POLITECNICO MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

Executive Summary of the Thesis

Modelling and Validation of an Adaptive Cruise Control using a high fidelity Dynamic Driving Simulator

Laurea Magistrale in Mechanical Engineering - Ingegneria Meccanica

Author: Mattia Levantaci

Advisor: Prof. Edoardo Sabbioni

Co-advisor: Michele Asperti

Academic year: 2021-2022

---

1. Introduction

ADAS (Advanced Driver Assistance Systems) are one of the most promising technologies currently under development, since they are decreasing collisions and fatalities on the road. Their aim is to help, not substitute, the human driver during driving and parking maneuvers. Their functioning is based on the coupling of sensors and control algorithms. The former are used to retrieve information from the environment that the latter use to control the vehicle. The number of ADAS on the market is quite big, ranging from Cruise Control to Emergency Braking (EB) , Lane Keeping Assistant (LKA) and many others. One of the most common assistance system is the Adaptive Cruise Control(ACC). This system gained its popularity thank to its comfort, due to the reduced driver effort on long drivings, and its safety, provided by the sensors monitoring. The ACC allows to maintain a certain speed selected by the driver, without requiring any throttle input from him, while also enforcing a safety distance from a vehicle, or any obstacle, detected in the same lane as the vehicle where the ACC is implemented, called ego vehicle in the following discussion. The ACC system relies on sensors, commonly radars and cameras, to retrieve information from the environment and act consequently. The Adaptive Cruise Control is a renowned and established system, however it still presents some issues. An issue investigated by many authors is the alternation between the control modes. Indeed, the ACC system must pass from controlling the vehicle speed, to keep the value desired by the driver, to control the distance from an obstacle detected in front of the vehicle, in the most smooth and comfortable way. [1] implements a switch logic to alternate these two modes without chattering, while [2] defines three modes, to control respectively speed, relative distance between the obstacle in the front and the ego vehicle, as well as the relative speed between them. Another issue is the optimal definition of the relative distance between the ego vehicle and the obstacles. Many different spacing policies can be defined indeed[3], some of them focuses on comfort while others more on safety. This thesis proposes the design of an Adaptive Cruise Control system, where these issues have been investigated with the use of a high fidelity driving simulator, specifically a DiM400 cable-driven Dynamic Simulator. This simulator is the newest addition to the VI-Grade driving simulators, having a unique cable driven system for the lower stage, coupled with a 6 degrees-of-freedom upper stage.

2. Adaptive Cruise Control model

The Adaptive Cruise Control has been modeled in Simulink, which is ready to be compiled on a Dynamic Driving Simulator, it co-simulates with the driving simulator softwares. The vehicle model is defined in VI-CarRealTime, that integrates the Equations of Motion (EoMs) during the simulations. This model is a simplified 14 DoFs vehicle model, which namely are 6 for the chassis and 2 for each wheel. The vehicle used in the tests is a *CompactCar*, that represents a front wheel drive vehicle, with a traditional internal combustion engine. Its main data are reported in Table 1.

	Value	Unit
Mass	1383	kg
Wheelbase	2577.4	mm
Front track	1521.5	mm
Rear track	1555.9	mm
CG height	563.9	mm
Roll inertia	318.96	$kg * m^2$
Pitch inertia	1252.66	$kg * m^2$
Yaw inertia	1217.10	$kg * m^2$

Table 1: Vehicle model data.

VI-WorldSim has been used to build and import scenarios for the simulations. This software, developed by VI-Grade, enables to select an environment and populate it with actors, like vehicles or pedestrians, or any other inanimated obstacle. In VI-WorldSim the behaviour of the actors is editable as well, hence any vehicle can wander or can be triggered to perform a determined action or can follow a specific trajectory. To test the ACC system a four-lane highway environment has been selected. The sensors attached to the vehicle and their settings are defined here as well. In the designed ACC system a "Virtual Camera" is used. This sensor represents an ADAS camera, capable of retrieving the position, size and orientation of the targets, as well as the road lanes. Thus, this sensor sends to the ACC control system the position and speed of any obstacle present in the same lane as the ego vehicle. The sensor settings for the ACC application are reported in Figure 1.

Figure 1: Virtual Camera settings.

Regarding the definition of the desired distance, its expression is reported in Equation 1. The desired relative distance is composed by three components. The first derives from a constant time-headway spacing policy [3], where t_{hw} is the time-headway and v_{ego} is the ego vehicle speed. Then a constant component has been added, named d_{offset}, to have a not-null relative distance for situations where the speed reaches zero. The last contribution is given by the relative speed between the obstacle and the ego vehicle, represented by t_{rs}, which is a time constant like the time-headway, and v_{rel}, available from the Virtual Camera.

$$d_{des} = t_{hw}v_{ego} + t_{rs}v_{rel} + d_{offset} \quad (1)$$

This definition brings to the best possible results in terms of safety and comfort, as it accounts for the vehicle absolute speed and the state of the obstacles in the scenario.

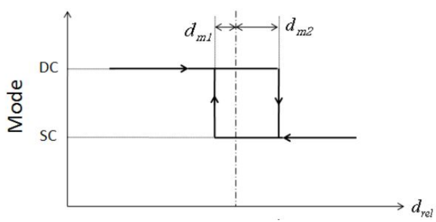


Figure 2: Mode switching logic.

Based on the relative distance between the ego vehicle and the lead one, the ACC presents three different modes, representing three different actions from the controllers, reported in Table 2. To precisely define the activation of each mode, marginal distances have been defined, as represented in Figure 2. If no obstacle is detected, or if it is far enough, the mode is "Cruise", hence the ACC system maintains the speed required by the driver. In this mode the driver can also change speed using two specific buttons, increasing or reducing the desired velocity value each time one of them is pressed. If the obstacle is too close, the mode switches to "Safety-critical". In this mode the controller acts on the relative distance, to brake and avoid hitting an obstacle on the road. When the lead vehicle is in the region between d_{m1} and d_{m2}, the system is set in "Follow" mode. In this mode the controller acts on the error defined in Equation 2, to keep the same speed as the lead vehicle, and maintain the desired distance from it.

$$e_{follow} = v_{rel} + w_d(d_{des} - d_{rel}) \qquad (2)$$

The error is given by two contributions, the relative speed v_{rel} and the error on the desired distance, weighted using w_d, a constant parameter. Of course, if the lead vehicle starts accelerating too much, and overcomes the speed desired by the driver, the ACC system stops following it.

		Mode
$d_{rel} \leq d_{des} - d_{m1}$		Safety-Critical
$d_{m1} \leq d_{rel} \leq d_{m2}$	$v_{lead} \leq v_{set}$	Follow
	$v_{lead} > v_{set}$	Cruise
$d_{rel} > d_{des} + d_{m2}$		Cruise

Table 2: Decision making for the ACC system mode.

2.1. ACC controllers

The Adaptive Cruise Control system consists of three different controllers in parallel, one for each mode. Their output is a longitudinal force, that multiplied by the rolling radius, gives the total theoretical wheel torque that has to be applied to the vehicle to fulfill the controller scope. A FeedForward (FF) contribution has been added, that gives a fast correcting action, since there is no need to wait for an error to build up before reacting. Two FF controllers have been defined, one active for "Cruise" mode and one for "Follow" mode. When the "Safety-critical" mode is active, no FF force is provided to the system. The FF contribution output is a longitudinal force, that aims to balance the resistant forces that in this case are the inertial, aerodynamic and rolling ones.

To attenuate the disturbances in the system FeedBack (FB) controllers have been added as well. This contribution is given by PI(D) controllers. In "Cruise" mode the controller is a PID, with constant gains. Instead, the "Follow" mode controller is a PI, since the derivative contribution introduced some disturbances in the output signal. Its gains are function of the error on the relative distance between ego and lead vehicles. The "Safety-critical" controller is again a PID, with variable gains as well. For this mode a more complex gains scheduling is defined, since this controller has to regulate all possible critical situations, as well as most of the braking actions. Thus a fuzzy logic controller has been used to define the controller gains, to implement a set of rules to account for any possible condition. The fuzzy control rules for the ACC are composed of 25 simple rules. The two inputs are the relative distance error, and the relative speed with respect to the lead vehicle. The distance error input is transported in 5 linguistic variables, namely VF(very far), F(far), NF(not far), C(close) and VC(very close), ranging from 5 meters, for VF, to 50 meters, for VC. The relative speed input is defined as NL(negative large), NM(negative medium), ZE(close to zero), PM(positive medium) and PL(positive large). Its value ranges from -10 to 10 m/s. With the system sign conventions, a PL relative velocity means that the lead vehicle is faster than the driven one. The output is the controller proportional gain, based on the defined set of rules.

The selected vehicle model has an internal combustion engine, hence the requested torque by the controllers is split between the engine and the brakes. Positive torque requests are regulated by the engine. Instead for negative torque, the residual demand that cannot be provided by the engine is requested to the brakes.

3. Offline tests

To test the functionality of the ACC system, other agents are added in the scenario, to verify the interaction of the ego vehicle with other cars in the environment. The system has to be tested at least in some basic maneuvers, to ensure that comfort and safety requirements are respected, and that the sensors work properly. The first test is in the simplest possible scenario, where a vehicle is traveling in the same lane as the ego, proceeding at a lower speed. To simplify the discussion, this scenario is called maneuver A and its schematics is reported in Figure 3.

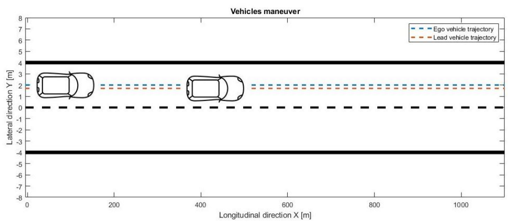
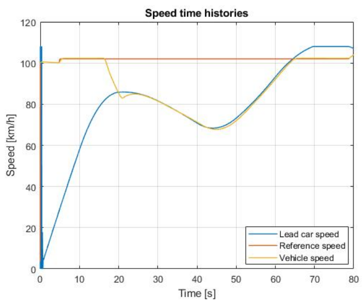
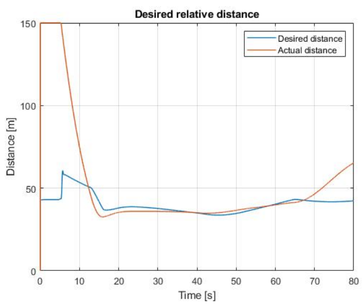


Figure 3: Schematics of maneuver A.

The speed profiles, reported in Figure 4a, show that the ego vehicle manages to brake and follow the lead one, without oscillations, leaving the desired distance defined in the controller, as shown in Figure 4b.



(a) Speed profiles.



(b) Desired and actual distances between ego and lead vehicle.

Figure 4: ACC test maneuver A.

The next maneuver, named B in Figure 5, represents the case where the lead vehicle overtakes the ego on one side and then changes lane, settling in front of the ego vehicle. A really common situation in highway driving conditions.

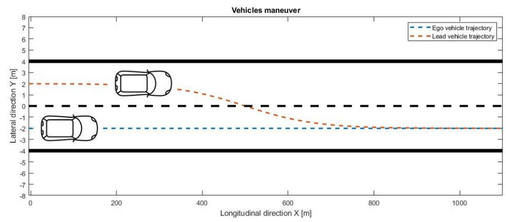


Figure 5: Schematics of maneuver B.

This maneuver tests the virtual camera functioning as well. Indeed, the sensor has to recognize properly the lanes, and when a vehicle moves in and outside them. From Figure 6 it can be seen that the vehicle manages to brake and then follow the lead vehicle as in Maneuver A. The braking action is quite strong since the lead vehicle suddenly moves in front of the ego and moves at lower speed as well.

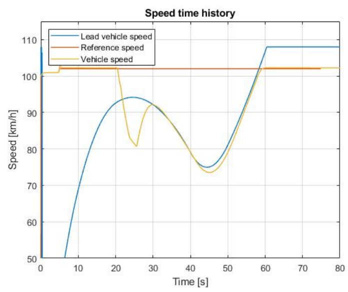


Figure 6: Speed profiles during maneuver B.

The ACC system has been tested in other maneuvers as well, and with different speed profiles, to verify that the controllers respond accordingly to the situation. Once the ACC system has been verified to work properly in any maneuver, it can be tested on the dynamic driving simulator.

4. Tests on Dynamic Driving Simulator

The tests on the Dynamic Driving Simulator are crucial so that drivers can understand if the system is comfortable or not. The tests are composed of three driving simulations, in the first ones the drivers can drive freely, while in the second and third one the testers are asked to activate the ACC system. In the second test the ACC has a lower gains setup, while in the third one the ACC presents higher gains. To have comparable results, the lead vehicles trajectory and speed are equal in all tests, as well as the set speed to test the ACC, close to 120 km/h. The system has been tested by 19 people, both males and females, from 24 to 46 years old, that have been asked to reply to 30 questions. The testers give their feedback on a 1 to 5 scale, except for questions 16 and 25, where the percentage of positive answers (YES) is reported. Among these people, seven already used an Adaptive Cruise Control system in a real vehicle, and twelve tried a driving simulator. It is curious to see that of the seven people that tried an ACC, only 2 replied that they commonly use it on the highway, while the other five still prefer to manually drive. After these preliminary questions, the testers are asked to evaluate the simulation and the driving simulator realism, as reported in Table 3.

ID	Question	Score
6	**Rate how realistic the simulation is**	3.84
7	**Rate the driving simulator realism (cockpit, pedals, steering...)**	4.11
8	**Rate the driving simulator scenario realism (speed and distance perception, sounds...)**	4.00

Table 3: Answers on driving simulator realism.

The answers in Table 4 represents the evaluation of the two ACC systems, labeled as 2 and 3. The system with softer gains shows better scores both from comfort, as expected, and safety point of view. It seems that the testers felt stronger braking as dangerous rather than evaluating the system capable to brake more and avoid collisions. Among the 19 testers, 7 replied that they would prefer to drive manually rather than use the test 3 system, while for test 2 system only 2 people told they would prefer to drive manually. These two people replied NO to both questions 16 and 25, showing that they prefer to driver manually rather than using an assistance system like the Adaptive Cruise Control

ID	Question	2	3
9/18	**Rate the ACC model from a safety point of view (distance from other vehicles, sounds...)**	4.42	4
10/19	**How do you consider the distance to the leading vehicle kept by the ACC model? (1 too close, 3 adequate, 5 too far)**	2.95	2.89
11/20	**Rate the ACC model from a comfort point of view (accelerations, speed. . .)**	4.42	3.74
12/21	**Rate the ACC model from a comfort point of view during braking**	3.95	3.11
13/22	**Rate the ACC model from a comfort point of view during accelerations**	4.47	4.05
14/23	**How much does the system reduce the driving effort?**	4.11	4
15/24	**If you normally use an ACC, rate how similar this model is**	3.67	3.2
16/25	**Would you use the ACC model in the proposed scenario instead of manually driving?**	89%	63%
17/26	**Rate how willingly you would use the model in a real driving situation**	3.79	3.42

Table 4: Answers to ACC systems evaluation.

The last questions are meant to compare the three systems, shown in Figure 7. Once again the ACC in test 2 is the preferred one, even compared to the first test, where the drivers could drive freely. The ACC with lower gains has been considered the system more similar to the testers driving style, and 18 out of the 19 drivers would prefer this system in their own car.

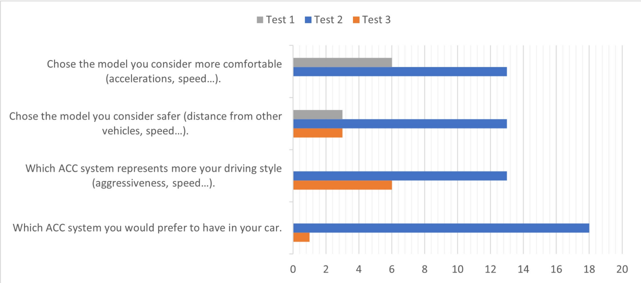


Figure 7: Answers to questions from 27 to 30.

The significant data from the simulations have been analysed as well. The average is not computed on the complete driving, but the considered data are only those for which the ego vehicle has a car in front of it. Hence, these results represent the vehicle dynamics while following another vehicle. On average, the accelerations are similar in all three tests, while the ACC systems brake more than the drivers, as shown in Table 5. This because the Adaptive Cruise Control forces the two vehicles to stay at a defined distance, so when the lead vehicle moves in front of the ego vehicle it brakes quite a lot.

	1	**2**	**3**
Average acceleration $[m/s^2]$	0.32	0.28	0.30
Maximum acceleration $[m/s^2]$	1.27	1.43	1.45
Average deceleration $[m/s^2]$	0.43	0.53	0.58
Maximum deceleration $[m/s^2]$	1.79	4.52	4.90

Table 5: Average results during the simulations.

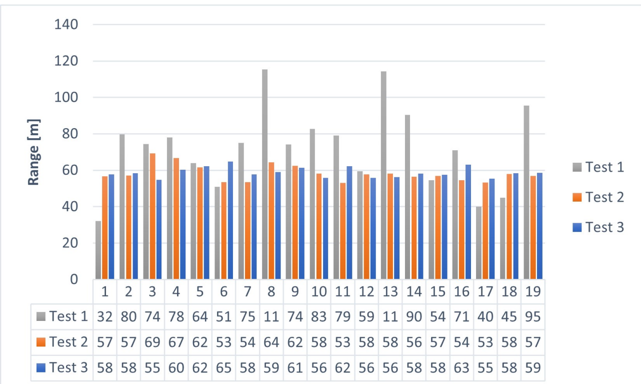


Figure 8: Average ego vehicle distance to the leading car for the different tests.

In Figure 8 it can be seen that both systems leave comparable relative distances between ego and lead vehicles, while different driving styles are represented from test 1 results. However, testers evaluate the relative distance as adequate on average, evident from answers to questions 10 and 19.

At last, speed data are analysed. Results in Table 6 show that the average speed is similar in all tests, since the ego vehicle is forced to follow the lead ones. However the standard deviations display that by manually driving the vehicle oscillates more, while the ACC systems stabilize the vehicle speed close to the one of the lead vehicle, avoiding uncomfortable speed oscillations.

	1	**2**	**3**
Avg speed $[km/h]$	112.36	111.69	111.33
Std. dev. $[km/h]$	10.33	8.46	8.78

Table 6: Average speed and standard deviation for the different tests.

5. Conclusions

From offline simulations the mode switches result stable and smooth, and the system reacts properly in all the tested maneuvers. The Virtual Camera signals are reliable, since the obstacle and lane detection do not bring any issue to the simulation. To prove the comfort of the designed ACC system, a high fidelity driving simulator has been used. Testers have provided their feedback about two ACC systems tuning variants and also about the driving simulator itself. The results show that the system satisfies safety and comfort requirements of a state of the art Adaptive Cruise Control system.

References

[1] M. Tomizuka Kim S. G. Mode switching and smooth motion generation for adaptive cruise control systems by a virtual lead vehicle. *IFAC Symposium on Transportation Systems*, pages 490–496, 2009.

[2] Vakili Sasan. Design and formal verification of an adaptive cruise control plus (acc+) system. Master's thesis, McMaster University, Hamilton, Ontario, Canada, 2014.

[3] Liu Y. Wu C., Xu Z. Spacing policies for adaptive cruise control: A survey. *IEEE Access*, pages 50149–50162, 2020.

POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Modelling and Validation of an Adaptive Cruise Control using a high fidelity Dynamic Driving Simulator

Tesi di Laurea Magistrale in
Mechanical Engineering - Ingegneria Meccanica

Author: **Mattia Levantaci**

Student ID: 945263
Advisor: Prof. Edoardo Sabbioni
Co-advisors: Ing. Michele Asperti
Academic Year: 2021-22

Abstract

Humans have always sought to foster and improve their way of life, and nowadays one of the most significant variables affecting living standards is transportation. The automotive industry has been concentrating on two key areas since the creation of the first cars: boosting effectiveness to cut expenses, and increase power production (taking environmental concerns as well), attempting to minimize accidents, fatalities, and injuries. These two goals created new requirements that are now part of the process of new car development through simulations. Engineers and scientists are nowadays able to digitally simulate what would occur in the real world, to speed up testing and prevent any mishaps during the design process, which, as a result, reduces development costs and time to market.

Driving simulators are adopted in a variety of contexts. They are compatible with Driver in the Loop and the so-called Software/Hardware In the Loop testing. This is done by means of a virtual environment that replicates systems under development, simply using computer models. HiL (Hardware in the Loop), on the other hand, involves some tangible components in the simulation environment, such as sensors, actuators, and so forth. Costs increase as a result, although these tests may occasionally be required. ADAS (Advanced Driver Assistance Systems) are one of the most promising technology in development, since they are increasing and growing prominence in the automotive industry and decreasing collisions and fatalities on the road.

Software programs that can create test scenarios are essential to the entire simulator industry. A simulation scenario should take into account a number of factors, including:

- Vehicle editing: the test vehicle should be created with as much accuracy as possible, from the choice of fuel source to the specification of vehicle dynamics parameters, and so forth.
- Editing the simulation environment, both from a purely graphic standpoint (urban, extra-urban, highway, etc.) and from the perspective of traffic (the addition of Non Played Characters, such as other cars or pedestrians, may be required).

- The method of acquiring simulation outputs, both in real-time and after the simulations, is essential since they are relevant and extremely important. So how they can be logged (e.g., if it requires additional machines or software).

Modern cars come with a variety of driver assistance features that improve safety and reduce driver fatigue. The Adaptive Cruise Control (ACC) system has been implemented as a result of recent advances in sensor technology. This system is defined to regulate the vehicle speed. If there is no lead vehicle or if the lead vehicle is moving at a higher speed than the driver had intended, the ACC system should keep a safe distance from it while also maintaining the driver's planned pace. It should also respond swiftly if the lead car moves and enters in the same lane as the driver. This system though, still shows some issues regarding its safety and comfort. This thesis looks into the complete creation of such advanced system and its testing and validation, using a high fidelity dynamic driving simulator. The work comprehends the issues statement, the modelling of the control system and the testing using the driving simulator software. Finally, the system is tested on the dynamic driving simulator by some testers.

Keywords: ADAS, Adaptive Cruise Control, Driving Simulator

Abstract in lingua italiana

L'umanità ha sempre cercato di migliorare il proprio stile di vita, e oggigiorno una delle principali variabili che modificaano gli standard di vita è senza dubbio quella del trasporto. L'industria automobilistica si è sempre concentrata su due aspetti chiave dalla creazione del primo veicolo: aumentare l'efficienza per ridurre le spese, e aumentare la capacità produttiva (anche per preoccupazioni sull'ambiente), cercando di minimizzare incidenti, fatalità e incidenti. Questi due obbiettivi hanno creato nuovi requisiti che oggi fanno parte del processo produttivo di auto con l'aiuto di simulazioni. Ingegneri e scienziati sono oramai capaci di simulare digitalmente quello che accadrebbe nel mondo reale, per velocizzare i test e prevenire errori durante la fase di progetto, che riduce costi di produzione e tempo di sviluppo.

Simulatori di guida sono utilizzati in una molteplicità di contesti. Sono compatibili con simulazioni Diriver in the Loop, e anche con Software/Hardware in the Loop. Questo è possibile utilizzando un ambiente virtuale che replica gli ambienti lavorativi sotto sviluppo, utilizzando semplici modelli al computer. Simulazioni HiL (Hardware in the Loop), d'altra parte, coinvolgono elementi fisici, come sensori, attuatori e così via. I costi quindi aumentano di conseguenza, anche se questi test possono essere indispensabili in alcune situazioni, specialmente in campo automobilistico. Sistemi ADAS (Advanced Driver Assistance System) sono una delle tecnologie più promettenti attualmente in produzione, poichè il loro utilizzo è in aumento e il loro sviluppo è promettente, dato il ridotto numero di collisioni e fatalità in strada.

I software per creare scenari sono essenziali per l'intera industria dei simulatori. Un ambiente simulativo deve considerare molti fattori:

- Modifica del veicolo: la vettura per i test deve essere creata più realistica possibile, dalla scelta del tipo di motore alle specifiche per la sua dinamica, e così via.
- Modifiche all'ambiente della simulazione, sia da un punto di vista grafico (urbano, extra-urbano, autostradale...), sia dal punto di vista del traffico (aggiunta di altri veicoli o pedoni)

- L'acquisizione dei dati, sia in tempo reale che non, è essenziale poichè sono relevanti e imponrtati per lo studio dei risultati. Questo vale sia per come caricare i dati e come questi siano accessibili (per esempio se c'è bisogno di altri software o componenti).

