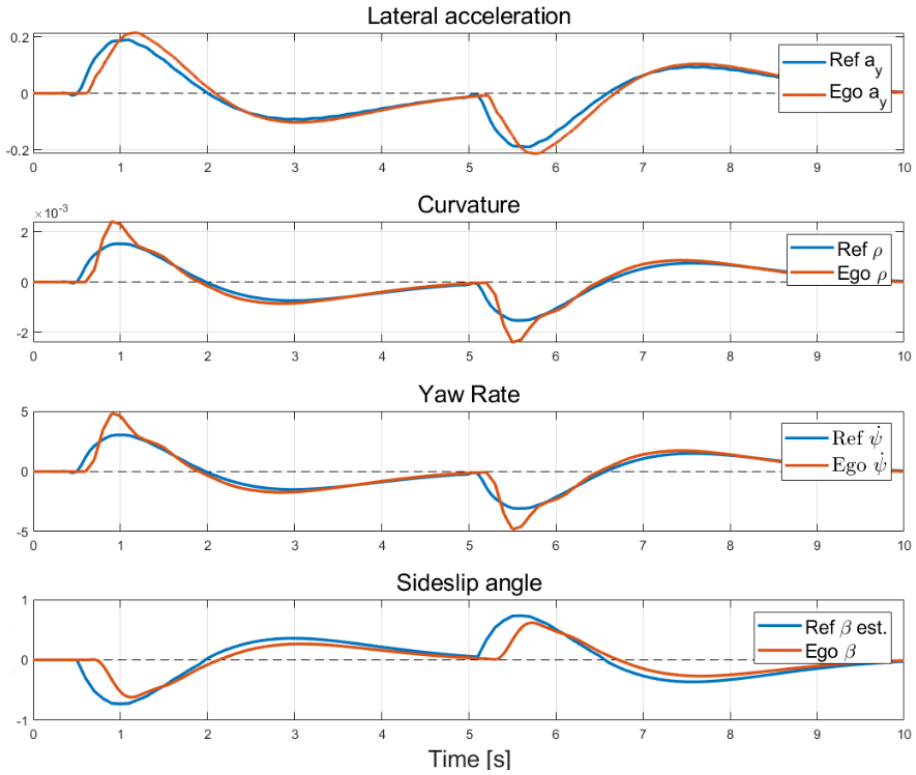

Figure 4.43: Trend of the quantities describing the motion of the ego vehicle and the respective references. Units: [Lateral acceleration: g], [Curvature: 1/m], [Yaw rate: °/s], [Sideslip angle: °].

4.5.2 Test 2 - Incorporation of cornering stiffness characteristics into the MPC controller

In the third chapter, in paragraph dedicated to the controller, the modifications introduced on the model integrated in the controller were exposed from a theoretical point of view. To improve the adaptivity of the model, the $C_{\alpha,F/R}(a_y)$ maps were inserted, capable of providing the drift stiffness values corresponding to the current lateral acceleration value. The principles and limits that accompany this choice were exposed in paragraph 3.1.7.

To test the effects of the introduced modification, a comparison was performed between two simulations carried out respectively with constant drift stiffnesses and with drift stiffnesses as a function of lateral acceleration. The scenario adopted is a straight road on which an ego vehicle and a lead vehicle travel. The maneuver performed by the ego vehicle is a double lane change, whose maximum required lateral acceleration is equal to $a_{y,max} = 0.75$ g.

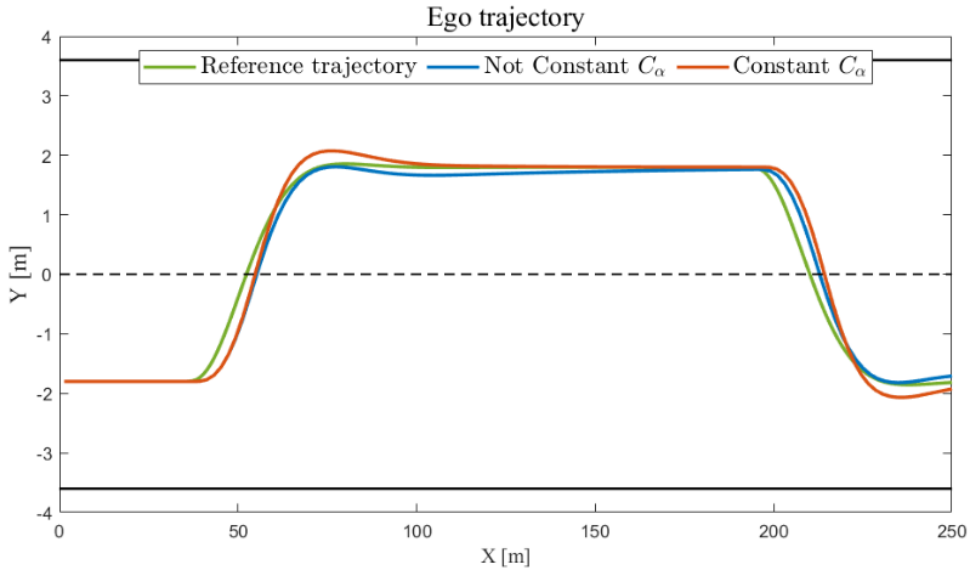

Figure 4.44: Comparison of ego vehicle trajectories with constant/variable Cα.