Auto moderne hanno una varietà di sistemi di aiuto alla guida, che aumentano la sicurezza e riducono la fatica del guidatore. Il Cruise Control Adattivo è stato introdotto in commercio dopo i recenti sviluppi nella tecnologia sensoristica. Se non sono presenti veicoli di fronte al guidatore o se questi ultimi sono più veloci, il sistema mantiene la velocità impostata dal pilota, sempre mantenendo una distanza di sicurezza. Se il veicolo davanti si muove nella stessa corsia del pilota il sistema deve rispondere rapidamente per evitare incidenti. Questo sistema tuttavia presenta ancora alcune problematiche, riguardanti il suo comfort e sicurezza. Questa tesi comprende la creazione di questo sistema di assistenza avanzato, e la sua valutazione, usando un simulatore di guida ad alta fedeltà. Questo lavoro comprende la discussione di tali problematiche, il modellaggio del sistema di controllo and il suo testing utilizzando i software del simulatore di guida. Infine, il sistema è stato testato da alcuni tester sul simulaotre di guida dinamico.

Parole chiave: Cruise Control Adattivo, simulatore di guida, ADAS

Contents

1 | Introduction

In this chapter, the main subjects of the Thesis are introduced, to have a better understanding of the models, and the need to develop ADAS systems for safety and comfort reasons. Firstly the process to validate vehicles and ADAS is presented, then the method used in this thesis. Then a focus on the Adaptive Cruise Control system and driving simulators is presented.

1.1. ADAS

Advanced Driving Assistance Systems (ADAS) are one of the most researched and developed topic in the last years in the automotive field. Their aim is to help, not substitute, the human driver during driving and parking maneuvers. Their functioning is based on the coupling of sensors and control algorithms. The former are used to retrieve information from the environment that the latter use to control the vehicle. The number of ADAS on the market is quite big, from Cruise Control, Emergency Braking (EB) , Lane Keeping Assistant (LKA) and many others. Even though none of these systems is meant to provide autonomous driving, if not properly modelled they can easily result in fatal accidents, hence a rigorous study is required to guarantee both safety and comfort of passengers [19]. In Figure 1.1 the typical working regions of some ADAS is reported.

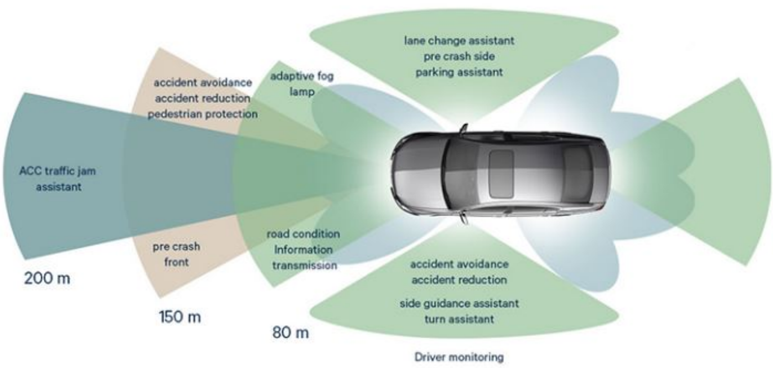


Figure 1.1: Typical ADAS on a vehicle and their working regions.

Before introducing the models, it is essential to understand the need of such assistance systems, and why they are becoming more and more important in the automotive industry. According to the *The National Motor Vehicle Crash Causation Survey* [2], the majority of critical reasons for crash events are driver related, where 41.3% are due to recognition errors, like inadequate surveillance (20.9%) and internal or external distraction (3.7% and 3.6%); 33.4% are related to decision errors, such as aggressive behaviour (1.4%) or misjudgment of gap or other's speed (3.2%); 10.3% are due to performance errors like panic or freezing (0.3%); and finally 7.9% are non-performance errors, like being asleep (3.1%).

It is then clear that most of crashes are associated to avoidable human errors, not due to external non-controllable factors. In these situations ADAS provide life-saving support to the driver, depending on their level of technology, helping humans to recognize and even avoid hazards and threats. The level of damages reduction thanks to some ADAS is visible in Tables 1.1, 1.2 and 1.3 [7]. For these reasons ADAS are becoming more common in any commercial vehicles available in the market.

Automatic Emergency Braking (AEB)

Event	**Percentage reduction**
Front-to-rear crashes	⇓ 50%
Front-to-rear crashes with injuries	⇓ 56%
Claim rates for damage to other vehicles	⇓ 14%
Claim rates for injuries to people in other vehicles	⇓ 24%
Large truck front-to-rear crashes	⇓ 41%

Table 1.1: Real-world benefits from AEB systems.

Lane Departure Warning (LDW)

Event	**Percentage reduction**
Single-vehicle, sideswie and head-on crashes	⇓ 11%
Injury crashes of the same types	⇓ 21%

Table 1.2: Real-world benefits from LDW systems.

Blind Spot Detection (BSD)

Event	Percentage reduction
Lane-change crashes	$\Downarrow$ 14%
Lane-change crashes with injuries	$\Downarrow$ 23%
Claim rates damage to other vehicles	$\Downarrow$ 7%
Claim rates for injuries to people in other vehicles	$\Downarrow$ 9%

Table 1.3: Real-world benefits from BSD systems.

1.1.1. Classification

Based on the capabilities of ADAS, the Society for Automotive Engineers taxonomy (SAE)[6] has defined six levels of vehicle automation driving, from 0 to 5, as reported in Figure 1.2. These levels represent how much the driver has been taken out of the loop to ensure a safer driving. For higher levels the driver will have less control on the vehicle, so the system will greatly help the driving, and vice versa. The maturation and affordability of new technologies have turned the possibility of having self-driving cars into reality, it is already possible to see self-driving functions implemented in commercial cars. In fact many countries are preparing to adapt new laws permitting self-driving vehicles on public roads.[1]

The levels of driving automation are defined as follow, according to SAE taxonomy:

- **Level 0 (No automation)**: the human driver has total control of the vehicle. Most of commercial cars present on the road still have this level of automation. Since systems like emergency braking still represents a Level 0. This is because since they do not assist during the driving, but still give but only in emergency situations.

- **Level 1 (Driver assistance)**: cars with systems that partially help the driver to drive are included in this level, such as Adaptive Cruise Control. In this example the driver still has to control the steering wheel for instance.

- **Level 2 (Partial driving automation)**: a vehicle with more than one ADAS installed is usually in this class, these systems are in control of many aspects of the driving action, even though the driver can always take control of the car.

- **Level 3 (Conditional driving automation)**: this level presents a forward step on a technological point of view. The car is equipped with logics and sensors that can take decisions based on the external environment and substitute some driver commands. A good example is an automatic overtake system, that can evaluate the traffic conditions and, after the approval of the driver, perform the maneuver automatically.
- **Level 4 (High driving automation)**: the vehicles in this level can already be set in self-driving mode, where often the human action is not necessary at all.
- **Level 5 (Full driving automation)**: this level represents the full automation of vehicle driving, where steering wheel and pedals can be removed, since no input is ever required from the driver. These systems are still under development and are one of the most discussed topic in the automotive field.

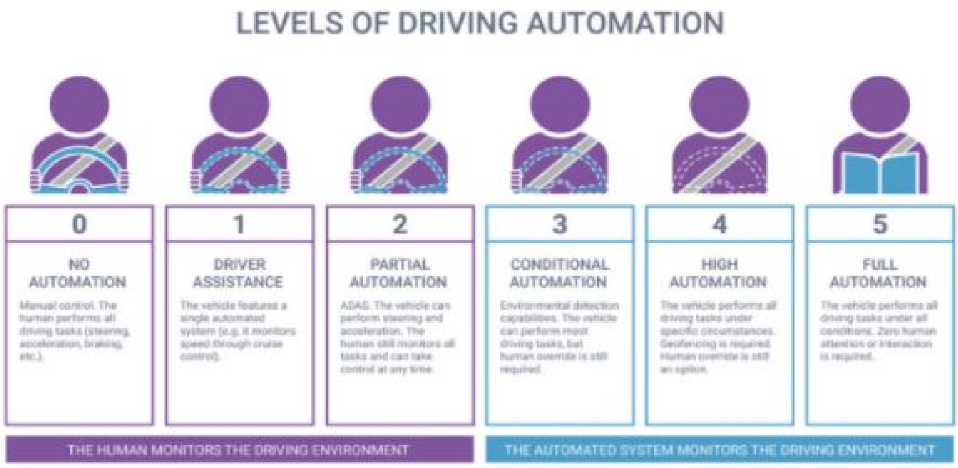


Figure 1.2: Automation driving levels.[19]

1.2. Adaptive Cruise Control

Many driver assistance systems have been introduced since the first automobile, with most of these features which are related to driving automation. One of the most common assistance system is the Cruise Control(CC). This system allows to maintain a certain speed selected by the driver, without requiring any throttle input from him. Nowadays, this system is present in almost any commercial vehicle, and its popularity is due to its comfort, especially in long highway drivings. This because, without this ADAS, the driver is required to continuously push the throttle pedal to keep a constant speed.

Due to the necessity to have a safer system, the Adaptive Cruise Control has been introduced. This system is an advanced version of the conventional CC, having all its features for the driver comfort, but also monitoring the environment to ensure the passengers safety. To do so, the ACC ensures to keep a desired distance from any upfront obstacle. If no impediment is detected, the systems works as a conventional CC. So, if the upfront vehicle is moving at slower speed than the driver the system slows down and keep a suitable distance. The system is designed to never exceed the desired speed set by the driver when activating the Cruise Control, so if the lead vehicle accelerates too much, the system does not follow it anymore.

The ACC system can present extensions such as Stop-and-Go systems and Cooperative Adaptive Cruise Control (CACC). Stop-and-Go is meant to adapt the system for urban driving conditions, where the traditional ACC presents many issues due to low speed driving, and continuous start and stop conditions. With this extension the ego vehicle can completely stop behind the lead vehicle, and then start again, unlike the ACC system. This model also requires more detailed sensor information [22], due to the fact that urban environment is more complex. The system has to react to frequent encounters with other vehicles or pedestrians in front of the ego vehicle, and work properly at low speed.

CACC is an enhanced version of the ACC that exploits Vehicle-to-Vehicle (V2V) or Infrastructure-to-Vehicle (I2V) communication. This system allows the control system to have information about multiple vehicles in a platoon such as velocities, acceleration, throttle and brake commands. This system has the same basic idea as the ACC, but various researches shows that the CACC has the ability to improve traffic flow and string stability [16]. String stability, for a vehicle platoon, means whether spacing errors increases downstream. For instance, a vehicle decelerating in the front of the platoon introduces errors for the other vehicles, and in case the platoon is string unstable, this error magnitude increases going downstream. CACC system is able to retrieve such information with a minimum delay and adapt the control action consequently, while traditional ACC reacts only to the behaviour of the first vehicle ahead of the driver[12].

1.2.1. Performance requirements

As previously said, the Cruise Control system has been introduced for comfort reasons. The Adaptive Cruise Control system is a more complex system, that looks at the environment as well, the system requirements are the following[10]:

- **Stability**: every control system must provide stability, hence proper modelling and fine tuning are essential, since improper controllers can introduce instability into a stable system. In the case of an ACC stability means that the ego vehicle speed and relative distance converge to the desired values. It is important that the transient performance is smooth and stable.
- **Comfort**: ACC acts on the longitudinal dynamics of the ego vehicle, thus accelerations and decelerations are unavoidable. This directly affects the passengers comfort. Parameters such as jerk and pitch rate are monitored as well to ensure a certain comfort level. It is crucial that these parameters are kept as low as possible, while fulfilling the system requirements, to provide a comfortable driving, considering that this system is commonly used for long times. It is also essential that the system behaves as close as possible to the human driving since the driver would feel uncomfortable if the vehicle behaviour is not easily predictable.
- **Safety**: like many other ADAS, safety is the most important requirement. The system must prioritize safety in case of an emergency, like a strong braking of the lead vehicle. Hence the controller has to be capable of correctly retrieving the information from the environment, and, based on that, adapt its action, even not considering the comfort in these specific emergency cases.

1.3. Vehicle testing procedure

Before introducing the validation process used in this research, the official testing and validation procedure for vehicles is presented, regarding both active and passive safety tests.

The European New Car Assessment Program is the entity, supported by the European Union, that defines the testing procedures and the evaluation of the passive safety on cars, with the introduction and use of specific test protocols. It has been founded in December 1996, and from 2009 it deals with vehicle active safety as well[14].

1.3.1. Passive safety tests

Passive safety is evaluated with multiple impact tests, ranging from frontal to lateral, and both against a pole and against people. To estimate the effects of these impacts on the human body, mannequins are placed inside the cars, that can simulate how the human body would react.

Euro NCAP has four evaluation systems, three expressed in stars with different colors, and a fourth method expressed by a points evaluation, to grade the whiplash effect due to a rear impact. The three different categories are the following:

- **Adult Occupant Protection (AOP)**: the test is performed with two mannequins placed in the front seats. The score is given with stars, 0 to 5, and a score, from 0 to 40, looking at the impact consequences.
- **Child Occupant Protection (COP)**: the children safety is evaluated with two mannequins, that replicates two kids around three years old, placed in the rear seats. The score is given with blue stars, from 0 to 5, based on the impact results.
- **Vulnerable Road User Protection (VRU)**: test to evaluate safety for pedestrians. The score is represented by green stars, from 0 to 4, depending on the crash results.

1.3.2. Active safety tests

In 2010, after the introduction of the Autonomous Emergency Braking system on the Volvo XC60 as a standard equipment, Euro NCAP started to perform tests on technologies like this one. This led to the introduction of new rating system and tests such as:

- **Autonomous Emergency Braking Car-to-Car (AEB C2C)**: various tests are performed, where the tested vehicle must avoid hitting the car in the front:

  Car-to-Car-Rear Stationary (CCRs): a vehicle that is going to collide with the back of another one that is standing still.

  Car-to-Car-Rear Moving (CCRm): a vehicle that is going to collide with the back of of another one that is standing still.

  Car-to-Car-Rear Braking (CCRb): a vehicle that is going to collide with the back of of another one that is standing still.

  Car-to-Car-Front Turn-across-path (CCFtap): a vehicle turns inside the path of another one moving at constant speed.

- **Autonomous Emergency Braking Vulnerable Road Users (AEB VRU)**: various tests are performed, where the interaction of the vehicle with human agents (pedestrians, cyclists etc) is tested.

- **Lane Support Systems (LSS)**: many situations are tested for these systems as well, for instance the maneuver, like overtaking, and the speed of the ego vehicle. These tests are meant to evaluate the capacity of the system to stay in the lane markings.

- **Speed Assist (SA)**: tests performed to validate systems like Adaptive Cruise Control, but even simpler systems that help the driver to stay below the speed limit with sound or visual signals.

1.3.3. Role of a Driving Simulator in ADAS Testing

Now that the validation methods have been presented, it is crucial to understand why and how Driving Simulators are so important and efficient for the development and testing ADAS systems.

These systems are based on complex control algorithms, thus their elaborated architecture implies long times for tuning and debugging, and the validation tests require many facilities, such as vehicles, tracks, mannequins etc. Each failure can also bring to the total break of the equipment. For example, in an Emergency Braking system, a malfunction, or even a non-optimal tuning of the controllers, can bring to catastrophic consequences, both for the vehicle and for the driver, in case there is one.

In these situations Driving Simulators show their full potential, both in terms of time and money consumption. Modern technology allows to perform simulations as close as possible to real tests. Driving simulator, even considering its maintenance and depreciation, has lower costs than tests with prototypes, and with a considerable time saving. When dealing with failures, the simulation can be reset in an instant, and all the variables can be easily checked real time or after every try, while for prototype tests these data must come from sensors, that can suffer from malfunctioning due to load conditions.

Many driving simulator softwares present the possibility to select some scenarios useful for NCAP tests like the ones already reported, as in Figure 1.3. This feature enables to perform simulations as close as possible to the real tests for systems certification.

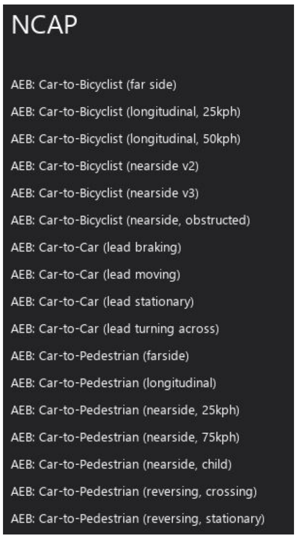


Figure 1.3: Some of the possible scenarios for NCAP tests.

1.4. Driving simulators

Simulations have become more and more widespread in the engineering field. And driving simulators are becoming more and more important and reliable in the automotive field. The focus of this research is on the Adaptive Cruise Control system, tested and validated with the use a dynamic driving simulator. Nowadays, simulators are used for validation, testing and even training. These systems provide a large number of freely tunable variables, such as scenarios, materials, traffic and so on. The potential of the driving simulator is expressed by its fidelity, that defines how much the simulations are realistic. In the market various models are present, from desktop simulators to high fidelity dynamic models, as shown in Figure 1.4

(a) Commercial desktop driving simulator.

(b) DiM400 driving simulator installed in Politecnico di Milano, Bovisa campus.

Figure 1.4: Different types of driving simulators.

In the proposed research, a DiM400 cable-driven Dynamic Simulator is used to test an Adaptive Cruise Control system. This simulator is the newest addition to the VI-Grade driving simulators, having a unique cable driven system for the lower stage, to enable a larger motion envelope for even longer time exposure. For the upper stage, a new hexalift component enables an improved motion envelope by increasing the available vertical travel, which in turn leads to a better vertical feel under combined loading events. Cable-driven Dynamic Simulators structure is reported in Figure 1.5. The lower platform, that translates and rotates in a plane, represents the chassis and tire dynamics, while with an upper stage, a 6 degree-of-freedom mechanism, faithfully recreates the higher-frequency vehicle body motion. This architecture allows to recreate a realistic dynamics, ultimately leading to a better and more immersive driving experience.[23]

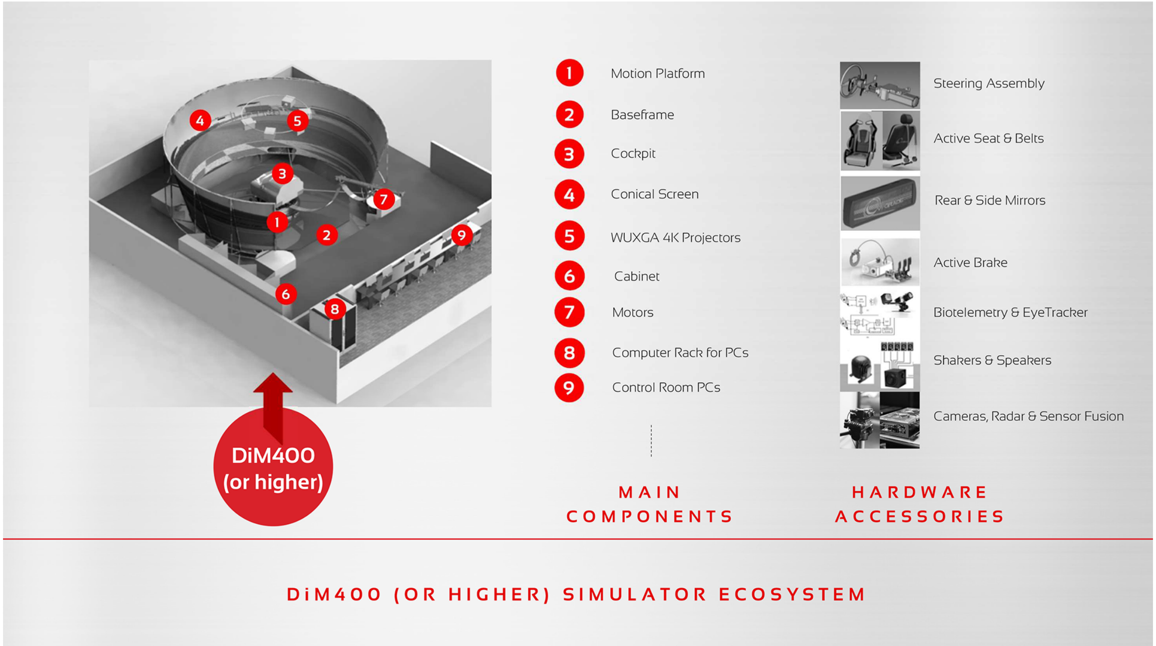


Figure 1.5: DiM400 Driving Simulator structure.

1.5. Goal statement

The Adaptive Cruise Control is a renowned and established system, however it still presents some issues, both from safety and comfort point of view. An issue investigated in some papers is the alternation of the control modes, from speed to distance control[10], [18], [11]. This work has the purpose to test and evaluate a modelled ACC system using an high fidelity driving simulator, as well to improve the ACC logic using the feedback of the driving simulator drivers. To do so, a whole Adaptive Cruise control system has been built, starting from structure and logic present in the literature, then improving its performance based on the simulations results. Lastly, the proposed system has been tested offline and evaluated by a number of tester, in the dynamic driving simulator.

1.6. Thesis structure

The dissertation is divided in multiple sections, first introducing the software and tools used for the creation and simulation of the model. Then the building and testing process is reported.

- Firstly, the general implementation of dynamic models in the driving simulator and their applications are presented. Defining the terminologies and the use of the driving simulator softwares. Hence the setup for the vehicle model, scenario and events tested.
- Then the focus is on the Adaptive Cruise Control system, reporting its structure. The system modes are reported, and the switch logic is presented as well, that define the control action applied by the system.
- The controller types and structure are then described. The desired vehicle behaviour is defined, that is the reference of the controllers. The ACC system is composed of PID controllers with a feedforward contribution in parallel. The gains scheduling, consisting in a fuzzy logic controller, is reported as well.
- Then the first tests are presented. These are performed offline, exploiting the driving simulator software. In these simulations the system is tested in some simple maneuvers, to verify that the model respects the system requirements.
- The last step has been the validation through Human in the Loop simulations. Some testers have been asked to test the system, and then trough a questionnaire validate the model.

2 | Driving simulator

In this chapter the software used by the driving simulator are briefly presented. In particular, the steps for offline simulations using the driving simulator scenario are reported, from the creation of the vehicle model, the event, the scenario and the co-simulation with the ADAS system.

2.1. Terminologies

Firstly, a brief introduction on the terminologies and definitions that are used in the driving simulator environment is presented.

- **Environment**: it represents the scenario where the simulations are performed. A great advantage of driving simulators is the possibility to change the scenario freely, being able to have many different maps, obstacles, ranging from cars to human, or even animals and in-animated objects. Even the weather conditions and the time of the day can be freely changed. In Figure 2.1, two examples of different environments are reported.

(a) Highway scenario.

(b) Urban scenario with rain.

Figure 2.1: Example of different scenarios and conditions.[24]

- **Ego vehicle**: the vehicle controlled by the user, during offline tests it will be controlled by the defined algorithms, while in the online simulations it will be directly controlled by the driver.

- **NPC vehicles**: Non-Player Character vehicles represent the traffic during the simulations. They can be given various commands or triggers, or simply let wander in the map to simulate a random situation.
- **Agent**: it defines an active element in the simulation, such as vehicles, human and animals.
- **Obstacle**: anything still, that can become dangerous for agents in the simulation. Often this term is also referred to other vehicles in the simulation, as they must be avoided as well.
- **Virtual sensors**: in the driving simulator environment virtual sensors are implemented. They present the same logic as real sensors, for instance a virtual Lidar has the same sensitivity to the material reflectivity as a real one. The sensors used for the study, will be later presented.

2.2. Simulation scenarios

A key feature in any simulation is the environment in which it is performed. Below the mostly used environments for ADAS applications are presented.

- **Highway**: the most common scenario when high speed tests and co-axial lanes are required. This environment is perfect to test systems like Lane Keeping (LKA) or Adaptive Cruise Control (ACC) that require long roads and multiple lanes. A vehicle moving in a four lane highway is reported in Figure 2.2.

Figure 2.2: Highway scenario in VI-WorldSim.

- **Extra-urban**: this scenario is used for mid speed simulations, it is really useful to test ADAS that tends to substitute a command of the driver, as already mentioned like an ACC. It is also possible to test crossroads and junctions. In Figure 2.3 a crossroad is represented, with a cyclist as well.

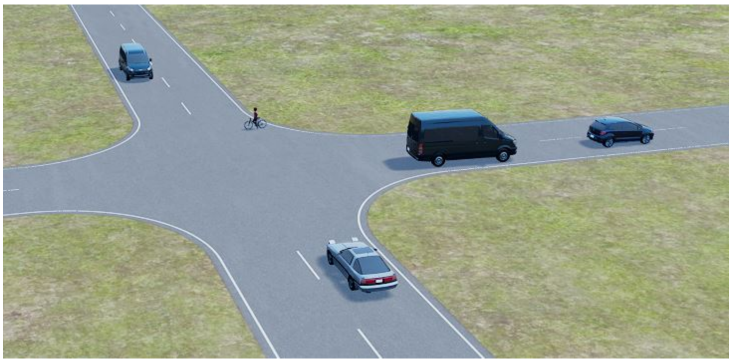


Figure 2.3: Extra-urban scenario in VI-WolrdSim with multiple agents.

- **Urban**: this scenario is used for low speed tests, where the driver has to start and stop many times at crossroads and roundabouts. This scenario is usually populated with many pedestrians and vehicles. It is really helpful to simulate safety systems as emergency braking (EB). In Figure 2.4 a typical neighborhood is represented.

Figure 2.4: Urban scenario in VI-WolrdSim.

For a more complete and realistic simulation, also weather conditions and time of the day are crucial parameters. VI-WorldSim allows to add rainy or foggy conditions, and to change the lighting of the simulations. For ADAS, checking these variables is important because sensors and cameras can be affected, as in real life situations.

2.3. Virtual sensors

ADAS rely on sensors to retrieve information about the environment and the possible obstacles. To take this into account, various types of virtual sensors are implemented in VI-WolrdSim, that give the status of the ego vehicle and the other agents in the simulation. Listed below are the most common VI-WorldSim sensors used for ADAS applications:

- **State Manager**: it returns the current position of a selected agent acting in a specific simulation. It gives an array with GPS signals, one about collisions and one on the actor position. GPS information are reported in Figure 2.5.

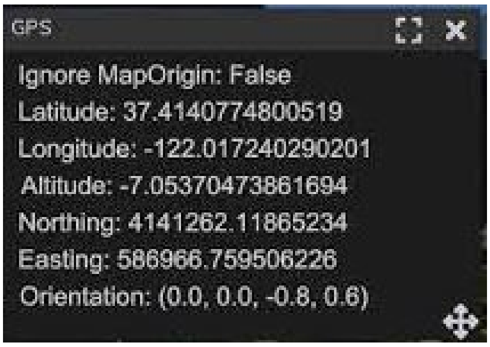


Figure 2.5: GPS sensor information.

- **Virtual Camera**: it represents an ADAS camera that can detect obstacles in front of the driver and lane markings. This ADAS camera has as outputs the objects identification, hence their position, dimensions and orientation. It detects the road lanes as well.

- **Depth Camera**: it simulates a camera which returns depth information for each pixel. The returned image is composed by shades of gray, so if an object is closer the color will be darker and viceversa. The output is reported in Figure 2.6.

Figure 2.6: Depth camera sensor output example.

- **Semantic segmentation Camera**: this camera has been annotated with class and instance labels, so it returns an array of pixels where different colors represent different class labels or instance labels, so any object identified, both moving or not. An example of the sensor output is represented in Figure 2.7.

Figure 2.7: Semantic segmentation camera sensor output example.

- **Radar**: it reproduces an electronically-scanning radar. It publishes an array of objects identifiers, the location of a return, its power and velocity with respect to the sensor.
- **LiDAR**: it represents a mechanically-scanning LiDAR rotating around its yaw axis. It publishes an array of returns, where a return is a 3d point relative to the sensor and the its intensity.
- **Ultrasonic sensor**: it simulates an ultrasonic sensor. It publishes the range to a solid surface, commonly used for low speed maneuvers. The output from this sensor is reported in Figure 2.8.

Figure 2.8: Ultrasonic sensor output example.

2.4. Vehicle model

The integration of the Equations of Motion (EoMs) during the simulations is performed by VI-CarRealTime, that works in co-simulation with the control model built in Simulink. The vehicle model used in this thesis is thus the simplified vehicle mode directly available in VI-CarRealTime, which is composed by five rigid bodies:

- Vehicle chassis (sprung mass)
- 4 wheel parts (unsprung masses)

It includes 14 degrees of freedom (DoFs), the chassis has 6 DoFs while each wheel has 2 DoFs, one to describe the motion with respect to the vehicle body and the other to account for the wheel spin. There is also the possibility of splitting the chassis in two parts that are restrained through linear stiffness so as to account for the chassis compliance in the relative DOFs which have been allowed.

Suspension and steering system properties, such as kinematic compliance, and component properties, are defined using lookup tables. Brakes and powertrain subsystems are described using differential and algebraic equations, so no additional physical part is added to the model.

The vehicle model's intent is to accurately predict the overall vehicle behaviour for cornering, braking, and acceleration-performance studies for four-wheeled vehicles with independent-front and independent-rear suspensions. The simplified model is described in terms of commands and functions that use an internal development environment working as a symbolic manipulator tailored for deriving multi-body equations and a code generator. This simplified model is meant to run faster than real time, making it useful for HiL simulations and driving simulators applications. [25]

2.4.1. Editing the vehicle model

The vehicle model data and system configuration can be edited in VI-CarRealTime Build mode. It is possible to import example models or create a new one from scratch. In Build mode the vehicle is divided in subsystems, shown on the left side in Figure 2.9 in a tree view. Instead, on the right side of Figure 2.9 the user can edit the parameters for the selected subsystems.

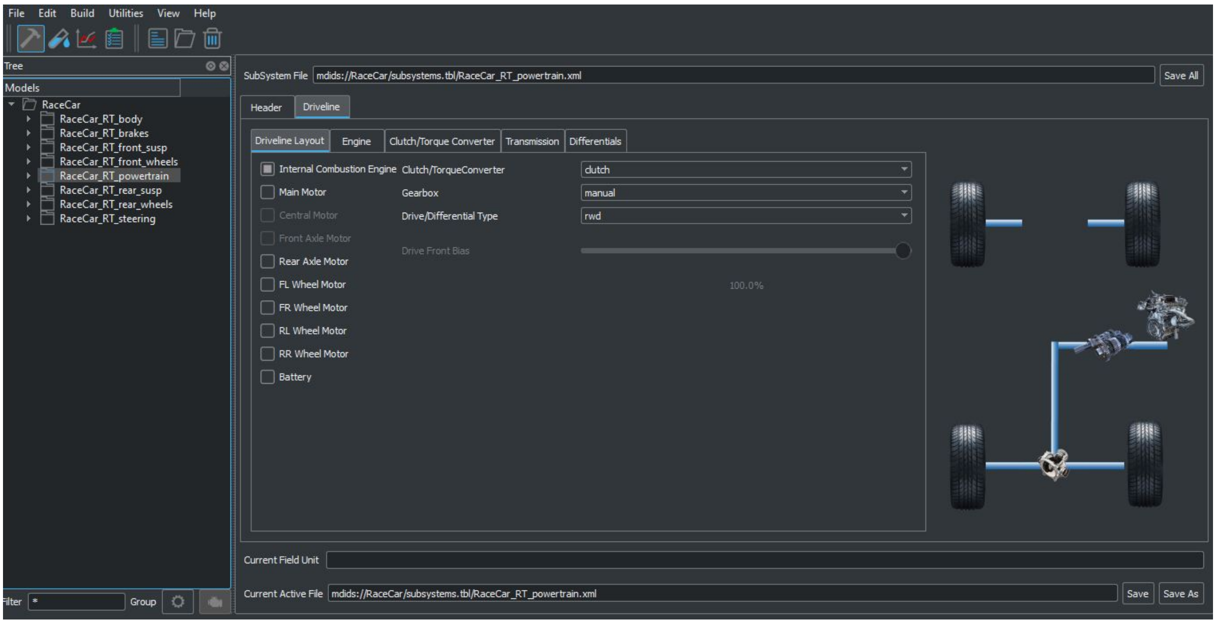
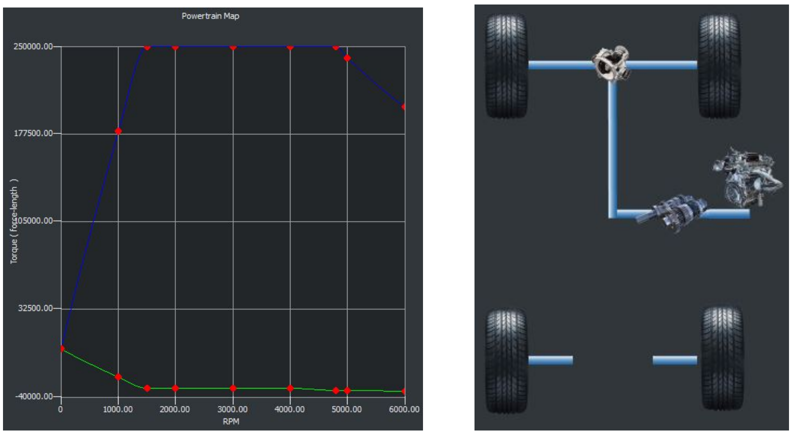


Figure 2.9: VI-CRT GUI: powertrain settings.

The model chosen to test the Adaptive Cruise Control is a *CompactCar*, model available in VI-Grade as part of its software suite. This vehicle is a front wheel drive one, with a traditional internal combustion engine, a typical vehicle for ACC applications. Nevertheless, the type of vehicle does not affect the logic of the ACC controller since, being a feedback controller, it can handle the variations in the vehicle model. This type of vehicle has been chosen for the sake of generality and simplicity, being the topic of this research not on a particular type of vehicle. The engine map is already defined in VI-CarRealTime, and it regulates the output torque from the engine given the torque input required by the ACC model. The engine map of the *CompactCar* model is reported in Figure 2.10a. The driveline layout is reported in Figure 2.10b, showing the differential, the transmission and the internal combustion engine. No modification to the wheel, brakes and steering systems have been made, being not relevant for the ACC application. Table 2.1 reports the main data of the vehicle model.



(a) CompactCar engine map in VI-CarRealTime.

(b) Compact car driveline layout.

Figure 2.10: Powertrain settings for the vehicle model.

	Value	Unit
Mass	1383	kg
Wheelbase	2577.4	mm
Front track	1521.5	mm
Rear track	1555.9	mm
CG height	563.9	mm
Roll inertia	318.96	$kg * m^2$
Pitch inertia	1252.66	$kg * m^2$
Yaw inertia	1217.10	$kg * m^2$

Table 2.1: Vehicle model data.

2.4.2. Event building

When the vehicle settings are saved, the next step is to create and simulate an event in VI-CarRealTime Test mode. Also in this case many example events are given, such as *StraightAcceleration* or *OpenLoopSteering* maneuvers. If the user would like to create a specific event it is possible to run *FileDriven* events, in this case it is necessary to use VI-EventBuilder. During the definition of the event, simulation settings such as initial speed or end condition are defined. In the example reported in Figure 2.11, the steering has been defined as OpenLoop, doing so, the steering demand will be controlled by the Simulink model during the co-simulation, for instance for a LKA application, while other inputs are in Machine, so the VI-Driver will control the throttle, brake and gear. At this point it would be possible to run the created event and post-process the results in the Review mode, and seeing an animation using VI-Animator. However, the focus of this research is on advanced control models application, defined in Simulink, so when the event file is run, the necessary files for the simulation are created, hence the review will be performed in Simulink.

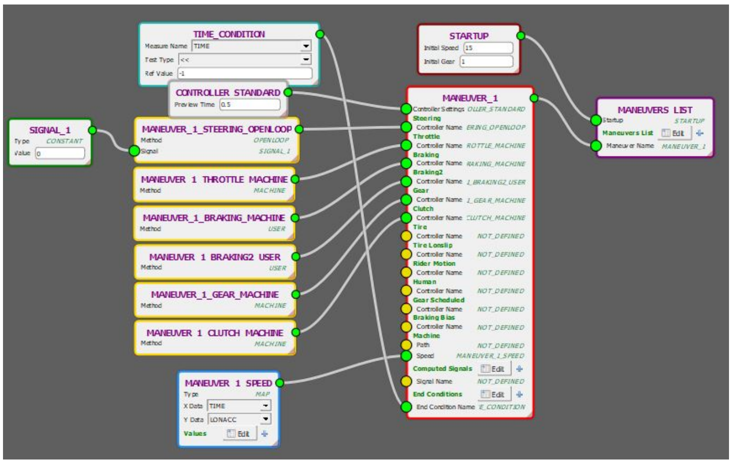


Figure 2.11: Example of FileDriven event in VI-EventBuilder.

2.5. Co-Simulation VI-CRT and Simulink

In order to build more complex models and simulations, VI-CarRealTime allows the user to connect vehicle models with MATLAB environment, where control models are usually developed. The vehicle model is made available in the MATLAB/Simulink environment as an S-function, that has inputs and outputs that can be connected to other blocks. All the car data is retrieved from the "send_svm" file, created when a test event is run in VI-CarRealTime. To connect the two softwares no additional Toolbox is required, it will be simply necessary to add the script "addpath_vicrt_2021.m" to the MATLAB directory. Once vehicle model and event are defined and present in the MATLAB directory, it is possible to run the VI-CarRealTime solver directly from Simulink. The next step is to add the scenario that will be used in the driving simulator and the sensors to simulate a real environment, using the driving simulator software VI-WorldSim.

2.6. VI-WorldSim

VI-WorldSim is the software developed by VI-Grade to build and import scenarios for driving simulator tests, specifically built to support ADAS development. VI-WorldSim Studio is the editor allowing to pick an environment and populate it with actors to define scenarios which can be used for simulation both on a VI-Grade driving simulator and, with the 2021 release, also on offline simulations.[26]

2.6.1. Editing the scenario

As already mentioned there are many available scenarios in VI-WolrdSim, depending on the application and the model to be tested. Once the map is loaded, the first step is the addition of agents and obstacles. Many types of vehicles can be added, as well as pedestrians and animals, and even inanimated objects. Each agent behaviour can be defined by the user directly in VI-WorldSim Studio. It is possible to impose a route or trajectory, defining triggers on the agents, or simply let them wander. In Figure 2.12 a trigger example is reported, in this simulation the pedestrian will start to walk on the crosswalk when the car reaches the blue line on the road. This is a typical test to evaluate the efficiency of an Emergency Braking for instance. For the vehicles controlled by the driver and the control algorithms, the behaviour must be set as "User-Controlled". The weather conditions and time of the day can be changed from here as well.

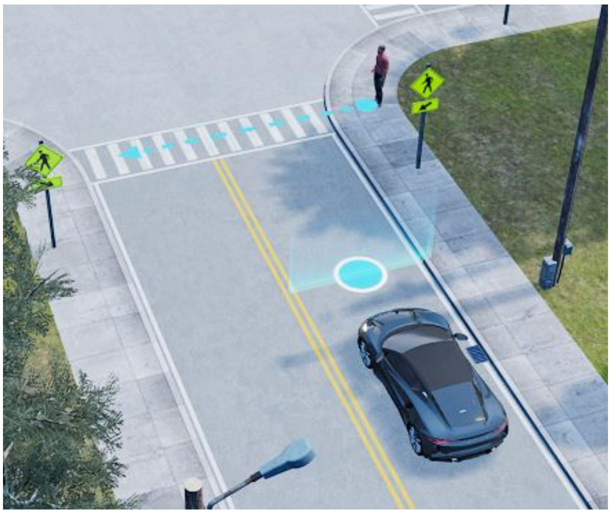
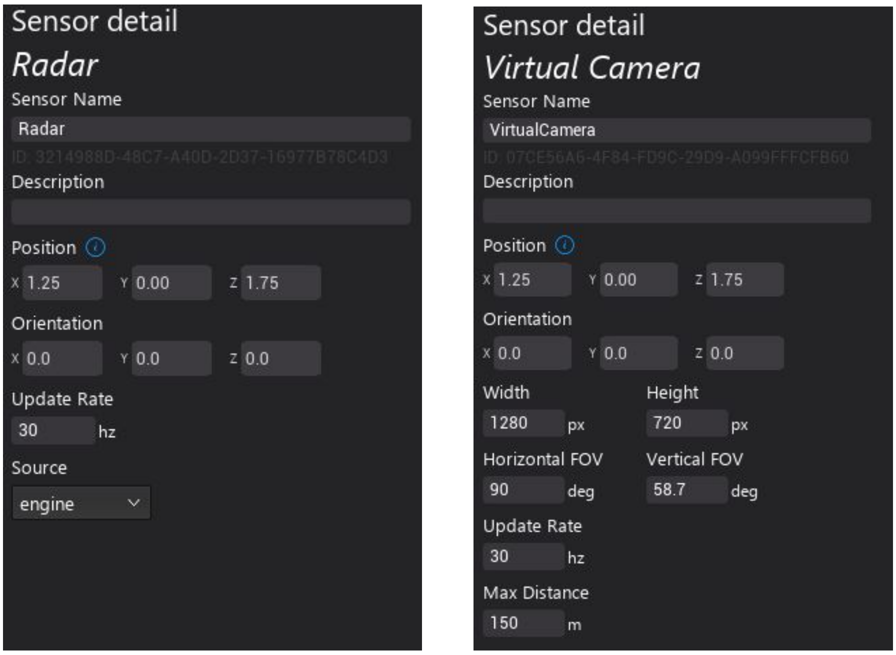


Figure 2.12: Trigger example.

The final step is adding the sensors to the ego vehicle. To do so, the user must select the desired sensor package and its properties, such as the position on the vehicle, the orientation, the update rate and so on. These devices will be recalled in the MATLAB/Simulink model using specific blocks that have as outputs the data arrays from the sensors.

For the Adaptive Cruise Control model in this thesis a radar and a virtual camera are mounted on the ego vehicle. These sensors are both located in the front of the vehicle, both looking forward. The maximum distance perceivable by the camera is 150 meters, a reasonable range for ACC applications. Reported in Figure 2.13 the sensors setup for the ACC model is reported. The radar setup, from Figure 2.13a, presents the position and orientation definition, as well as the source and the sensor rate. Both sensors are placed 1.25 meters in front of the front axle, and 1.75 meters from the ground. Figure 2.13b shows the other necessary setups for the virtual camera. For this sensor the POV, width and height must be specified as well, and the max distance perceivable by the sensor. By selecting 150 meters as maximum distance, the sensor will give as output this value when no obstacle or agent is detected.



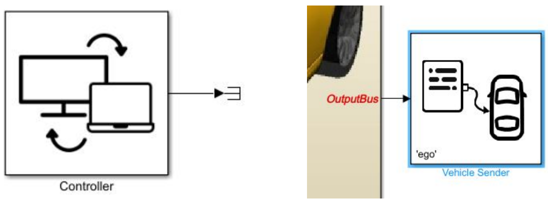
(a) Radar sensor settings.

(b) Virtual camera sensor settings.

Figure 2.13: Sensors settings used in the Adaptive Cruise Control system.

2.7. Co-simulation VI-WorldSim and simulator

Once the model is fully created, the first step to perform offline tests is to add the script "addpath_viworldsim_2021.m" to the Matlab directory, as it was done for VI-CRT. Doing so, it is possible to run a VI-WorldSim session on the laptop. It is essential to specify a fixed step solver with a step of 0.001s, as it is in the driving simulator. To recall the created scenario, specific blocks must be added. The Vehicle Controller manages the time steps of the simulation while the Simulink model is running, to synchronize the Simulink and the VI-WolrdSim steps. This block, reported in Figure 2.14a, is simply added to the Simulink model, without the need of any input and without feeding any other specific block with its output. The Vehicle Sender block, instead, is used to move a user-controlled actor in the scenario. For this reason it needs an input, which is generally provided by connecting it to the VI-CarRealTime S-Function, as shown in Figure 2.14b. All the sensors from VI-WorldSim blocks and their visualizers are available. By putting them in the model, all the outputs for the particular sensor can be used to define the control logic.



(a) Vehicle controller block.

(b) Vehicle sender block receiving the output bus form the S-function.

Figure 2.14: Blocks for offline simulations from VI-WorldSim library.

2.8. Deploy on SimWorkBench

The last step for the testing process is the testing of the ACC control logic built in Simulink on the driving simulator. This means the need to deploy the control algorithm to the Concurrent machine that is managing the driving simulator. This process is entirely managed by using the Simulation WorkBench (SimWB) software, which is in charge of the integration of additional software to VI-Grade suite on the driving simulator Concurrent machine. For the deploy it is necessary to set the abort time to *inf* into the Simulink model, and substitute the inputs and outputs of the VI-CRT S-Function with the MLToolkit RTDB blocks, so that both the S-function and the Vehicle Sender can be deleted. These blocks read and write values from/to any channel belonging to the Real Time DataBase (RTDB). This includes channels from the live simulation, as well as buttons and pedals in the simulator cockpit, enabling the manual activation of systems such as a Cruise Control. These variables are already defined in the RTBD Creator tab, in case the user would like to check any other variable from the simulation, it is simply necessary to use the RTDB Creator tab in the ML Toolkit in Figure 2.15 and create and upload the new blocks. For ADAS applications, it is likely that the control logic will substitute driver inputs as throttle, brake and steering, that are normally controllable in the cockpit.

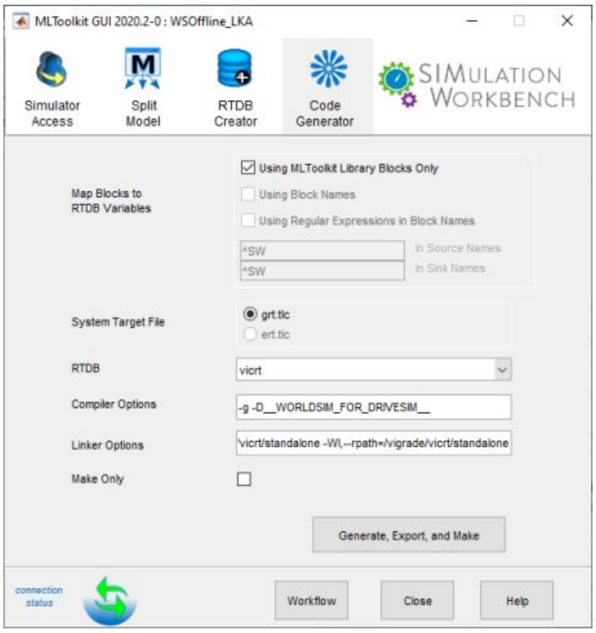


Figure 2.15: SimWB toolkit.

3 | Adaptive Cruise Control

The entire modelling process and the whole logic to create a system equivalent to the state of the art systems will be presented in this chapter, exploiting the features of the Driving Simulator environment.

3.1. Control System Architecture

In Figure 3.1 the overall scheme of the ACC control architecture is shown. The model consists of the signals from the driver, upper and lower level controllers, the ego vehicle dynamics and the data from the environment. The driver activates and deactivates the system and triggers the desired velocity when the system goes ON, which then can be freely changed using other two buttons in the cockpit. The upper level controller uses the data from the sensors in VI-WorldSim and information about the ego vehicle in VI-CarRealTime to set the desired states of the ego vehicle and define the control action to apply to the vehicle. This output is sent to the lower level controller that regulates the engine and braking action. As already explained in Chapter 2, the vehicle dynamics is managed by VI-CRT, and the environment, like road slope and surface conditions, are defined in VI-WorldSim. Vehicle dynamics can present uncertainties, as well as environmental ones, such as aerodynamic and rolling resistance which do not only depend on the vehicle properties but also from external factors. This issue is overcome by measuring the performance of the vehicle, and then providing these data to the controllers, to guarantee optimal performance[10].

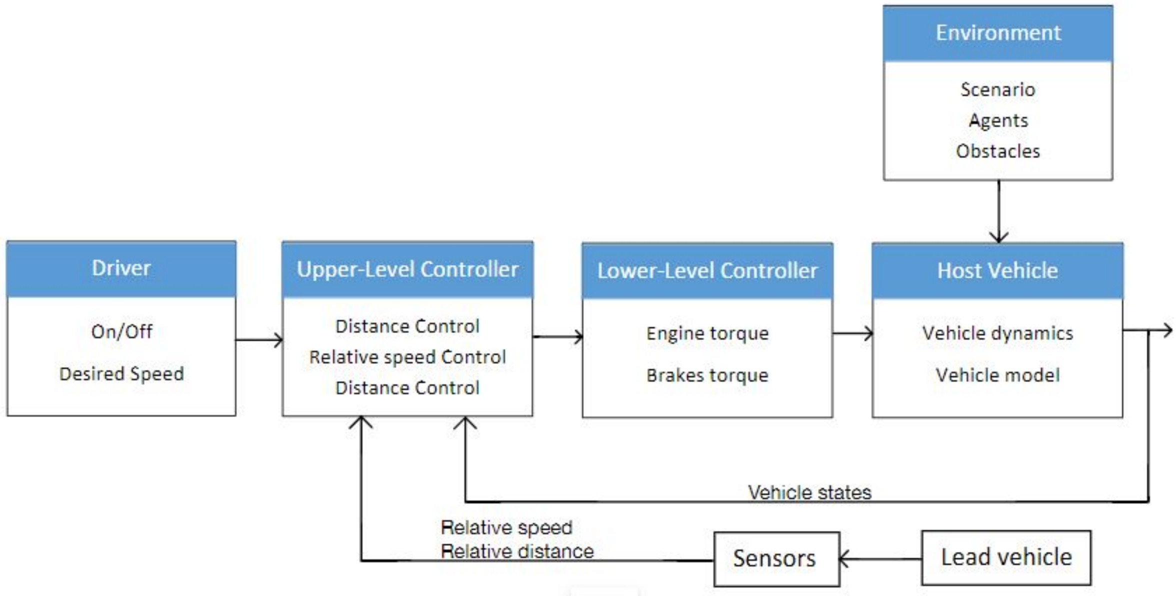


Figure 3.1: ACC Control Architecture Scheme.