Figure 4.44 shows a comparison between the actual trajectories of the ego vehicle and the trajectory defined as reference. In blue is the actual trajectory obtained after inserting the drift stiffness characteristics into the controller-integrated model, in orange is the one obtained with constant drift stiffnesses. The reference trajectory was designed to fall within

the range in which the results of a linear and a non-linear tire model are different due to large drift angles. As expected, the blue trajectory is more faithful to the reference one.

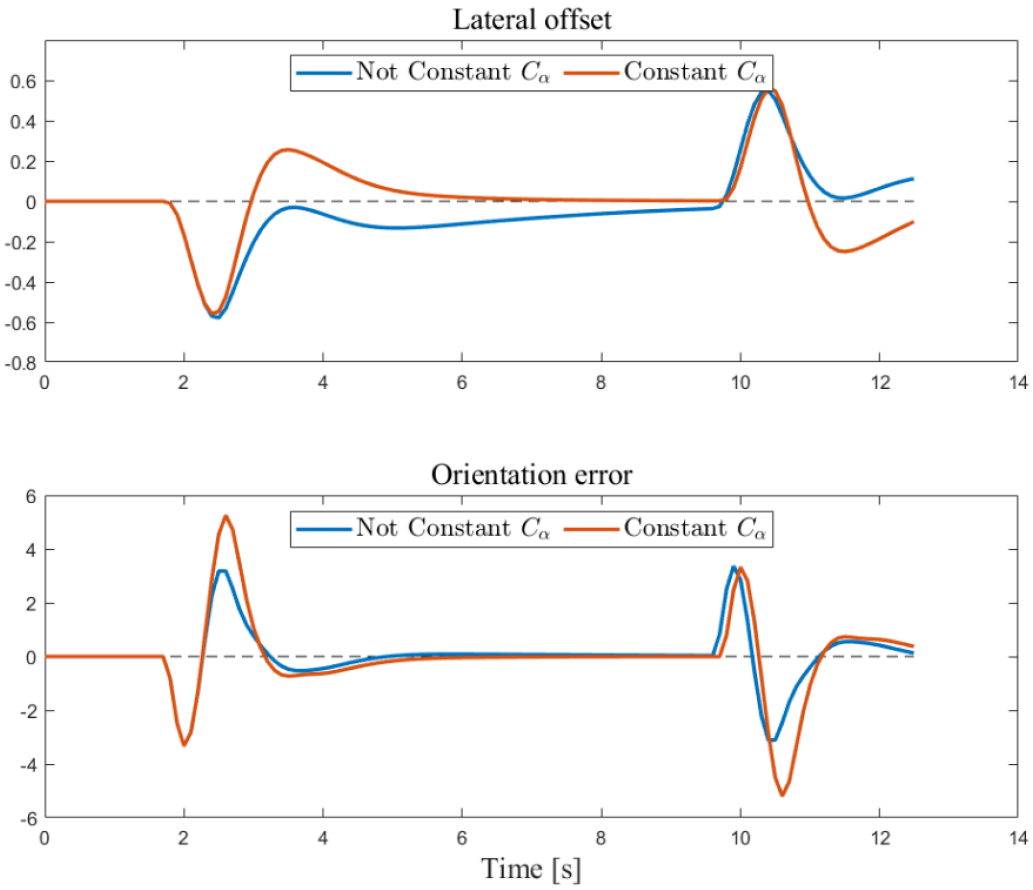

Figure 4.45: Comparison of trajectory errors with constant/variable Cα.

Figure 4.45 shows a comparison between the time trends of the position, expressed in m, and orientation errors, expressed in degrees,. The model with variable drift stiffnesses implements a faster correction on the yaw angle, thus allowing to considerably reduce the error associated with it. However, although in the final phase of the lane change maneuver the model with variable stiffnesses has a significantly smaller lateral deviation, the rapid correction of the steering angle causes the ego vehicle to travel almost parallel to the reference trajectory, with a residual lateral error that is compensated after a not insignificant time period.

Conclusions

In this thesis, the characteristics of an advanced assistance system capable of: planning the correct reference trajectory, establishing the driving strategy to be implemented and, if deemed appropriate, performing an overtaking maneuver were analyzed. First, the discussion exposed the principles on which this system is based from a purely theoretical point of view. On these, a complex and complete model was built in a MatLab & Simulink simulation environment, which is made up of three main components:

- Trajectory planner: able to analyze the surrounding scenario, the vehicle parameters and consequently establish the reference trajectory.
- Controller: adaptive and predictive type, which minimizes orientation and positioning errors by predicting the future states of the ego vehicle, carried out through an integrated vehicle model.
- Vehicle model: Double Track type, with roll and non-linear tire model (Pacejka's magic formula model).