3.2. Desired distance

Before introducing the control logic of the ACC system, it is fundamental to define the desired distance from the leading vehicle, which is the main parameter that regulates the control action. The desired distance is defined and calculated depending on the state of the ego vehicle as well as on the environment. This value is then used to select the control mode, that will be defined later in this chapter, as well as the feedback control gains, to ensure an optimal balance between comfort and safety.

In literature many ways to define this distance are presented, the simplest choice is to set a constant value. Despite its simplicity, it is easy to understand that this method presents many drawbacks, being very hard to find the right value for all the driving conditions. Indeed, a too high value can guarantee safety even at higher speed, but can present problems at low speed and in busy traffic conditions, especially when other cars move in front of the ego vehicle. Lower values will show higher comfort, because generally the distance error will be lower, but if the front vehicle brakes drastically it is unlikely that the system will brake fast enough.

To define the desired distance, the most widely used policy is the constant time-headway[27]. Time-headway is the time that the ego vehicle takes to collide with the front vehicle in case it suddenly stops. This method presents the advantage that the desired distance is automatically updated as function of the ego vehicle speed, to ensure both comfort and safety of the passengers. In this case the desired distance is defined as:

$$d_{des} = t_{hw} v_{ego} \tag{3.1}$$

where t_{hw} is the time-headway and v_{ego} is the ego vehicle speed.

A common extension to this policy is to add a constant component, to account for complete stops. Doing so, even at zero speed there will always be a not-null desired distance, called d_{offset} in Equation 3.2, which plays the role of a safety factor in the system[8]. Therefore, the definition of the desired distance is updated as follows:

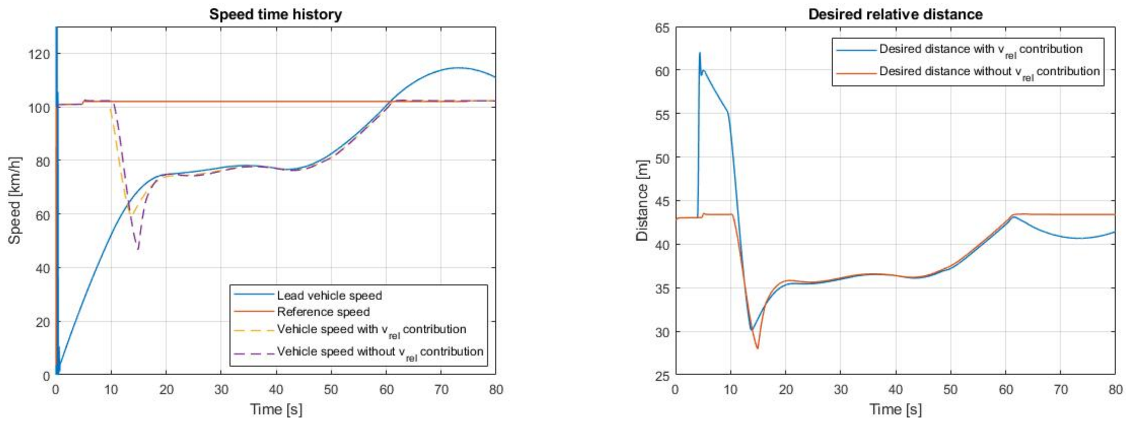
$$d_{des} = t_{hw} v_{ego} + d_{offset} \tag{3.2}$$

A further improvement in the system is accounting for the relative speed of the ego vehicle with respect to the lead vehicle. Doing so, also the state of the lead vehicle is considered to adjust the ego vehicle behaviour. This has shown an improvement in situations where the ego vehicle meets a lead vehicle travelling at lower speed. In this case the controller forces the ego vehicle to brake earlier, making it slower for a moment, to smoothly achieve the right distance between the two vehicles. By adding this contribution the controller does not brake too much when it becomes slower, this gives less deceleration and less speed overshoot. This statement is confirmed by the simulation results in Figure 3.2, where the vehicle that brakes less is the one where the spacing policy used is the one in the Equation 3.3. Without this contribution the controller brakes more to reach the desired distance, that can be dangerous in an highway scenario.

The final expression to define the desired distance is:

$$d_{des} = t_{hw} v_{ego} + t_{rs} v_{rel} + d_{offset} \tag{3.3}$$

Where t_{rs} is a time constant like the time-headway, and v_{rel} is the relative speed between the ego and lead vehicles, available from the sensors. The signs have been defined to have higher desired distance when the lead vehicle is slower than the ego.



(a) Speed profiles in the simulation with different spacing policies.

(b) Desired distance in the simulation with different spacing policies.

Figure 3.2: Comparison of different spacing policies simulation results.

3.3. Mode switch for ACC system

An ACC system has to account at least for two working modes, speed control and distance control. The modes activation depends on the difference between the actual relative distance between the lead and the ego vehicles and the desired one defined in the controller. To implement this logic alternation, it is necessary to set a switch between the two control algorithms. In literature many approaches have been presented to cope with this task.

The simplest way to define the proper mode is given by:

$$mode = \begin{cases} DC & d_{rel} \leq d_{des} \\ SC & d_{rel} > d_{des} \end{cases} \tag{3.4}$$

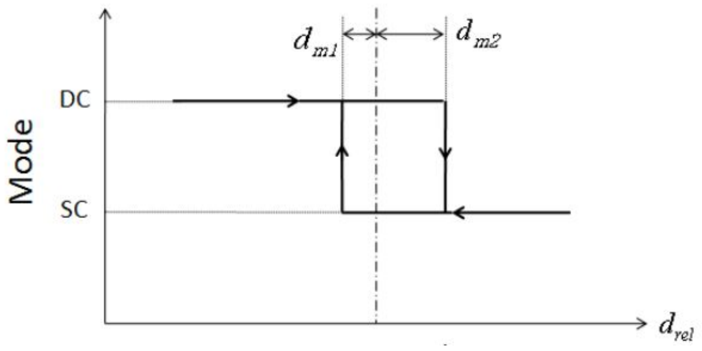


Figure 3.3: Mode switching logic.[11]

However, this scheme can present many issues, since it can cause chattering in the controllers switch if the actual distance is close to the desired one. To avoid so, marginal distances have been introduced to introduce hysteresis in the mode switching, as shown in Figure 3.3. This logic, is called mode switching scheme[11], whose control structure is presented in Figure 3.3. Where the dashed line represents the desired distance defined in Equation 3.3, and DC and SC refer to distance and speed control respectively. The mode decision is then defined as follow:

$$mode = \begin{cases} SC \rightarrow DC & d_{rel} - d_{m1} \leq d_{des} \\ DC \rightarrow SC & d_{rel} + d_{m2} > d_{des} \end{cases} \tag{3.5}$$

Where d_{m1} and d_{m2} are the marginal distances. The input signals for the high-level controller depend on the active mode. The error for the distance controller is defined as:

$$e_1 = \begin{cases} d_{des} - d_{rel} & mode = DC \\ 0 & mode = SC \end{cases} \tag{3.6}$$

While the error for the speed controller is defined as follow:

$$e_2 = \begin{cases} 0 & mode = DC \\ v_{des} - v_{ego} & mode = SC \end{cases} \tag{3.7}$$

The output of the speed and distance controllers is given as input to a switch function which has the role to forward the information to the low level controller and leaves active only the contribution corresponding to the active mode.

This last switch function is defined as follows:

$$a_d = \begin{cases} a_1 & mode = DC \\ a_2 & mode = SC \end{cases} \tag{3.8}$$

Where a_d, a_1 and a_2 are the outputs from the controllers, as defined in Figure 3.4.

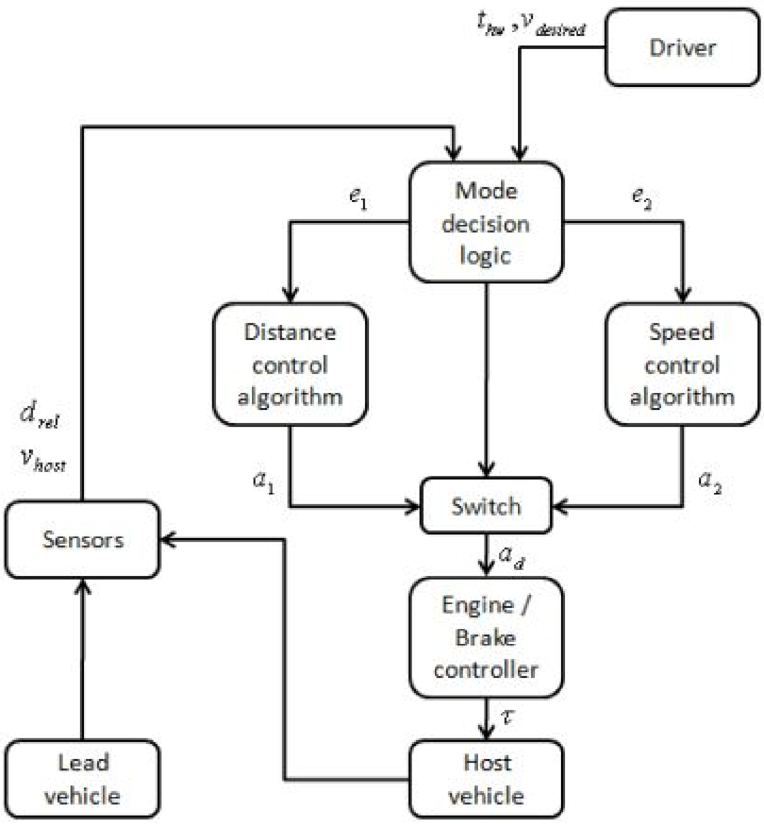


Figure 3.4: ACC control switching scheme.[11]

This scheme allows to prevent chattering of the controllers, however, it leaves a range between the two modes where no control action is applied. Hence, a third mode has been added to the system to fill this gap.[18].

The final operational modes of the ACC systems are thus:

- **Cruise**: it represents the standard Cruise Control, defined as SC in Figure 3.3. The controller has as inputs the error between the ego vehicle speed and the driver desired speed.

- **Follow**: in this mode the controller matches the obstacle vehicle speed while reaching the desired distance. Its error is the relative speed between the two vehicles and a weighted component of the distance error. The latter is added to stabilize the system on the desired distance. So the input error is defined as follow, where w_d is a weight to have speed and distance errors which are of the same order of magnitude:

$$e_{follow} = v_{rel} + w_d(d_{des} - d_{rel}) \tag{3.9}$$

- **Safety-Critical**: the controller directly acts on the distance error to avoid collisions. In this mode safety is prioritized over comfort.

To define the alternation between the three control modes, the relative distance is not the only variable to look at, while also the speed of the lead vehicle must be considered, both to avoid breaking traffic rules and for safety reasons. In Table 3.1 the logic of the controllers switch is reported, highlighting all the possible conditions, where v_{lead} represents the speed of the lead vehicle, and v_{set} is the speed that the driver sets when the ACC system is activated.

<table>
<tr><th></th><th></th><th>Mode</th></tr>
<tr><td colspan="2">$d_{rel} \leq d_{des} - d_{m1}$</td><td>Safety-Critical</td></tr>
<tr><td rowspan="2">$d_{rel} > d_{des} - d_{m1} \land d_{rel} \leq d_{des} + d_{m2}$</td><td>$v_{lead} \leq v_{set}$</td><td>Follow</td></tr>
<tr><td>$v_{lead} > v_{set}$</td><td>Cruise</td></tr>
<tr><td colspan="2">$d_{rel} > d_{des} + d_{m2}$</td><td>Cruise</td></tr>
</table>

Table 3.1: Decision making for the ACC system mode.

Thus, the final system presents three switches that define the three operatic mode of the ACC system. In Figure 3.5, the implementation of this logic in Simulink can be appreciated. The switches are defined using lookup tables function of the relative distance between ego and lead vehicles.

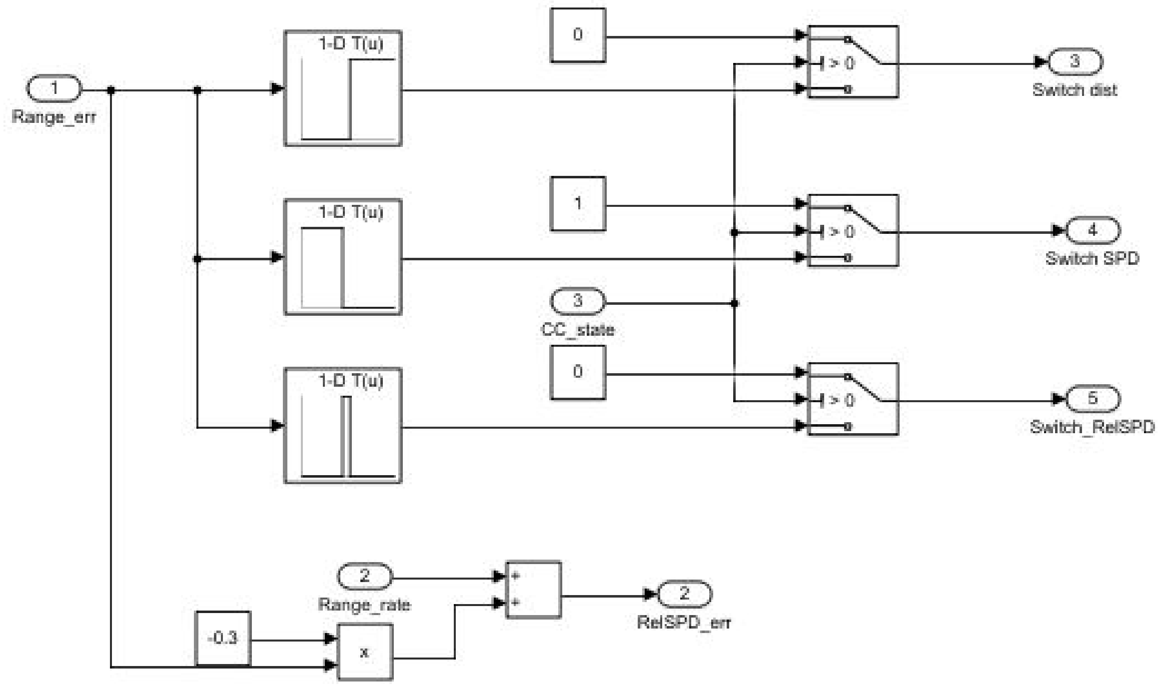


Figure 3.5: Simulink subsystem for ACC mode switches.

3.4. Extensions to conventional ACC

The Adaptive Cruise Control system is built to account for many unpredictable situations, being the behaviour of other vehicles not foreseeable by the driver or sensors. For this reason, extensions or modifications of these system are present in the market, such as the Emergency Braking (EB) system, or the Cooperative Adaptive Cruise Control (CACC). These systems are built and tested offline to ensure a vehicle safe operation in a wide variety of situations and configurations, even though they have not been tested on the dynamic driving simulator.

3.4.1. Emergency Braking

As already mentioned, the ACC system must guarantee both safety and comfort. To achieve so, the controllers have been defined with variable gains, and a set of rules to cover all possible situations, so that the system can adapt to the changes in the environment, surrounding him. The definition of the desired distance takes this into account as well. Indeed its value must be a trade-off between long distances for safety and short ones for comfort. However, for ACC applications it is not feasible to leave very large distances between two vehicles. In fact, many spacing policies are nowadays used in ACC systems[27], but none of them defines distances over a few tens of meters, even at higher speed. Let us think at a really common example, when another vehicle overtakes the driver and then moves in the front. It is very unlikely that the other car returns in the front of the ego vehicle very far, but more probably just a few meters further. In this case, if the reference distance is too high, the error will be high as well, generating a strong reaction from the controller. For these reasons the desired relative distance is kept to a reasonable value. Therefore, to ensure safety in case a strong braking action is required, an Emergency Braking(EB) system is added to the model. The EB consists in an additional system, that is only active when an obstacle that travels at a speed much lower than the ego vehicle is detected. It is then deactivated when enough space between the two agents is reached. The decision to add this system in parallel to the ACC derives from the fact that EB must be used only in very few specific cases, so it is adopted only when it is absolutely necessary. The EB controller is represented in Figure 3.6.

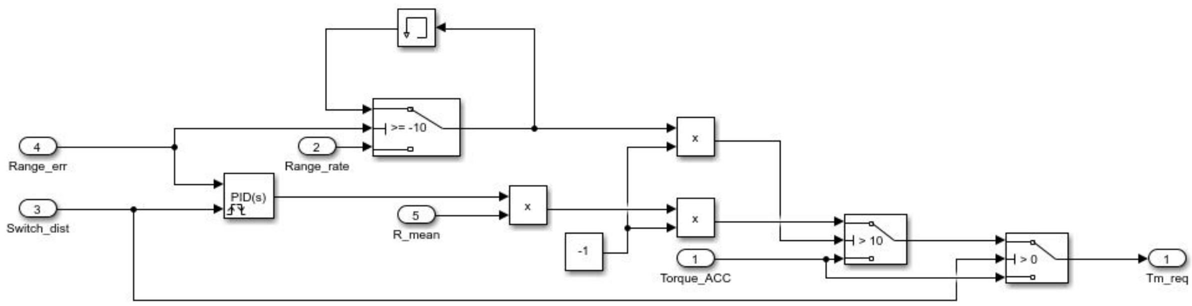


Figure 3.6: Emergency Braking block.

3.4.2. Cooperative Adaptive Cruise Control

CACC systems exploit modern V2V or I2V communications. This technology allows to retrieve information about vehicles far ahead, and act as a consequence to their behaviour. For example, when the second vehicle ahead starts to brake, it is convenient to start braking before the first vehicle ahead the ego starts braking as well. It has been demonstrated that using CACC systems and information from two vehicles ahead[5], instead of just looking at the first predecessor like in normal ACC, the vehicle reacts quickly to changes downstream. By being able to faster react, desired distance and brake intensity can be reduced, increasing the comfort level while ensuring safety and string stability, since the system is able to stop propagating downstream the disturbances coming from ahead in the series of vehicles, called platoon. Many information can be transmitted to the ego vehicle, for instance the position, speed and acceleration of the vehicles in platoon, to define the control algorithms on the states of many predecessors. These data are communicated with a time lag, depending on the adopted technology and the distance between vehicles. To cope with this issue, brake and throttle commands can be communicated as well. This can greatly help because there is an intrinsic dynamic lag between the throttle/brake demand and the effective vehicle acceleration or deceleration. So, if the ego vehicle can have this information, it can virtually react before other vehicles actually change their state.

Consider the case where the communicated data are from the first two predecessors[12], so the vehicle i uses data from vehicles $i-1$ and $i-2$, as depicted in Figure 3.7. The desired distance between vehicle i and $i-1$ is named as $d_{des,i,i-1}$, the same defined in Equation 3.3, like in a conventional ACC. In a CACC the controller uses the communicated information to define the desired state, so the optimal inter-vehicle distance between the two leading vehicles, considering a constant time-headway policy, is:

$$d_{des,i-1,i-2} = d_0 + h_0 v_{i-1} \tag{3.10}$$

Where d_0 and h_0 are the offest distance and time headway, as defined for the ACC in Section 3.2.

Hence, the optimal distance between vehicle i and $i-2$ is:

$$d_{des,i,i-2} = d_{des,i,i-1} + d_{des,i-1,i-2} = 2d_0 + h_0(v_i + v_{i-1}) \tag{3.11}$$

Finally, the desired ego vehicle states can be defined as follow:

$$\begin{cases} d_{des,i} = q_{d1} d_{des,i,i-1} + q_{d2} d_{des,i,i-2} \\ v_{des,i} = q_{v1} v_{i-1} + q_{v2} v_{i-2} \end{cases} \quad with \quad \begin{cases} q_{d1} + q_{d2} = 1 \\ q_{v1} + q_{v2} = 1 \end{cases} \tag{3.12}$$

Where q_d and q_v are constants to weigh the two contributions.

In Figure 3.7 it is represented the case where the desired ego vehicle position is over the $i-1$ vehicle. This would result in a collision. To avoid hitting the first vehicle ahead in the platoon, while looking at the leading vehicle, a minimum desired distance must be defined. Hence $d_{des,i}$ is saturated like in the following:

$$d_{des,i,min} = q_{d,min} d_{i-1,i-2} \qquad q_{d,min} > 1 \tag{3.13}$$

With this formulation, it is assured that the desired distance to vehicle $i-2$ will always be greater than the distance between the leading vehicles, plus a safety margin. This helps in situations where the second leading vehicle accelerates, while the first one is still moving at slower speed.

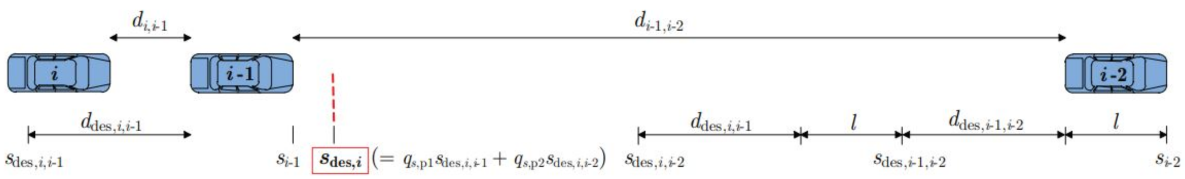


Figure 3.7: Representation of a vehicle platoon and the desired distances between vehicles.

In Simulink the desired distance, and its saturation, are computed as reported in Figure 3.8.

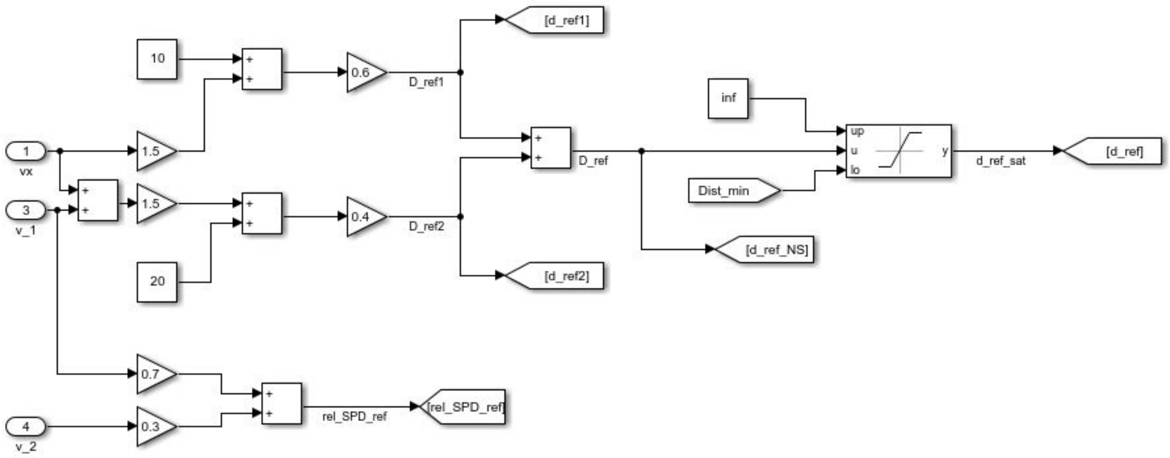


Figure 3.8: Cooperative Adaptive Cruise Control reference distance computation.

4 | Control model

In this chapter the developed ACC system is described, showing how the logics presented in the previous chapters are implemented in Simulink and VI-WorldSim.

4.1. Reference generation

The first step to design the Adaptive Cruise Control is the definition of the reference for the alternating controllers. Depending on the active mode, the reference can be defined through the actual state of the ego vehicle or even through the information coming from sensors in VI-WorldSim.

4.1.1. Cruise Control

To model an ACC, it is first necessary to build a conventional Cruise Control. The main point in that is to definition of how to trigger for the activation and deactivation of the system, as well as the reference speed that will be the input to define the error for the speed controller.

The activation state must change according to the inputs of the driver which is pushing a button to turn on and off the system. This variable is defined so that when an impulse is detected on the state change button it maintains the activated state as long as another impulse is detected, or when the driver uses the brake pedal. For offline simulations the input signal is replicating what happens for tests in the simulator, where the signal is sent directly from the buttons in the cockpit. In the model it is possible to activate separately the CC and ACC systems using two different buttons, so when only the CC is activated, the vehicle only maintains the set speed without looking at other vehicles. When one of these systems is active, the longitudinal dynamics of the vehicle is entirely controlled by the system, overwriting brake and throttle demand of the driver. This logic is depicted in Figure 5.1a.

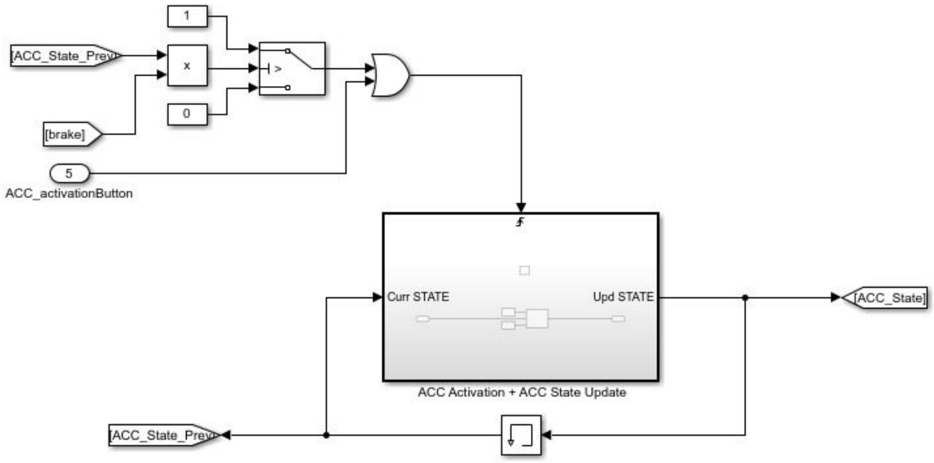


Figure 4.1: Cruise Control reference speed generation.

Once the activation state is defined, it is possible to define the reference speed. As reported in Figure 4.2, the reference speed value is computed using the actual vehicle speed at the time the state activation is triggered. Starting from the actual vehicle speed, it is upper rounded so as to obtain an even reference speed when expressed in km/h.

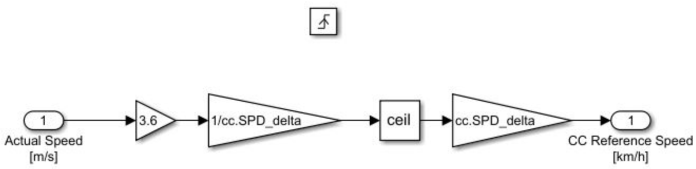


Figure 4.2: CC reference speed computation.

Another functionality included is the possibility to change the speed using buttons when the system is active. These buttons simply modify the reference speed value by 2 km/h up or down, each time they are pressed, so the controller will have a new desired value. In Figure 4.3, it is shown how these buttons action is implemented in the Simulink model.

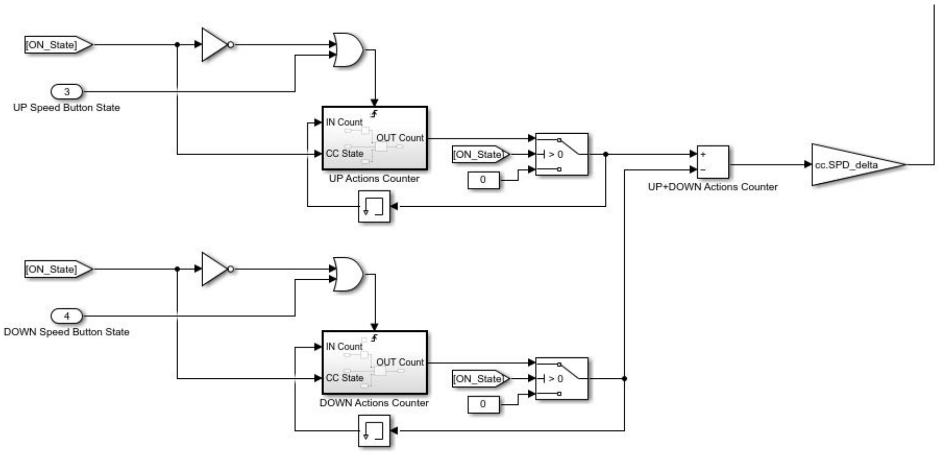


Figure 4.3: Blocks to change reference speed in Cruise Control.

4.1.2. Adaptive Cruise Control

The ACC system relies on sensors, defined in VI-WolrdSim. In this case, a radar and a virtual camera are used to provide the inputs for the controllers.

The radar is able to detect up to 64 targets and retrieve the following information:

- **tag**: numerical identifier of every different target [-].
- **yaw**: obstacle yaw angle in the sensor reference frame [rad].
- **range**: obstacle distance from the origin of the sensor reference frame [m].
- **power**: power of the returned signal [dB].
- **lat rate**: the obstacle's speed in a direction which is parallel to the sensor [m/s].
- **range rate**: the obstacle's speed in a direction perpendicular to the sensor [m/s].
- **status**: status of the track [0 - new, 1 - tracked] [-].

In Figure 4.4 an example of the radar output in terms of detected objects positioning with respect to the sensor reference frame is shown, where each color represents a different tag.

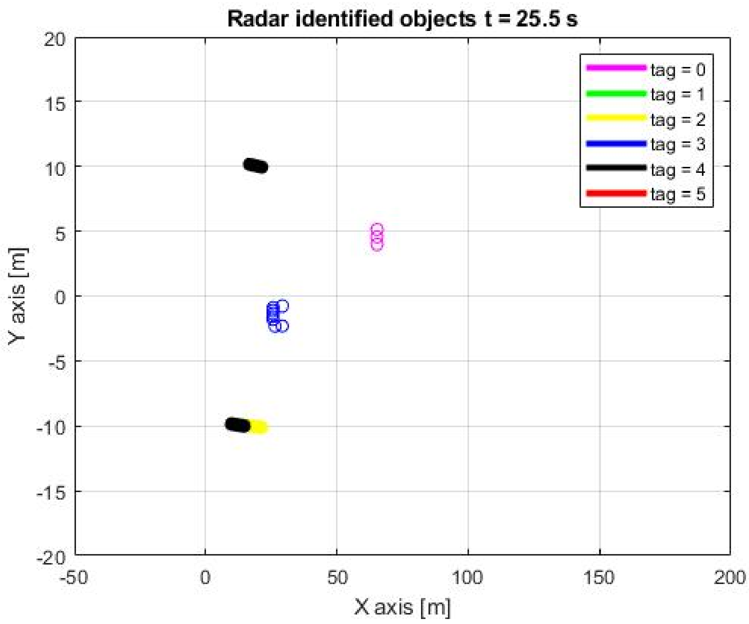


Figure 4.4: Example of objects identified by the radar.

The virtual camera is used for obstacles and lane markings detection. It gives the following outputs:

- **tag**: numerical identifier of every different target [-].
- **x, y, z**: position of the target centroid in the sensor reference frame [m].
- **roll, pitch, yaw**: angle of the target bounding box in the sensor reference frame [rad].
- **x, y, z sizes**: size of object bounding box [m].

The camera identifies lanes with a series of coefficients, used to compute a polynomial that defines the road lanes, between X_{min} and X_{max}, longitudinal distance from the sensor. In Figure 4.5 it is shown how the polynomials are defined in the reference system of the sensor when considering a camera mounted on the front bumper of the vehicle. Hence, using these information it is possible to set the controller to operate only when an obstacle is present in the same lane as the ego vehicle, so that it can work during a turn as well, if the lead vehicle is not obfuscated by other elements.

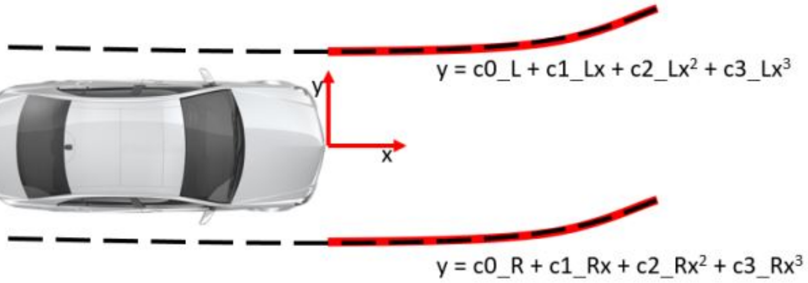


Figure 4.5: Lanes identification from Virtual Camera.

The camera is set to give as output the maximum range which can be seen by the sensor, equal to 150 meters, until an obstacle is detected in the same lane. The virtual camera from VI-WorldSim retrieves the dimension of the obstacle as well. Therefore the controller is identifying a possible obstacle when the vehicle starts to enter in the lane, and not only when its whole body is inside it, according to the following condition:

$$If\ y_{lv} - \frac{track_{lv}}{2} < \frac{lane\_width}{2} \quad then \quad Obstacle\ is\ detected \tag{4.1}$$

Where y_{lv} and $track_{lv}$ are the lateral position and the track width of the lead vehicle in the camera reference frame. In Figure 4.6, a qualitative representation of these quantities is reported.

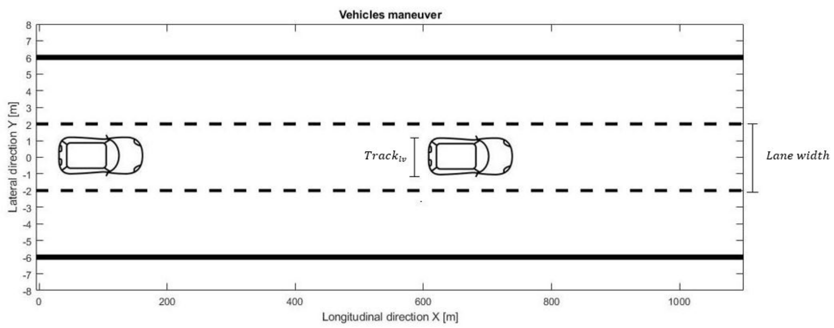


Figure 4.6: Lanes identification from Virtual Camera.

In Figure 4.7, it is reported the implementation in Simulink of Equation 4.1. With this subsystem the elements of interest, representing the data from the lead vehicle, are selected from the output arrays of the Virtual Camera sensor.

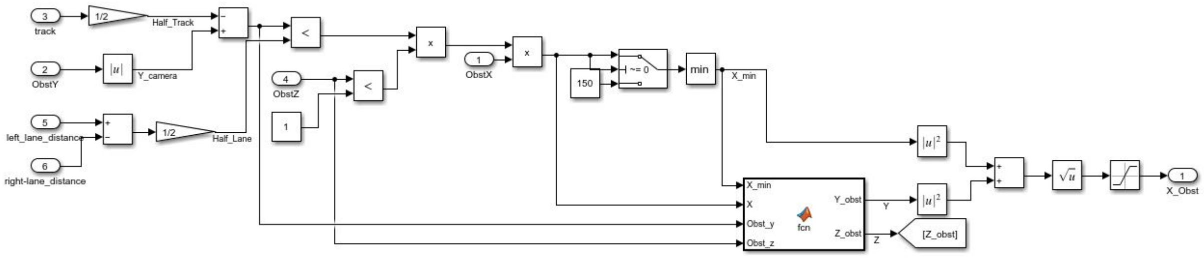
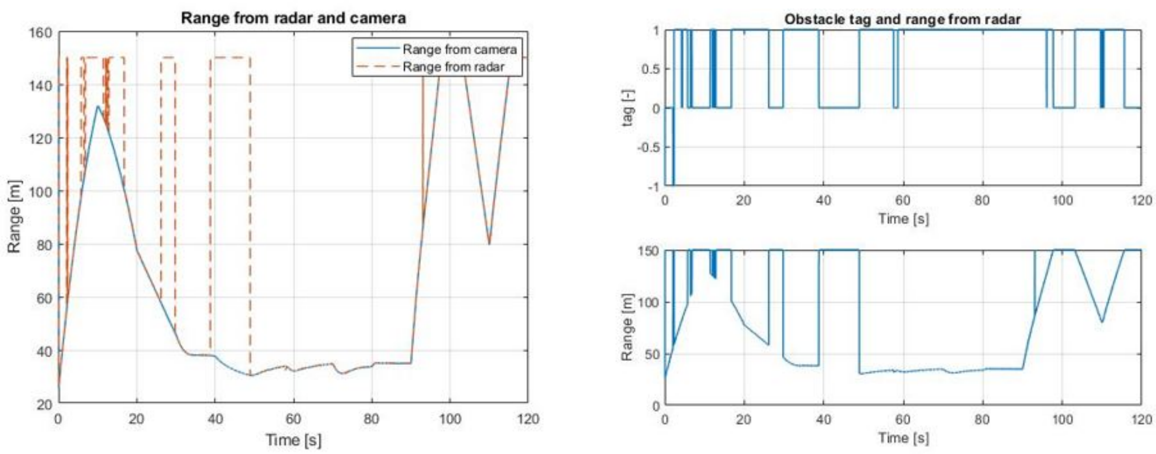


Figure 4.7: Range computation from Virtual Camera.

At first, the radar was used to retrieve the range and range rate, that were the inputs of the distance and relative speed controllers. However, the radar showed some issues during offline simulations. As reported in Figure 4.8a the radar loses track of the obstacle while the camera still shows a continuous signal. This can be due to a saturation of the radar channel, being unable to sample more objects in the environment. The tag from the radar confirms this as well, in fact when the obstacle is lost the tag changes consequently, as reported in Figure 4.8b. For these reasons, the camera has been used as single sensor for the ACC system, being the information from the radar not reliable in every situation. The virtual camera has been chosen as well because it represents an ADAS camera module, publishing an array of lane marking polynomials, an array of detected traffic signs and an array of detected obstacles.



(a) Range from radar and camera.

(b) Tag and range from radar.

Figure 4.8: Issues with radar in VI-WorldSim during offline simulations.

4.2. Controllers

The all system consists of three different controllers in parallel, that switch according to the logic presented in Table 3.1. These switches define the status of each controller. The controller is composed of feedforward and feedback components. The controllers have as output a longitudinal force, that multiplied by the rolling radius, gives the total theoretical wheel torque that has to be applied to the vehicle. Usually the vehicle dynamics in systems like an ACC is managed using brake and throttle demand. The decision to use the torque is due to the fact that this parameter has more physical meaning. This type of command signal can work with any kind of vehicle model. For instance, with an internal combustion engine car, it is split between engine and brake torque. In case of a vehicle with independent wheel drive electric motor, both positive and negative torques are regulated by them and so on. A great advantage is that all the engine and brakes configurations are defined in the vehicle model in VI-CarRealTime, that automatically updates the outputs of the simulation. The final control scheme is depicted if Figure 4.9.

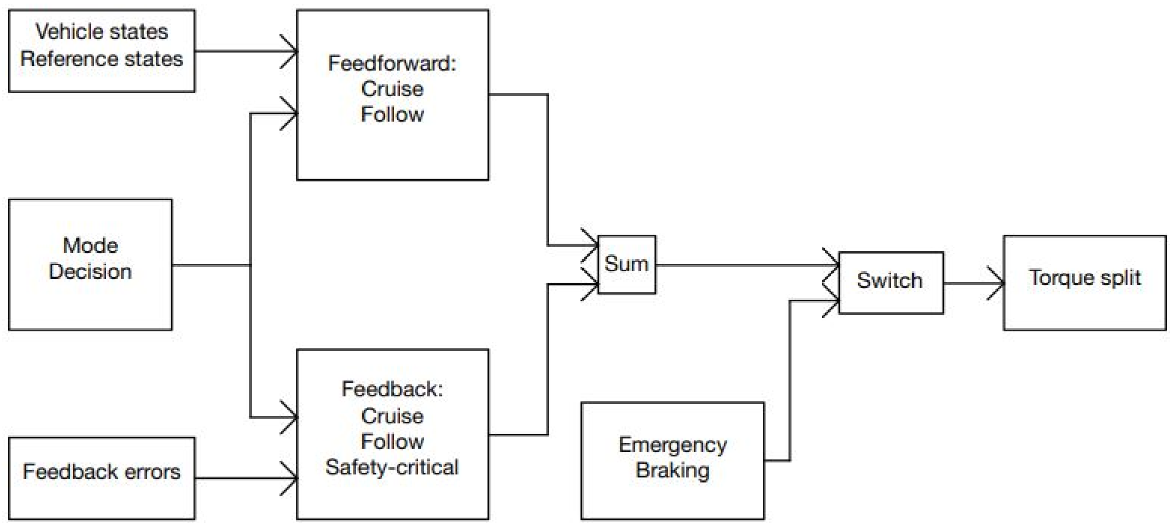


Figure 4.9: Final control scheme of the ACC system.

4.2.1. Feedforwand controllers

Feedforward (FF) control is not a self-correcting controller, therefore if the input adjustments fail to produce the correct output, the process continues to produce the wrong output. However, feedforward control takes corrective actions before the disturbances enters into the process and it does not affect the stability of the system. So the correcting action is faster than for a FB system because there is no need to wait for an error to build up before reacting.

For these reasons a FF contribution is added, working in parallel with the feedback one, that is in charge of attenuating disturbances in the system. In this case, two controllers have been defined, one active for the set speed and one for the relative speed. When the *safety-critical* mode is active, no FF force is provided to the system. Instead, when the *follow* mode is active the output force is the sum of the feedback and feedforward contributions. The FF contribution output is a longitudinal force, that aims to balance the resistant forces that in this case are the inertial, aerodynamic and rolling ones. They can be written as:

$$\begin{cases} F_{in} = m_v a \\ F_{aero} = \frac{1}{2}\rho C_d A_f v_x^2 \\ F_{rol} = f v_x \end{cases} \tag{4.2}$$

Where m_v and A_f are the vehicle mass and frontal area, f and C_d are the rolling resistance and aerodynamic drag coefficients, and ρ is the air density. v_x and a are the longitudinal speed and acceleration of the ego vehicle. This formulation is quite simple, but extremely high accuracy is not necessary for this application, asince there is the FB branch which helps in achieving the reference value. The relevance of these controllers is given by the fact that they give a contribution based on the vehicle and reference states.

When the CC mode is active, the feedforward contribution is computed using the reference speed, and its derivative, so a reference acceleration. This gives a force function of the reference state, and is particularly useful when the driver changes speed using buttons. The FF contribution reacts instantly to the state change, giving a force to obtain the transition towards the new reference. If only a FB controller was used here, the system would have been much less reactive.

In *follow* mode, the FF gives a contribution on the absolute speed and acceleration of the ego vehicle. This is necessary because the torque provided by the controller must take into account the vehicle state, not only looking at the relative speed with respect to the lead vehicle. Indeed, the FF contribution aims at maintaining a constant speed of the ego vehicle, while the FB part is in charge of minimizing the relative speed with respect to the lead vehicle. In Figure 4.10 the feedforward controllers subsystem in Simulink is shown.

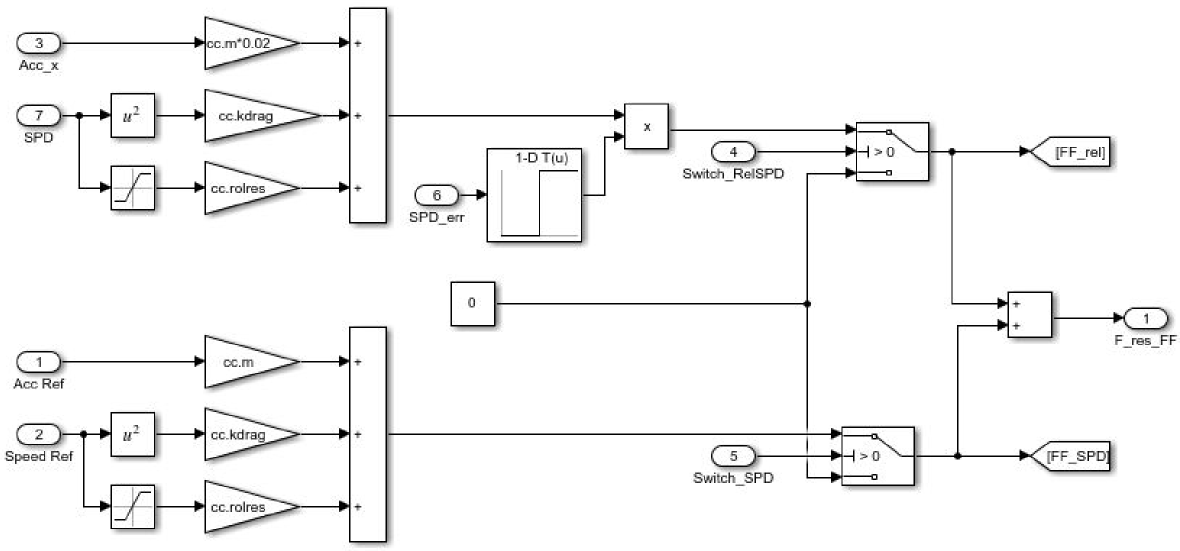


Figure 4.10: Feedforward contribution.

4.2.2. Feedback controllers

The great advantage of closed-loop controllers is that they measure the output, so they always look at the status of the system to reach the desired behaviour, thus being self-correcting.

The feedback(FB) contribution of the system is given by PI(D) controllers, that since many years are one of the mainly used controllers, like in [15] where a PI is implemented on the velocity and distance error. A PID acts on errors defined as the difference between the reference values and the states of the system. It evaluates the time derivative and integral of the error and feeds the sum of those and the error itself back to the system, each multiplied by a gain factor.

The controller on the set velocity, that defines the Cruise Control system, is a PID, so its control law is given by:

$$\begin{cases} F_{FB_{SPD}}(t) = k_p e_{SPD}(t) + k_I \int_0^t e_{SPD}(t)dt + k_D e'_{SPD}(t) \\ e_{SPD}(t) = v_{des}(t) - v(t) \end{cases} \tag{4.3}$$

e_{SPD} is the error between the desired speed and the actual vehicle speed. By imposing so, the controller will force the vehicle to follow the desired velocity. The output is a longitudinal force, called $F_{FB_{SPD}}$. By using a PID the controller is able to add a contribution on both the acceleration and position of the vehicle, by deriving and integrating the error. The derivative contribution is still little to avoid introducing disturbances in the system.

The controllers on the distance is a PID as well, while the relative speed one is a PI, and presents the same control law as the one in Equation 4.3 without the derivative contribution. This has been done because the input for this controller is the relative speed between the ego and lead vehicle. The differential operation on this signal introduces some disturbances and noise, even after being filtered, so it has been preferred to remove this contribution. The integral contribution instead is necessary to ensure that the system has zero error at steady state. This aspect is essential in an ACC since it can be used for very long time.

The emergency braking is another controller, that is not usually active though. It is a PID on the distance error as the *safety-critical* distance controller, but with much higher gains. As said in the previous chapter, the decision to use another controller comes from the fact that this control action must be active only when it is needed. So, the distance controller gains do not reach such high values, because it must be avoided to trigger an emergency braking reaction if not necessary. Therefore, the EB controller is switched on in case a vehicle, or any kind of obstacle, is approaching at a speed much lower than the ego vehicle. In this situation the system understands that a strong braking is necessary, and safety is prioritized. Of course, when the EB is activated, it overwrites the action of the other controllers, until the ego vehicle reaches an adequate safety distance from the lead vehicle. In Figure 4.11 the subsystem for the distance controller is represented .In Figure 4.12 both speed and relative speed controllers are shown. The output of these controllers is regulated by the mode switches defined on the ego and lead vehicles relative distance.