In the most advanced phase of the study, a range of scenarios was proposed, then used to perform simulations with the presented model. They have highlighted some of the critical issues of the assistance system, leading to the development of strategies to be implemented to make it capable of recognizing such critical issues and providing an adequate response.

The strategies implemented have made the assistance system capable of:

- Performing overtaking maneuvers with a vehicle traveling in the opposite direction.
- Performing overtaking maneuvers against a line of vehicles, even in the presence of an additional vehicle traveling in the opposite direction.
- Positioning the ego vehicle correctly within the carriageway, making the assistance system consistent with the rules of the Road Traffic Regulations.
- Implementing a rapid lane change strategy in emergency cases, having introduced a

check on the stability of the vehicle, based on non-linear dynamics techniques and graphic approaches, which identifies the maximum performance requirement.

5.1 Future Developments

During the discussion, the limits of the developed system were highlighted, if identified, which can be considered starting points for a subsequent study phase. Specifically, the developments that would bring significant benefits in terms of potential and functionality of the system are the following:

- Implementation of vehicle sensors within the model used, so as to eliminate the V2X hypothesis and make the system usable not exclusively with advanced infrastructures.
- Updating the controller used, in order to make the model integrated into it more consistent with the vehicle model used.
- Characterization of the signal alteration made by the controller, in order to make a more accurate estimate of the vehicle parameters and make the checks on the set of alternative trajectories more precise.
- Development of new strategies, such as to expand the range of scenarios correctly analyzed by the system; in particular, strategies are needed to be implemented in emergency cases, such as immediate stopping of the vehicle.

References

[1] ACI - ISTAT, "Report incidenti stradali 2023," 2024.

[2] Maserati, "Surround View Camera System - Safety," [Online]. Available: https://www.maserati.com/global/en/ownership/maserati-manuals/safety/surround-view-camera.

[3] S. I. J3016_202104, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," 2021.

[4] "Regulation (EU) 2019/2144 of the European Parliament and of the Council of 27 November 2019," Official Journal of the European Union.

[5] J. Z. S. K. a. S. T. Moritz Werling, "Optimal Trajectory Generation for Dynamic Street Scenarios in a Frenet Frame," IEEE International Conference on Robotics and Automation, pp. 987-983, 2010.

[6] R. A. K. a. A. Kelly, "Differentially constrained mobile robot motion planning in state lattices. Journal of Field Robotics (JFR)," Field Robotics, 2009.

[7] MathWorks, "Convert global states to Frénet states," [Online]. Available: https://it.mathworks.com/help/nav/ref/referencepathfrenet.global2frenet.html.

[8] X. T. Y. Q. Y. H. a. E. H. Zhewei Zhu, "A Survey of Lateral Stability Criterion and Control Application for Autonomous Vehicles," IEEE.

[9] N. X. H. C. Y. H. a. B. Z. Zhongliang Han, "Energy-efficient control of electric

vehicles based on linear quadratic," Applied Energy.

[10] MathWorks, "Dynamic capsule-based obstacle list," [Online]. Available: https://it.mathworks.com/help/nav/ref/dynamiccapsulelist.html.

[11] MathWorks, "Automated Driving Toolbox," [Online]. Available: https://it.mathworks.com/help/driving/index.html.

[12] MarthWorks, "Driving Scenario Designer," [Online]. Available: https://it.mathworks.com/help/driving/ref/drivingscenariodesigner-app.html.

[13] MathWorks, "Navigation Toolbox," [Online]. Available: https://it.mathworks.com/help/nav/index.html.

[14] MathWorks, "Highway Lane Change Planner and Controller," [Online]. Available: https://it.mathworks.com/help/driving/ug/highway-lane-change-planner-and-controller.html.

[15] MathWorks, "Scenario Reader," [Online]. Available: https://it.mathworks.com/help/driving/ref/scenarioreader.html.

[16] MathWorks, "Path Following Control System," [Online]. Available: https://it.mathworks.com/help/mpc/ref/pathfollowingcontrolsystem.html.

[17] MathWorks, "What is Model Predictive Control," [Online]. Available: https://it.mathworks.com/help/mpc/gs/what-is-mpc.html.

[18] MathWorks, "Adaptive MPC," [Online]. Available: https://it.mathworks.com/help/mpc/ug/adaptive-mpc.html.

[19] MathWorks, "Optimization Problem," [Online]. Available: https://it.mathworks.com/help/mpc/ug/optimization-problem.html.

[20] P. H. a. L. L. Bakker E., "A new tire model with an application in vehicle dynamics studies," SAE Trans. J. Passenger Cars, 1989.

[21] P. W. a. P. M. G. Schuring D. J., "The paper-tire concept: A way to optimize tire force and moment properties," SAE Paper No. 970557., 1997.