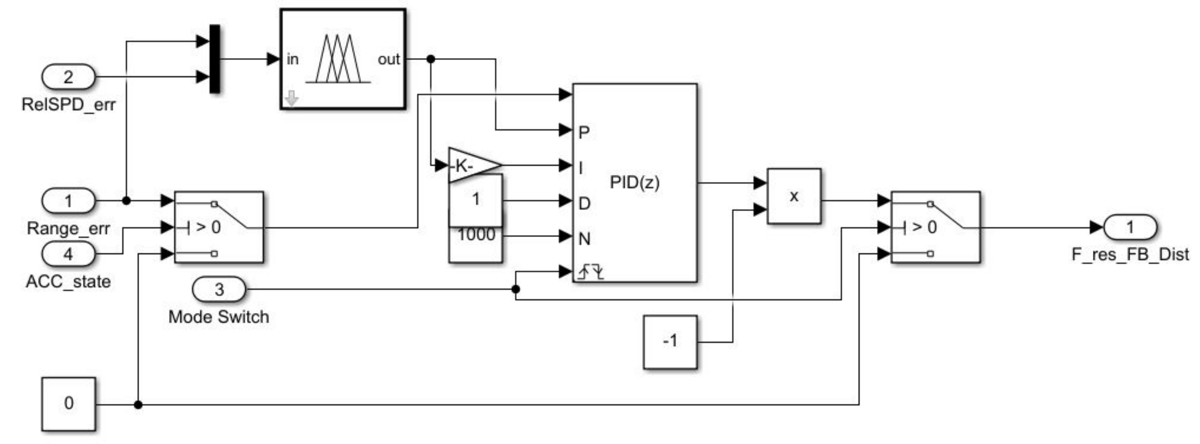


Figure 4.11: Distance feedback controller.

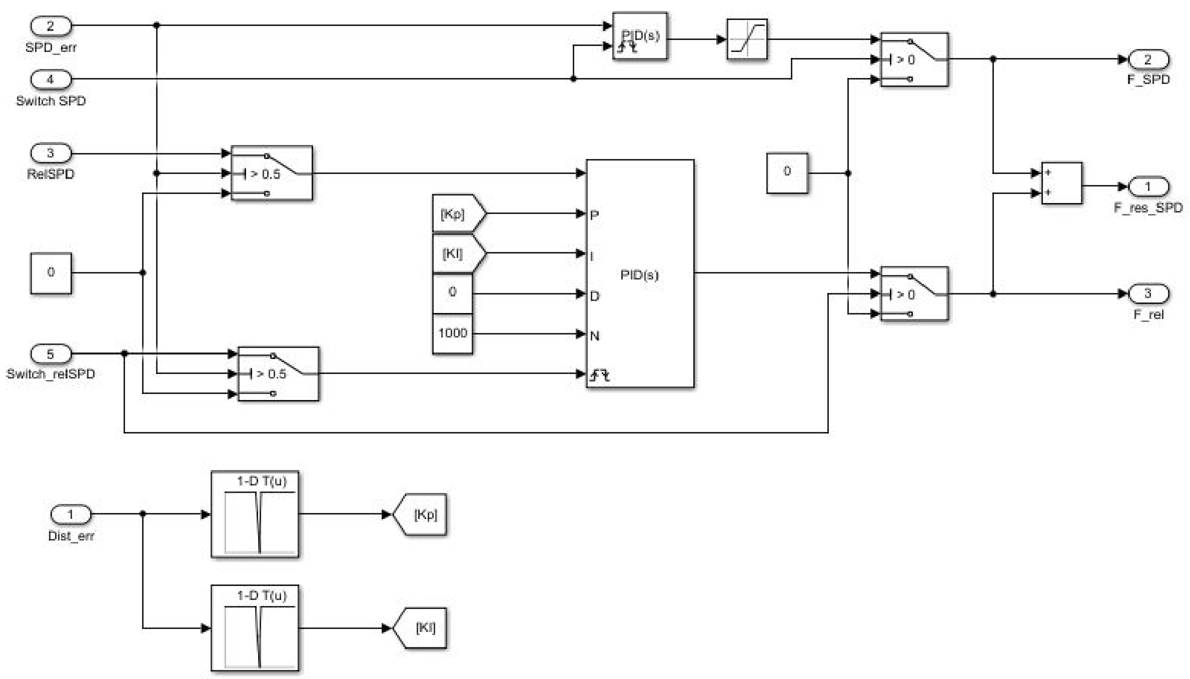


Figure 4.12: Speed and relative speed feedback controllers.

4.2.3. Torque split

The controller output is a torque, which is split based on the vehicle configuration and components. For instance, if the vehicle presents electric motors, capable of generating high braking torque, no mechanical braking demand could be required. For the sake of having a system that represents a generic commercial vehicle, the following tests re made using a vehicle with an internal combustion engine, completely built in VI-CarRealTime. With this model, the torque signal is split between the engine and hydraulic brakes. Hence, positive torque will be entirely provided by the engine, while for the negative one, the residual torque demand that cannot be provided by the engine is requested to the brakes. The split for this vehicle configuration has been implemented in Simulink, as depicted in Figure 4.13. It is then split to front and rear axles, with a ratio of 0.6/0.4. This ratio has been selected because during a braking, due to the load transfer, it is preferable to exploit more the front tyres, to avoid locking the rear ones which would lead to a tail-spin.

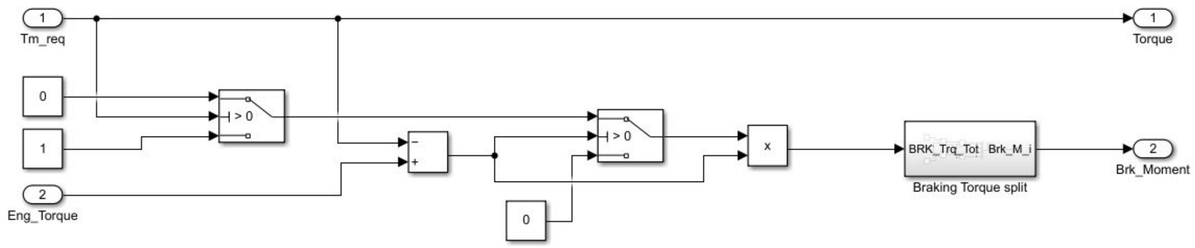


Figure 4.13: Torque split block in Simulink.

4.3. Gains scheduling

PID controllers are implemented defining gains, shown in Equation 4.3. The amplitude of their response is directly connected to them. These gains must be defined based on the application of the system that must be controlled. It is also crucial to understand that FB controllers can modify the stability of a system, so their tuning is essential.

The *Cruise* mode controller gains are kept constant, so they do not depend on the state of the system. This decision is taken to avoid complicating the system without any need. This PID works on set speed from the driver, and is always required to work so as to minimize a relatively small speed error. The only tricky condition is the speed change with the buttons, but a constant gains FB controller can easily manage that because there is the FF contribution which is leading the response in that phase. The Emergency Braking controller gains are constant as well, being defined for a very specific situation.

The *Follow* controller instead depends on the relative speed and distance between the vehicles, so it must be able to react to changes in the environment. For this reason the proportional and integral gains are defined depending on the distance error with respect to lead vehicle, as reported in Table 4.1. This mode is active in a pretty short range, and the gains are chosen to provide as much comfort as possible for the driver.

Distance error [m]	K_p	K_I
-20	500	70
-10	500	70
-6	400	50
0	200	30
2	300	50
5	500	70
20	500	70

Table 4.1: *Follow* controller gains.

The *Follow* mode is meant to trail behind the lead vehicle smoothly and in a comfortable way. The *Safety-critical* mode, as the name suggests, is instead supposed to avoid any accident and prioritize safety, even though comfort must not be totally neglected. This mode must also take into account many possible situations, like another vehicle moving in front of the ego vehicle, or the driver overtaking and so on. Hence, a more complicated set of rules is defined for this controller, to take into account all the possible scenarios, using a Fuzzy Logic control as in [20].

4.3.1. Fuzzy logic control

Conventional Feedback control systems are based on mathematical models described by one or more differential equations, defining systems inputs and outputs, or using the state-space representation to describe their property and behavior. On the other hand, the strength of the Fuzzy Logic Control(FLC) is that no characterizations of the nonlinear and often non observable vehicle dynamics are required. This logic is based on how expert the designer is about the system, that must know how to implement a set of empiric rules and the range of values for inputs and outputs. A FLC is conventionally composed by four blocks: a fuzzifier, a fuzzy rule base, an inference engine and a defuzzifier. The fuzzifier has the role of transforming raw data into linguistic values. For instance in case of an ACC, a value of range equal to 100 meters, will be read as "very far". This evaluation is subjective and is defined by the logic designer, that must understand what the numerical data represent. Fuzzy control rules are composed by many IF-THEN conditions, in which preconditions and consequences are linguistic variables. In the common case of multiple-inputs-single-output(MISO) systems the general rule is:

$$R^i : If\ x\ is\ A_i, ...,\ and\ y\ is\ B_i,\ then\ z = C_i,\ i = 1, 2, ..., n \tag{4.4}$$

Where x, y, z are linguistic variables and $A_i, B_i\ and\ C_i$ are their linguistic values. The inference engine has the role to replicate the human decision, with approximate reasoning to reach the desired control action. In the design of this ACC system, the most common interference method is used, the Mamdani's minimum operation logic[13]. If the inputs are fuzzy singletons, namely, A = u0 and B = v0, then the results C are given by:

$$R_C : \mu_{C'}(w) = \bigvee_{i=1}^{n} [\mu_{A_i}(u_0)\ \wedge\ \mu_{B_i}(v_0)]\ \wedge\ \mu_{C_i}(w) \tag{4.5}$$

Lastly, the defuzzifier is used to yield a non-fuzzy decision from a fuzzy control action. In this model the center of area (COA) model is used.

$$z^*_{COA} = \frac{\int_z \mu_c(z) z dz}{\int_z \mu_c(z) dz} \tag{4.6}$$

These operations are defined and performed in the *Fuzzy Logic Designer* in MATLAB.

The fuzzy control rules for the ACC are based on human experience, and are composed of 25 simple rules. The two inputs are the relative distance error, and the relative speed with respect to the leading vehicle. Both inputs are fuzzified into five linguistic variables, as well as the output, that is the proportional gain of the distance controller. Doing so, the fuzzified PID is able to account for a multiplicity of situations, adapting its response based on the environment. The fuzzy logic linguistic variables are defined as in Table 4.2. With the system sign conventions, a PL relative velocity means that the ego vehicle is slower than the lead one, this means that the system will not brake much even if the vehicles are close. Indeed, a great advantage of the fuzzy inference system is the clarity of its logic. The control rule matrix is reported in Table 4.3, where it is shown how the system reacts depending on the inputs, based on the desired vehicle behaviour from the driver, defined with rules as in Equation 4.4.

Distance error [m]	Relative speed [m/s]	K_p [N/m]
VF(very far)=[5-13.8]	NL(negative large)=[-10- -6]	VL(very low)=[0-10]
F(far)=[13.3-24]	NM(negative medium)=[-6.6- -1.5]	L(low)=[9-80]
NF(not far)=[23.5-29.8]	ZE(close to zero)=[-2.5-2.5]	M(medium)=[70-153]
C(close)=[29.2-37.4]	PM(positive medium)=[1-5.4]	H(high)=[134-234]
VC(very close)=[40-50]	PL(positive large)=[5-10]	VH(very high)=[250-300]

Table 4.2: Fuzzy linguistic variables.

The decision to use a fuzzy control logic to define the controller gains, instead of defining the longitudinal force, has been taken to have coherence with the other controllers, knowing the optimal gain range and response of the PID controller. In literature fuzzified PID already showed good results for ACC applications, where the non-linearities of the system are implemented, with the use of fuzzy membership functions, on the output surface of the system[4].

Distance error / Relative speed	VF	F	NF	C	VC
NL	L	M	H	H	VH
NM	VL	L	M	H	VH
ZE	VL	L	L	M	H
PM	VL	VL	VL	L	L
PL	VL	VL	VL	VL	VL

Table 4.3: Fuzzy rules matrix.

Triangular membership functions are used in the system, to weigh the inputs and output accordingly to the defined rule set. A qualitative scheme of the fuzzy logic process is shown in Figure 4.14, where the shape and range of the membership functions are depicted. While the inputs membership functions present an almost symmetrical configuration, the output VL, L, M, H cases presents much lower values, whit respect to the VH case. This because all the small area cases are defined for comfort, while the last one is a safety condition, of course less extreme than the EB situation.

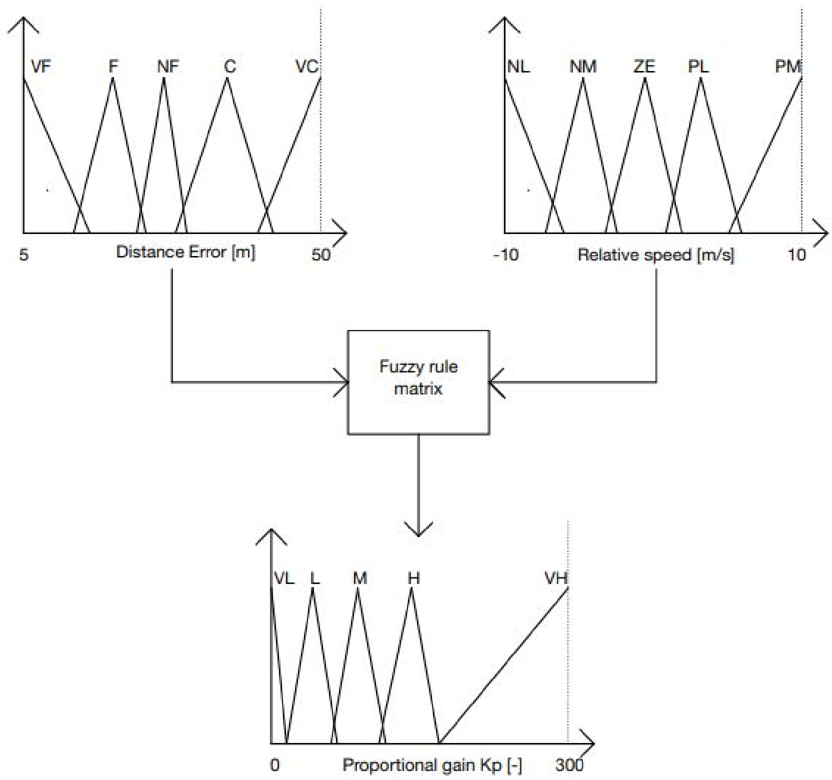


Figure 4.14: Fuzzy logic controller.

4.4. Inputs for VI-CarRealTime vehicle model

The whole ACC logic is defined in Simulink, although, as already mentioned, for the offline simulations the vehicle model is provided bi VI-CRT through an S-Function. This block has as inputs the data computed by the ACC model and returns all the dynamic data from the simulation such as velocities, acceleration, engine torque and so on, that are sent back to the control model. VI-CarRealTime S-Function has its own Input/Output ports, that are used to interface the ACC logic with the vehicle model.

As already mentioned, when the ACC is active, the control input is a torque, that in the proposed study is split between engine and braking torque. Hence, the port used is *Engine.engine_trq*, that represents the actual torque in Nm provided by the engine. In normal conditions the engine demand is regulated by the throttle provided by the driver. To overwrite this command, the port *Engine.control_mode* is used. This port has a switch that sets its value to 0 when the ACC is active, to control the engine in torque mode, and to 1 when the ACC system is not active, to use the throttle demand to regulate the engine. The excess of braking torque is requested to the brakes, using the channel *Brake_System.brk_mmag_sign.L1/L2/R1/R2*, that is the brake moment magnitude in Nm, for each wheel. As shown in Figure 4.15. The torque channels have switches to set their values to 0 when the ACC is not active. This because even though the engine control is set to throttle mode, this torques would be seen as an additional request. Hence, in normal driving conditions, the vehicle would accelerate on its own because of this torque request, while it should only move according to the driver demands sent by the pedals. The other inputs necessary for the simulation are the *Driver_Demands*, that represents the demands provided through pedals and steering wheel.

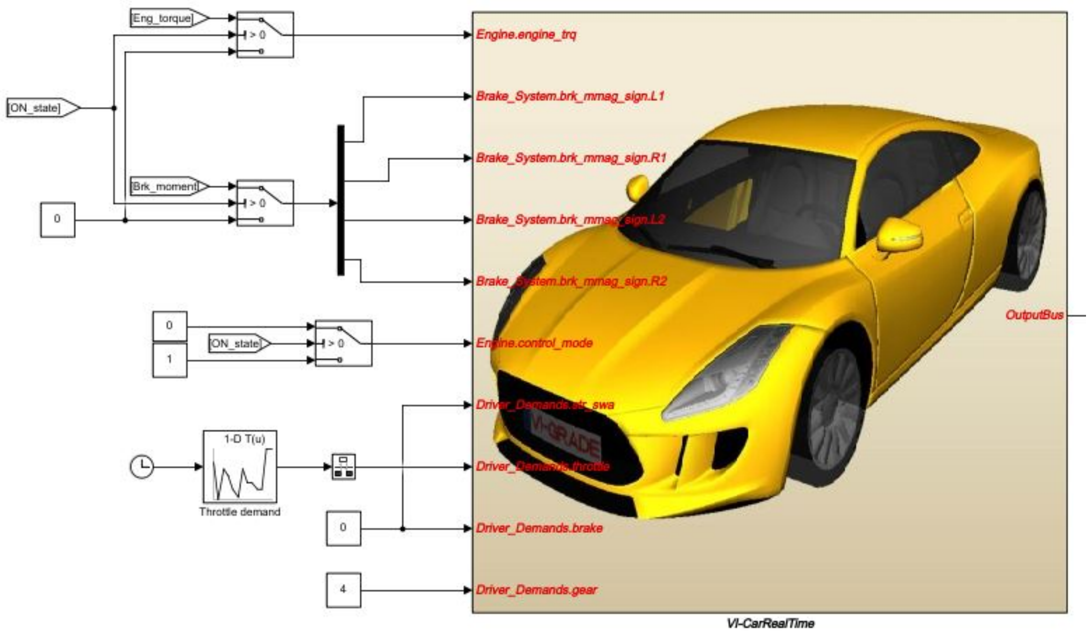


Figure 4.15: VI-CarRealTime S-Function and inputs from ACC system.

4.5. Adaptive Cruise Control comparison model

The proposed ACC model is compared with a state of the art Adaptive Cruise Control model.

This system presents two modes, the conventional CC, and a relative distance controller when a vehicle is detected in front of the ego. The target distance is defined using a constant time headway policy with four tuning variants. The used sensors are a radar and two cameras, to recognize the distance and the lanes, in the same way as in the proposed model. In this system though, the controller acts on the throttle and brake demands, to rule the vehicle behaviour, "overwriting" the driver demands. The Cruise Control is a speed PI controller that has as a reference the set speed. Its integral component is reset every time the system is activated or deactivated. The distance controller is instead a proportional controller.

This model has been used to compare the performance of the built system with a state of the art model, built for the Driving Simulator environment.

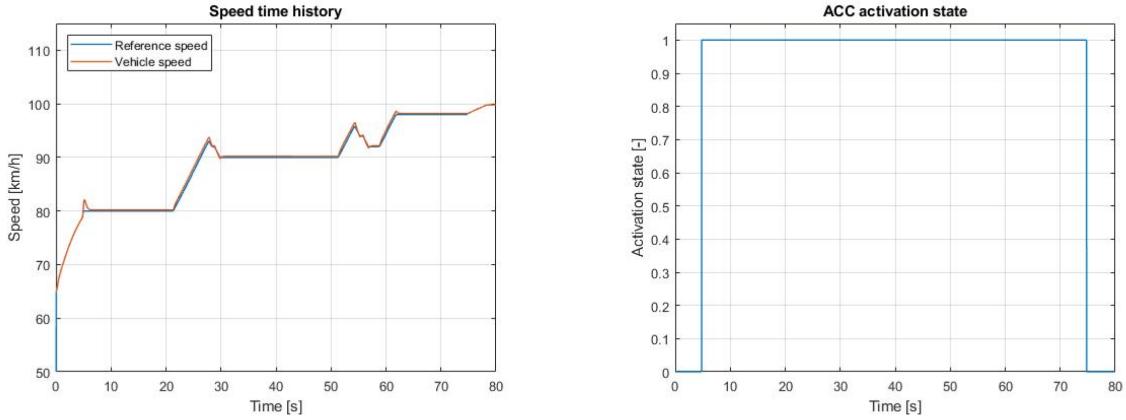
5 | Offline simulations

The first step for the model evaluation are offline tests. VI-WorldSim presents the possibility to perform simulations defining scenarios and agents behaviour. Through these simulations, it is possible to test all the functionalities of the ACC model in a wide variety of situations. This allows to deploy the model on the dynamic driving simulator only once its correct functioning has been verified, saving time and effort.

In this chapter, the results of some offline simulation tests are presented, to verify all the functionalities of the system and the tuning of its parameters.

5.1. Cruise Control tests

The first function to be tested is the Cruise Control. In this case the simulations are much simpler, because no lead vehicle is involved, so any simple scenario is sufficient to verify this feature. The vehicle accelerates based on the throttle signal until the CC activation button is pressed. When this happens, the vehicle maintains the speed until the system is active, and when the buttons to change the desired speed are pressed it reacts accordingly. In Figure 5.1a it is possible to see this behaviour, in fact the vehicle accelerates until the CC system is activated at 5 seconds. From this point on the vehicle keeps a constant speed except when the driver changes the desired speed and the vehicle reacts accordingly. The system is verified to respect the performance requirements, being able to closely follow the reference at steady state and during transients, when the driver pushes the buttons to modify the desired speed.



(a) Speed profils.

(b) System activation state.

Figure 5.1: Cruise Control system simulation outputs.

5.2. Adaptive Cruise Control simulation setup

The ACC model relies on the interaction of multiple agents, hence the behaviour of the lead car and other vehicles must be set as well. The behaviour of the various agents can be defined in VI-WorldSim, as explained in Chapter 2. However, the lead vehicles states are defined in the same Simulink model of the ACC. This decision is taken because in Simulink the coordinates and velocities of the vehicles, with respect to the absolute reference system are easily defined, and data can be checked with minimum effort. For this reason, any agent that has to be controlled is set as *User-controlled* in VI-WorldSim, like the ego vehicle. For the other agents no vehicle model is defined, as they only represent empty shells standing as moving obstacles for the ego vehicle. In fact only their position and orientation are relevant for the simulation, to define the relative distance and speed they have with respect to the ego vehicle. So the inputs sent to the *Vehicle sender* are their coordinates and orientation angles as reported in Figure 5.2. The x and y position are multiplied for a negative gain because VI-WorldSim and VI-CarRealTime have different sign conventions, and the yaw angle is set equal to π for the same reason, to have vehicles heading in the same direction.

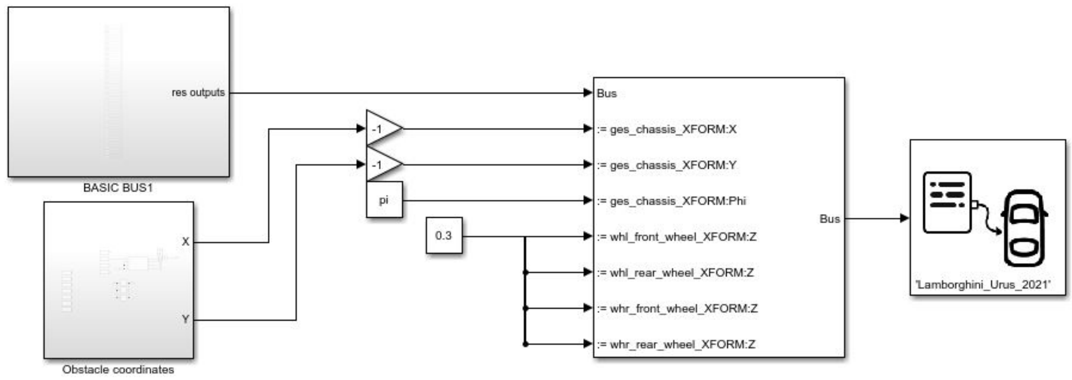
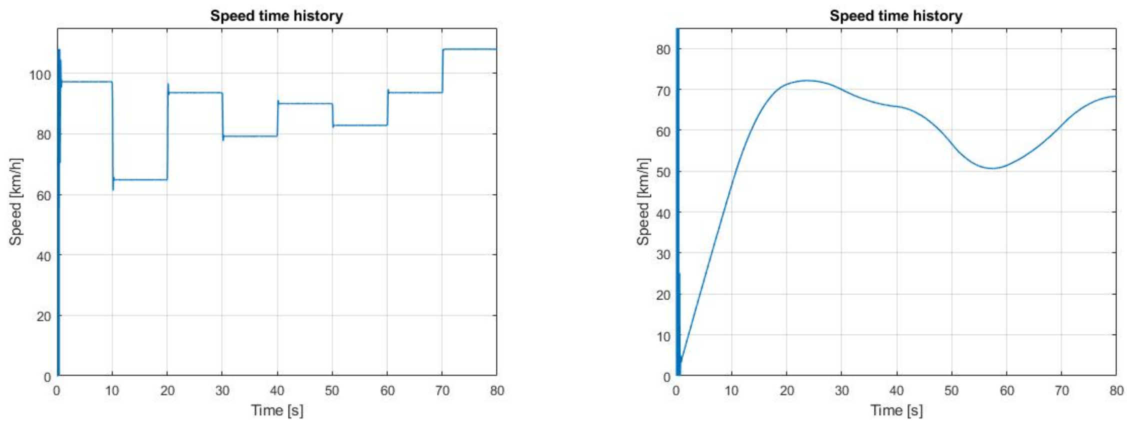


Figure 5.2: Typical lead car vehicle sender.

The issue with providing the coordinates, for instance as a vector defined over time, is that VI-WolrdSim will compute the vehicle speed as a step function, because the car has no inertia. This defines a non-realistic simulation, as shown in Fig 5.3a. Of course, using a complicated vehicle model to define the agents dynamics, uselessly slows down the simulation, as no particular data from these cars is needed. Hence, a one degree of freedom model for the vehicle longitudinal dynamics has been used to generate the coordinates of the obstacle vehicles. This allows to obtain realistic speed profiles to test the model, as reported in Figure 5.3.



(a) Speed with step profile. (b) Speed with realistic profile.

Figure 5.3: Difference of lead vehicle speed profiles during simulations.

5.3. Adaptive Cruise Control tests

To test the functionality of the ACC model, other agents are added in the scenario, to verify the interaction of the ego vehicle with other cars in the environment. The system has to be tested at least in some basic maneuvers, to ensure that comfort and safety requirements are respected, and that sensors work properly. When these simulations give satisfactory results, the ACC can be tested on the dynamic driving simulator, where more complex maneuvers which well represent real-life applications can be replicated, having the driver the control of the vehicle.

5.3.1. Maneuver A

The first test is in the simplest possible scenario, where a vehicle is traveling in the same lane as the ego, travelling at a lower speed. To simplify the discussion, this scenario is called maneuver A and its schematics is reported in Figure 5.4.

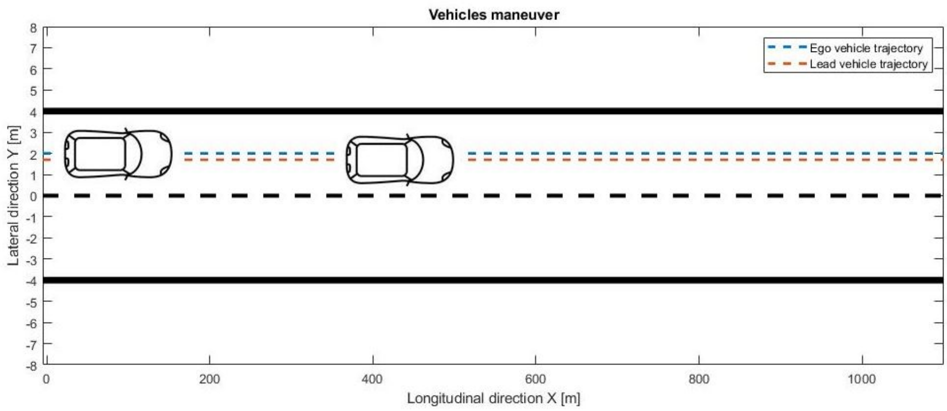
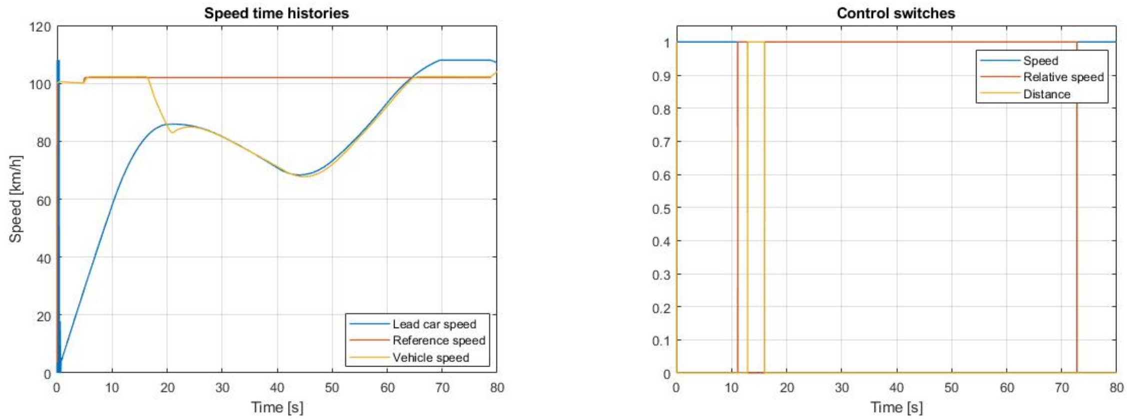


Figure 5.4: Schematics of maneuver A.

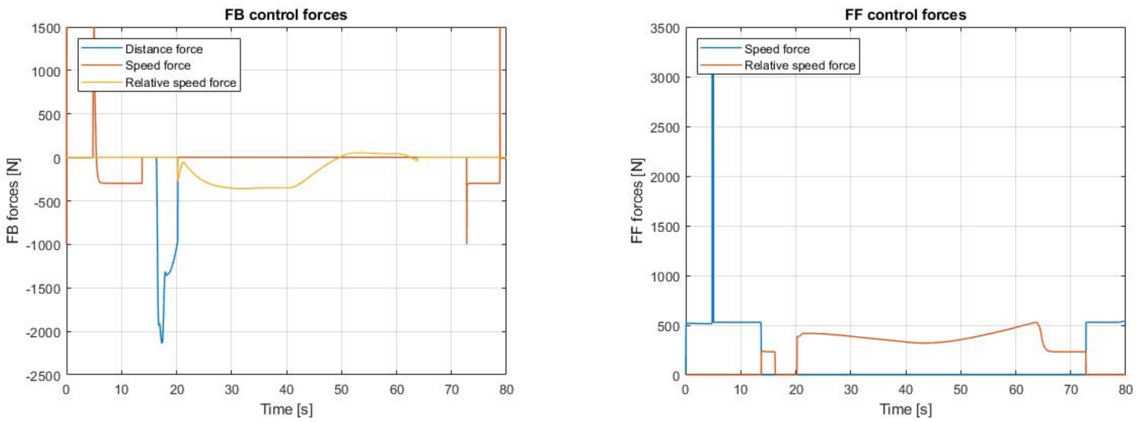
The ego vehicle travels at the set speed until the lead vehicle is detected, then the system has to brake when the obstacle reaches the minimum allowed relative distance, to avoid a collision. After braking, the ego car follows the same speed of the lead vehicle. Finally, when the lead car exceeds the Cruise Control set speed, the model stops following it. All the mentioned phases can be appreciated in Figure 5.5a, while, in Figure 5.5b also the alternation of the control modes is visible. At first, the controller brakes in distance mode, then when a large enough safety distance is reached, the ego vehicle follows the relative speed of the lead vehicle. Finally, the controller returns to work as a Cruise Control.



(a) Speed profiles. (b) Control mode switches.

Figure 5.5: ACC test during maneuver A.

The feedback and feedforward controllers generate two force contributions, where the alternation of the different modes is can also be appreciated if looking at Figure 5.6. The total force that gives the control torque is the sum of these two contributes. When the *safety-critical* mode is on, only the feedback contribution is present.



(a) Feedback force contribution. (b) Feedforward force contribution.

Figure 5.6: Control forces from ACC system.

The final controller output is a torque, that is split between engine and brakes. During the first braking maneuver, the engine cannot generate enough braking force, so the excess is required to the brakes. The maximum positive torque which can be developed by the vehicle is regulated by the engine map in VI-CarRealTime possibly not meeting the controller request. Nevertheless, this condition is not critical, being acceleration less important than braking for safety reasons.

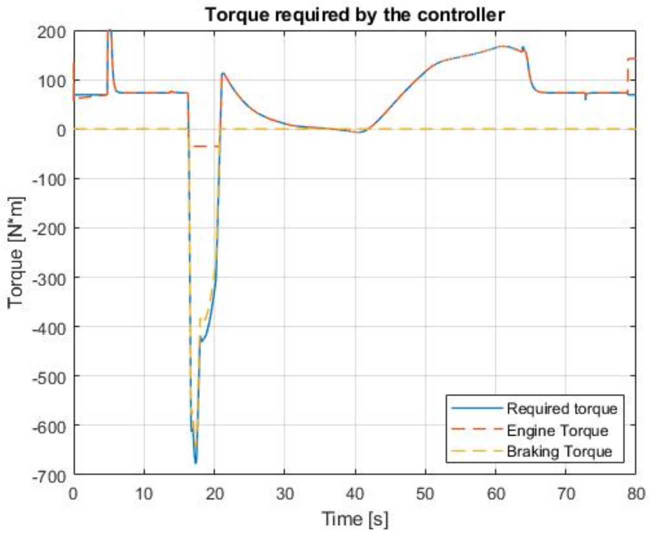
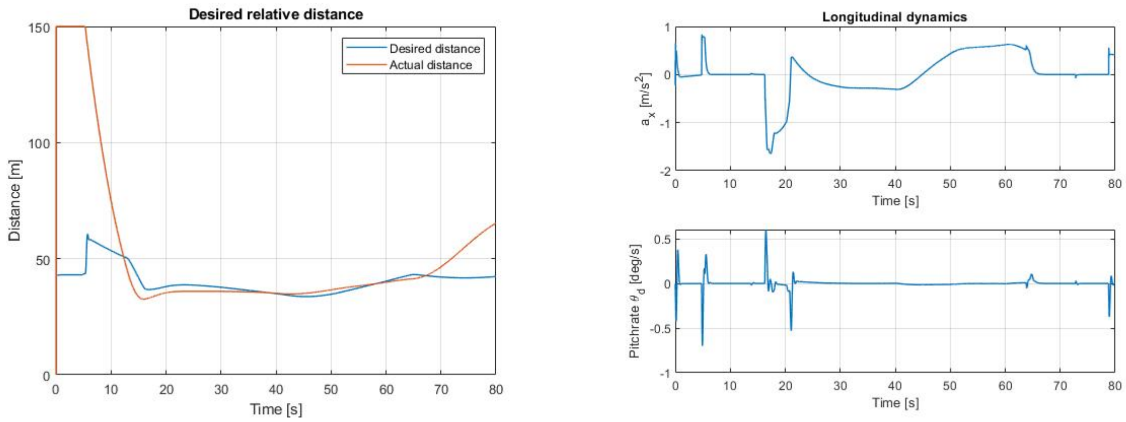


Figure 5.7: Torque signal required by the controller.

The system requirements defined in Section 1.2.1 are respected, as reported below. Looking at Figure 5.8a, it is possible to see the ego vehicle approaching the lead vehicle, and, when it reaches the desired distance, the controller forces the ego to maintain that value. Moreover, acceleration and pitch rate do not reach high values, meaning that the comfort of the passengers during the various maneuvers is preserved. While looking at safety, the relative distance between the two vehicles never reaches values excessively lower than the desired one, meaning that a wide enough safety margin is guaranteed.



(a) Distance and desired distance from lead vehicle.

(b) Longitudinal dynamics of the ego vehicle.

Figure 5.8: ACC system performance.

5.3.2. Maneuver B1

The next maneuver to be tested is the case where the lead vehicle overtakes the ego on one side and then changes lane, settling in front of the ego vehicle. This situation is really common in any highway driving condition. This maneuver tests the virtual camera functioning as well. Indeed, the sensor has to recognize properly the lanes, and when a vehicle moves in and outside them. The system response is verified as well. In fact, it is really common that the overtaking car comes back in front of the ego vehicle pretty close, so the distance error will be high. Therefore, the controller has to recognize the case where the lead car is faster, so low braking action is required, and vice versa. This maneuver is called maneuver B1 while commenting the results and its schematics is reported in Figure 5.9.

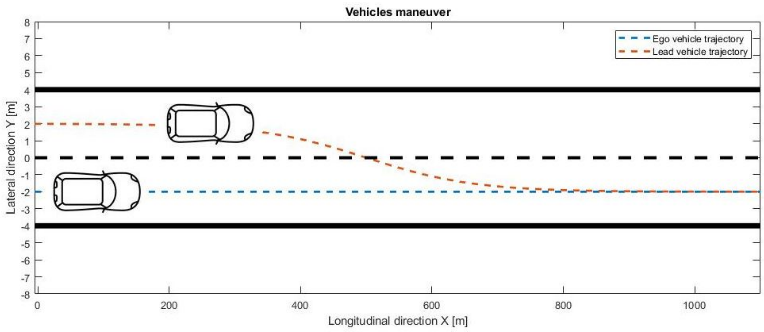
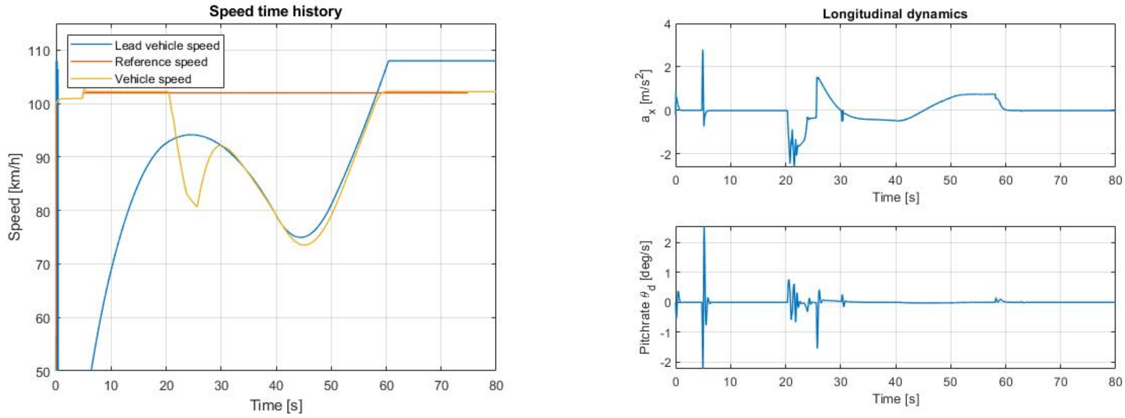


Figure 5.9: Representation of maneuver B1.

The lead vehicle moves in front of the driver quite abruptly, forcing the system to brake. In this situation a quite strong deceleration is necessary, being the lead car even slower than the ego one. In any case, by looking at Figure 5.10b, one can appreciate that the maximum deceleration is around 2 m/s^2, still in a comfortable range. Also the pitch rate shows some notable peaks in the first braking phase, hence there is a chance that the driver can badly perceive this phenomenon. By looking instead at Figure 5.10a, it is possible to notice that, in the first part of the maneuver, the ego vehicle is forced to reduce its speed to increase the space between the two cars. After the first braking phase, the ego follows the lead vehicle at the same speed, like in the previous scenario.



(a) Speed profile during maneuver B1. (b) Longitudinal dynamics during maneuver B1.

Figure 5.10: ACC system performance during maneuver B1.

In Figure 5.11, it is possible to see that, at 20 seconds, the lead vehicle moves in the same lane as the ego. In fact it can be seen that the range drastically reduces forcing the ego strong braking. At first the actual range is lower than the desired one, then the controller reacts and manages to get the ego vehicle around the reference value.

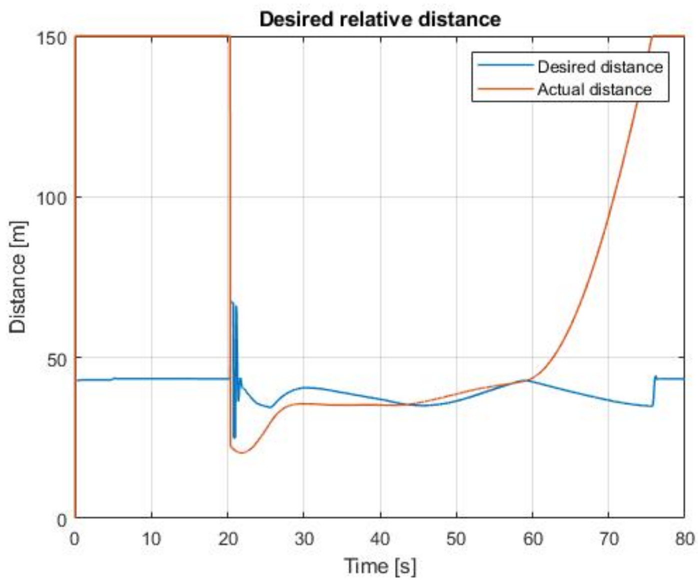


Figure 5.11: Desired distance and actual distance during maneuver B1.

5.3.3. Maneuver B2

To test the tuning of the controllers, a test similar to the B1 maneuver, but with a faster overtaking vehicle is presented. In Figure 5.12 a schematics of the maneuver, called B2 for simplicity, is reported. In this scenario the vehicle in second lane comes from behind the ego, and overtakes it at a higher speed.

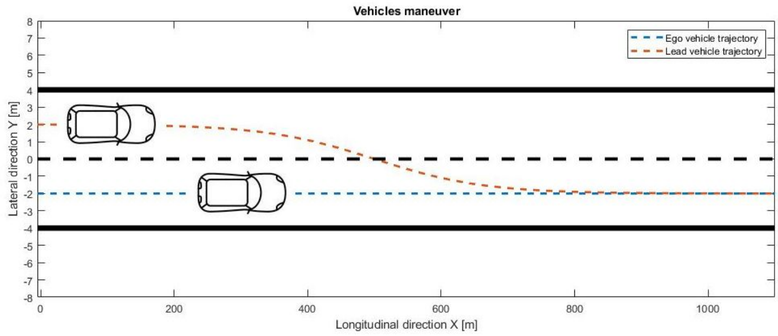
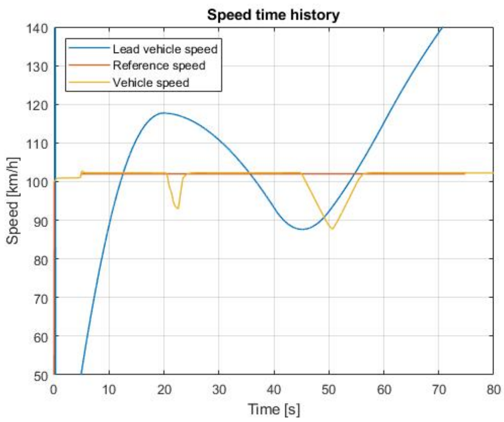
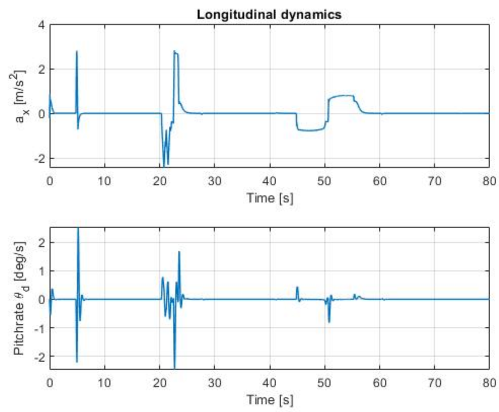


Figure 5.12: Representation of maneuver B2.

In this case the system generates a braking force for less time, since the lead vehicle moves further away on its own, being faster. In Figure 5.13 the dynamics of the maneuver is reported.



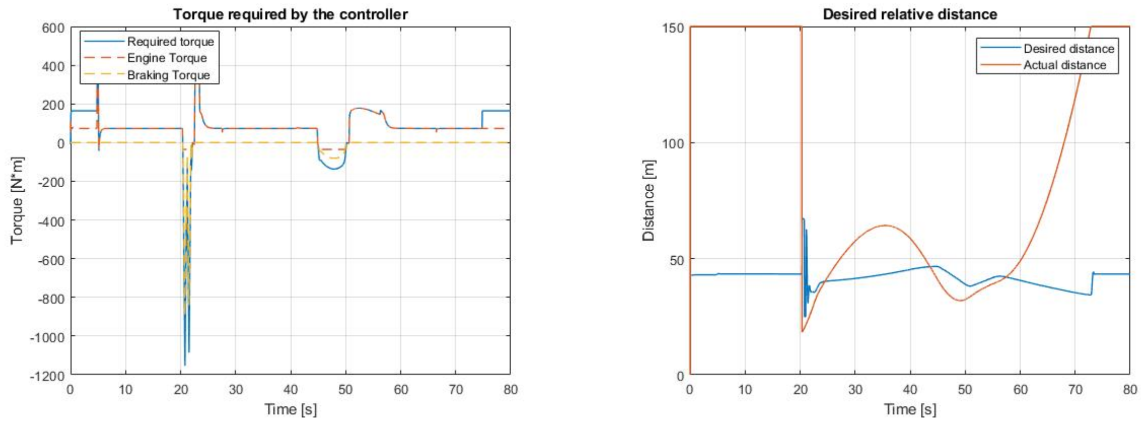
(a) Speed profile during maneuver B2.



(b) Longitudinal dynamics during maneuver B2.

Figure 5.13: Vehicle dynamics during maneuver B2.

The distance once again drops drastically when the lead vehicle is detected in the same lane of the ego, as reported in Figure 5.14a. After the first braking action the system travels at the set speed, being the lead vehicle too fast. For this reason the distance differs from the desired one for a certain trait in Figure 5.14b because the system does not follow the lead car. When the distance gets lower than the desired one, the controllers brake once again to avoid a collision.



(a) Torque requested by the controller.

(b) Distance and desired distance.

Figure 5.14: Torque requested and relative distance during maneuver B2.

5.3.4. Maneuver C

The last maneuver is an overtake as well, but in this situation the lead vehicle moves in front of the ego and then directly in another lane, scenario which can seldom happen in a multi-lane highway. This simulation, schematics is reported in Figure 5.15 and is named as maneuver C in the discussion. This test allows to prove the effectiveness of the ACC logic, but also to verify how the camera reacts to multiple lane changes as well.

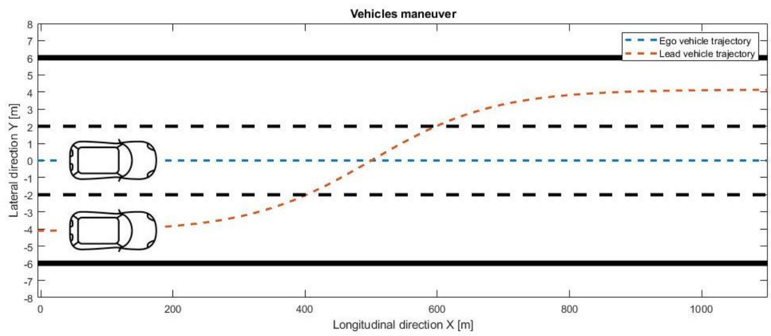
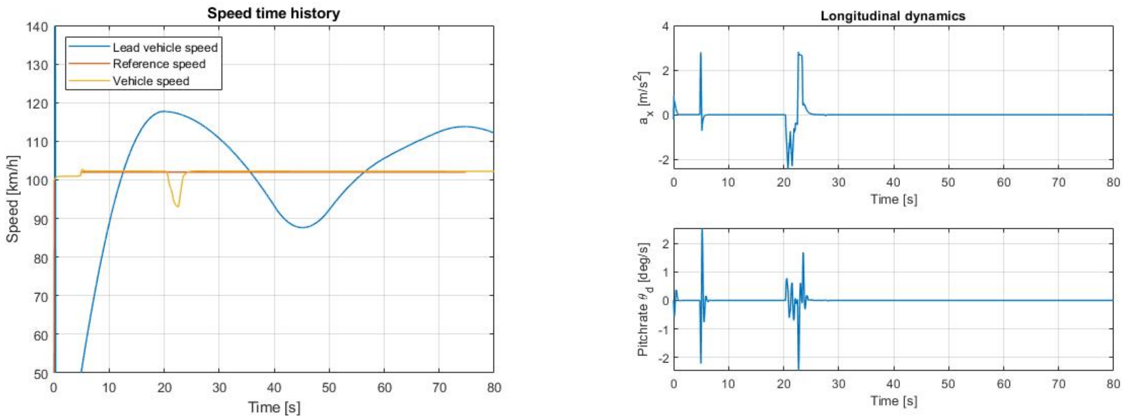


Figure 5.15: Qualitative representation of maneuver C.

There is a braking action when the lead car passes in front of the driver, but then the vehicle follows the speed of the Cruise Control because no car is in front of the driver, this behaviour is depicted in Figure 5.16a. The acceleration is comparable to the one in maneuver B2, as reported in Figure 5.16b.



(a) Speed profile in maneuver C. (b) Longitudinal dynamics in maneuver C.

Figure 5.16: Speed and dynamics in maneuver C.

In Figure 5.17, it is possible to see the range that suddenly drops and then comes back to 150 m, meaning that the camera detects the lead car when it enters the lane and then releases the tracking when the lead car moves to another lane.

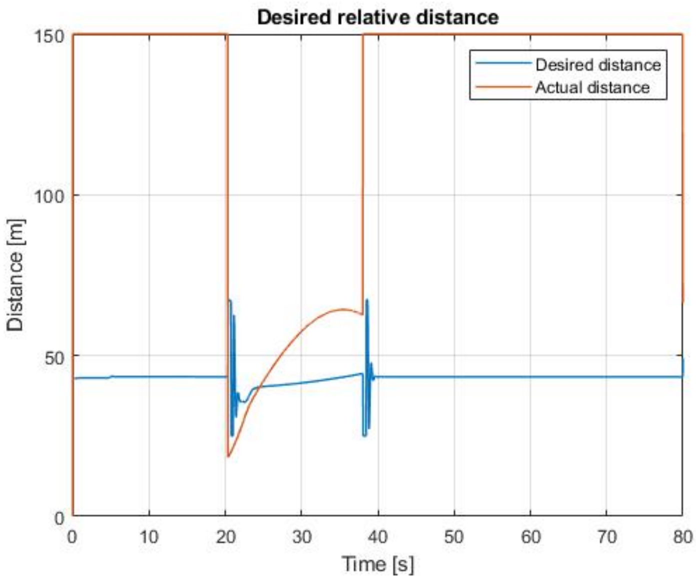
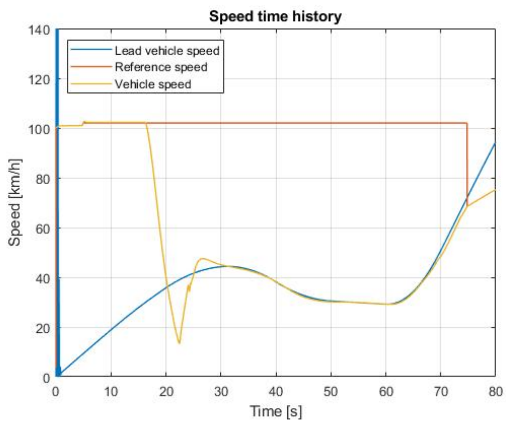


Figure 5.17: Distance from the camera and desired value in maneuver C.

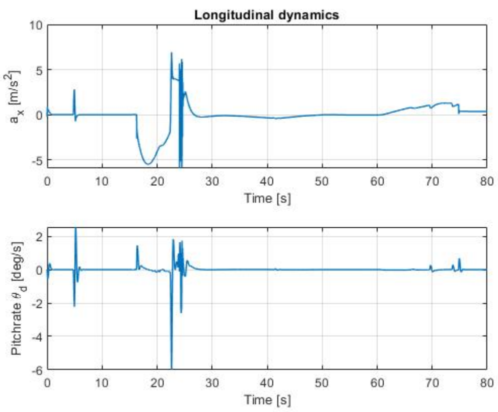
5.4. Emergency Braking

The Emergency Braking is a controller which is activated only once different conditions are satisfied. The obstacle distance from the ego vehicle is not enough to activate this controller, since it is also required that the obstacle is much slower than the ego car as well. The EB action overwrites the ACC controller force to enforce safety in the most critical conditions. To test the EB system some simple offline simulations have been performed, simply creating a scenario with two vehicles in the same lane travelling at different speeds. The tests replicate the case when the driver encounters a vehicle travelling at very low speed, that can happen in case of an engine break down or a tire puncture for instance, or simply in presence of traffic. In those cases the priority is to brake enough to avoid the collision no matter the achieved deceleration.

As reported in Figure 5.18, the deceleration is high and the speed drops as well accordingly. From these plots it is easy to understand that this controller does not have to be triggered in a normal driving condition. In fact in the proposed example the ego vehicle reaches very low speed, that is not safe in an highway scenario with other fast travelling cars, but in this case the braking action was unavoidable to avoid an accident with the lead vehicle.



(a) Speed profile during the EB maneuver.



(b) Longitudinal dynamics during the EB maneuver.

Figure 5.18: Emergency Braking test results.

The controller manages to avoid the collision with the lead vehicle since a non-null relative distance is always maintained, according to Figure 5.19. The strong braking action is necessary to maintain a safe distance between the two vehicles, in fact the relative distance never falls below 20 meters. After the first braking, the system gets back to its usual functioning as an ACC, following the lead vehicle and maintaining the desired distance. In real life applications the driver would probably not follow this vehicle and try to overtake it, if possible. To automatize this maneuver automatic overtake systems are already developed, where, with rear sensors or with V2V communication, it is possible to understand if the other lanes are free to overtake the lead vehicle.[21]

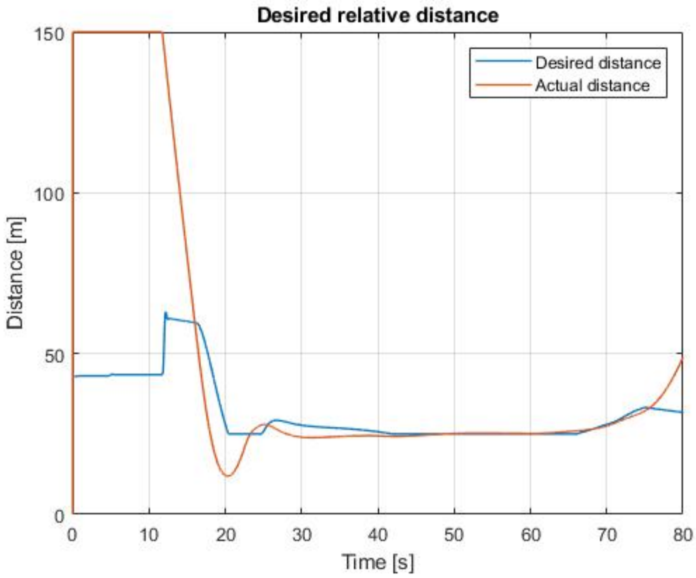
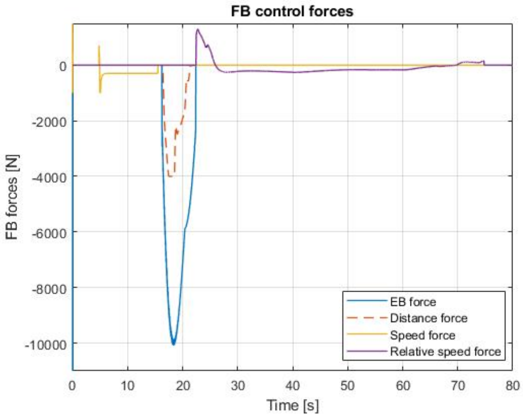
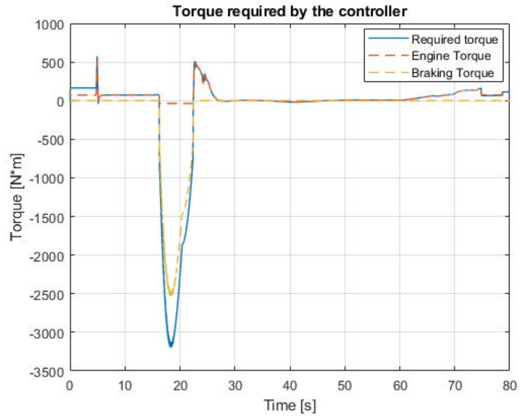


Figure 5.19: Distance in an EB maneuver.

In the plot of Figure 5.20a, the feedback forces generated by the controllers is shown. These forces turn to be much higher than those encountered upt to this point for the ACC testing since the necessary braking action has to be stronger to avoid a collision. In the plot of Figure 5.20a, the longitudinal force from the ACC distance controller is reported as well, to highlight how the EB generates a much higher force. This force is still computed during the simulation but it is overwritten by the EB one, that regulates the vehicle dynamics until the two vehicles are far enough. Looking at the torque plot in Figure 5.20, it can be seen that the sum of the engine negative torque and the braking torque is lower than the required one in module. This is due to the vehicle limits defined in VI-CarRealTime, where mechanical brakes and tyres are defined, limiting the maximum achievable longitudinal force.



(a) Feedback forces during the EB maneuver.

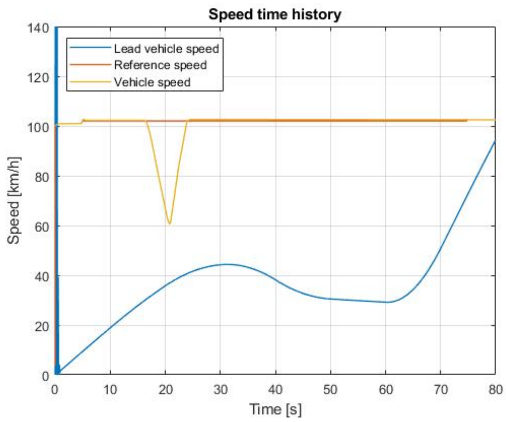


(b) Torque required during the EB maneuver.

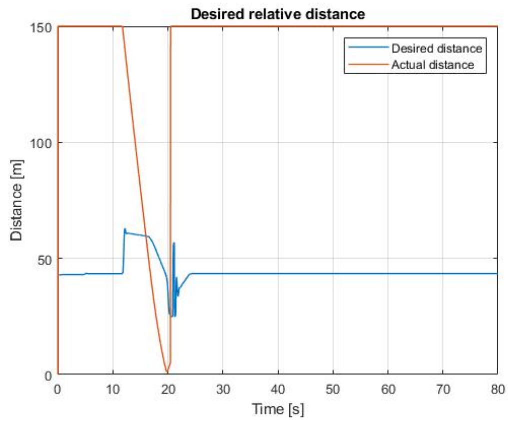
Figure 5.20: Emergency Braking test controller outputs.

To prove the importance of the EB controller the same maneuver has been performed deactivating it, to see if the vehicle would collide or not with the lead one.

In VI-WorldSim offline simulations there is no contact physics, so the ego vehicle instead of colliding with the lead one simply passes through it. For this reason, the range in Figure 5.21b jumps instantly from zero to the maximum value for the camera, being the ego vehicle actually in front of the lead one. This suggest that without the EB controller there would have been a collision in this situation.



(a) Speed profile without EB controller.



(b) Distance with lead vehicle without the EB controller.

Figure 5.21: Simulation results without the EB controller.

5.5. Cooperative Adaptive Cruise Control

In VI-WorldSim to replicate the V2V communication the "State Manager" has been used. This block enables the user to retrieve the agent states, such as position, speed and acceleration, that can be used in the control model. As already presented in Chapter 3, the model retrieves information from two vehicles in front of him, to define the vehicles desired distance and the control model reaction. In VI-WorldSim it is still not possible to have signals like throttle or brake demands, except for the ego vehicle, so this CACC system does not use such information.

5.5.1. Maneuver A

The first simulation includes the ego vehicle and other two predecessors all traveling in the same lane, thus showing a similar situation to maneuver A. In figure 5.22 the three vehicles are represented. The controller defines the desired distance from the first lead vehicle, called d_{ref1} in the figure below, and the relative desired distance between the two lead vehicles, called d_{ref2}. Doing so, the ego vehicle can respond to variation of both vehicle states, ensuring safety, comfort and string stability. To simplify the discussion, the first lead vehicle is called 1, and the platoon leader 2, as depicted in Figure 5.22, the maneuver is called maneuver A.

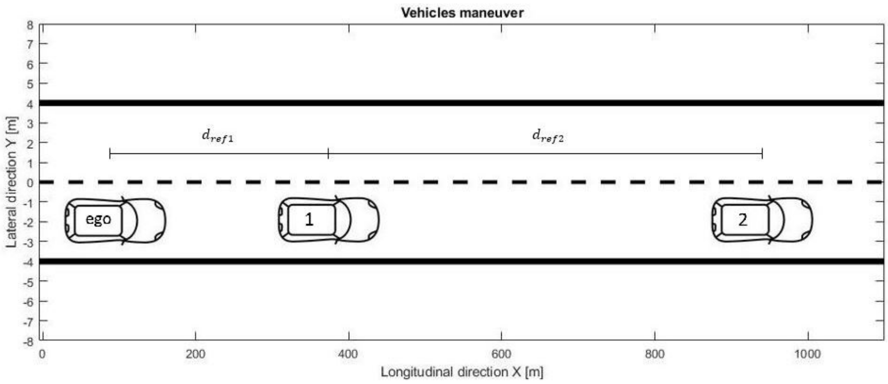
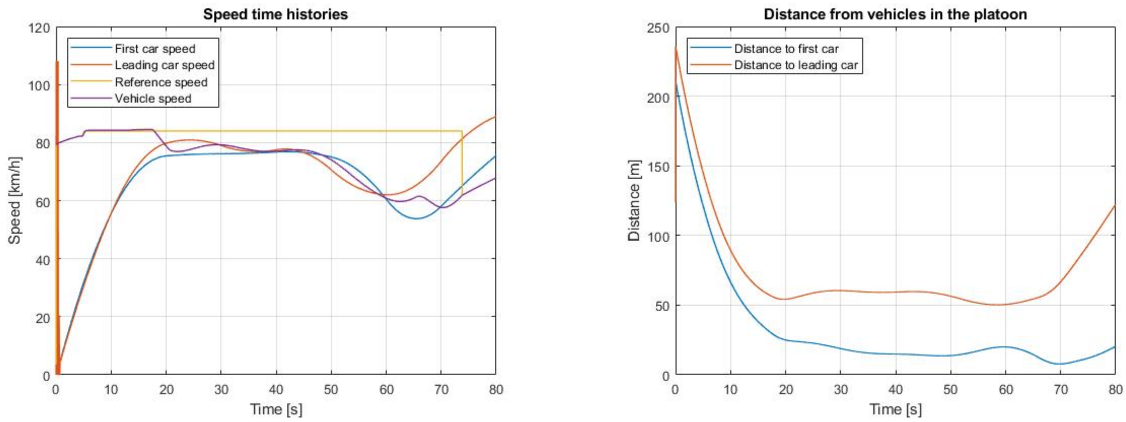


Figure 5.22: Representation of Maneuver A.

Using a CACC system, it is possible to maintain a low relative distance to the first lead vehicle, in this case called vehicle 1, and still ensure safety. This can be appreciated in Figure 5.23b.



(a) Speed profile using CACC.

(b) Distance with lead vehicles using CACC.

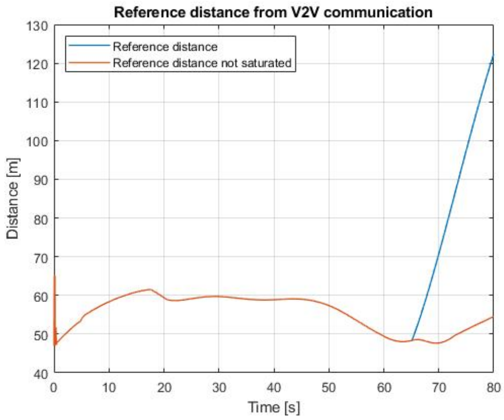
Figure 5.23: Simulation results using CACC system.

The controller has as reference value the desired distance from the vehicle 2. This value is the sum between the desired range between the ego vehicle and vehicle 1, and the desired distance between vehicles 1 and 2, called respectively "Reference distance 1" and "Reference distance 2" in Figure 5.24b, represented in Figure 5.22 as well. Hence, the control system acts on the distance to vehicle 2, and does not directly look at the vehicle 1. For this reason, a saturation of the desired distance value must be applied, otherwise the risk to hit vehicle 1 would be high. The minimum desired distance is in fact defined on the actual distance between the twp lead vehicles.

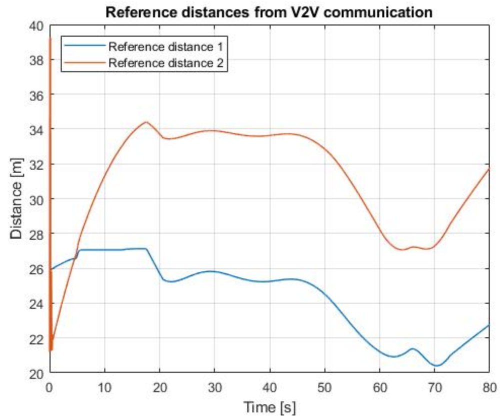
$$d_{min} = q_d * (d_{ref2} - d_{ref1}) \quad q_d > 1 \tag{5.1}$$

d_2 and d_1 are respectively vehicle 2 and 1 distances from the ego vehicle, so their difference represents the distance between the two lead vehicles. In case vehicle 1 hits vehicle 2, a minimum constant value for the relative distance is defined as well, because otherwise the ego vehicle would hit vehicle 1 in this situation. With this definition the distance will never get below a certain value, so the collision with the first lead vehicle is avoided.

At 65 seconds, the vehicle 1 decelerates a lot, and consequently the desired distance from the lead one increases to avoid the collision, as reported in Figure 5.24a. By doing so, the ego vehicle slows down. Without this condition, the controller would keep the desired distance to the platoon leader, but hit the first one that is much slower. The two contributions of the relative distance are shown in Figure 5.24b.



(a) Reference distance from the platoon leader.

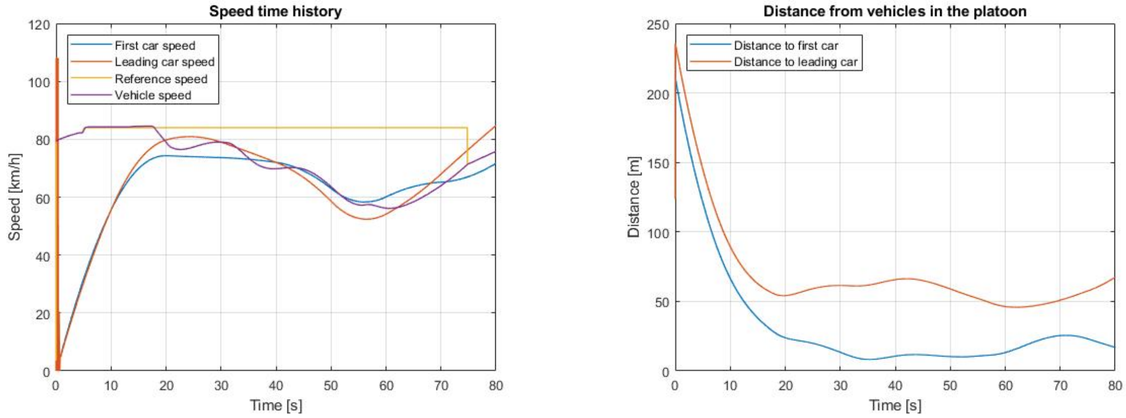


(b) Desired distances from the lead vehicle and the platoon leader.

Figure 5.24: Reference distance computed by the CACC system.

5.5.2. Maneuver B

Maneuver B has the same vehicle configuration as maneuver A, but in this case vehicle 2 slows down, to show if the model reacts to this speed change before vehicle 1 slows down as well. In Figure 5.25a it can be seen that around 60 seconds the ego car brakes to keep a safety distance from vehicle 2. Vehicle 1 will have to brake as well to avoid hitting the leader of the platoon, but when this happens the ego has already left some space between them, knowing that a perturbance is coming upstream. The potential of this system is the fact that it reacts to both vehicles states, and it manages to react to the behaviour of the first predecessor in the platoon knowing what is happening upstream.

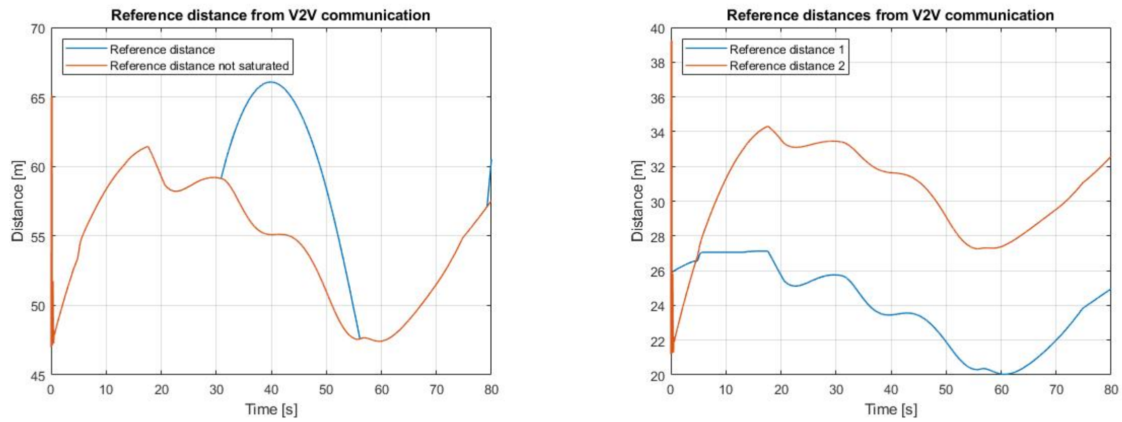


(a) Speed profile in CACC during maneuver B.

(b) Distance from lead vehicles in maneuver B.

Figure 5.25: CACC results in maneuver B.

When the vehicle 1 is too close the desired distance is saturated like in the previous case to avoid a collision, as reported in Figure 5.26a. Instead, when vehicle 2 starts to slow down, the reference distance returns to be the one computed from the vehicle states, to leave the desired range from vehicle 2. The computed reference distanced are shown in Figure 5.26b.



(a) Reference distance from the platoon leader.

(b) Desired distance from vehicle 1 and 2.

Figure 5.26: Reference distance computed by the CACC system during maneuver B.

5.6. Comparison with model from literature

The last offline tests performed to evaluate the model are those comparing the ACC logic designed in this thesis with the Adaptive Cruise Control from literature, which represents a state of the art model that controls the brake and throttle of the ego vehicle These tests have been performed to be identical to the ones done the built ACC system, so at a speed around 100 km/h, and with the same lead vehicle speed profiles.

5.6.1. Maneuver A

The first scenario is the encounter of a vehicle ahead going slower, so the ego car has to slow down and then follow the other one, its schematic is represented in Figure 5.27.

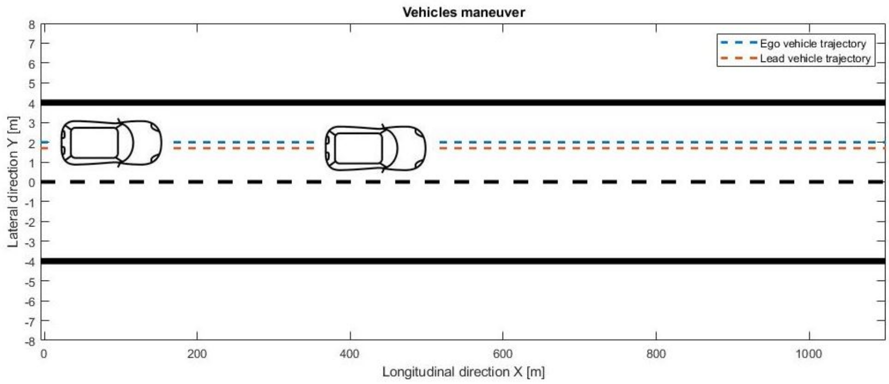
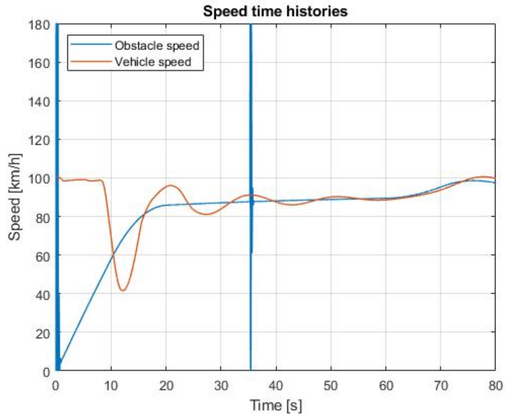
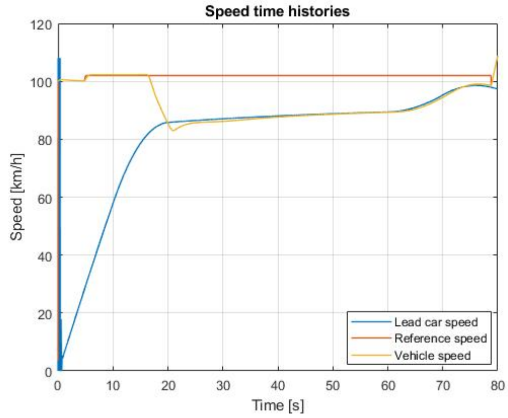


Figure 5.27: Maneuver A to compare the two systems.

In Figure 5.28 it can be seen that this model brakes more with respect to the ACC designed in this thesis, even with the lowest gains configuration. The first braking action is strong, and the vehicle reaches high deceleration value. After that, the speed still oscillates a bit before the ego vehicle follows the lead car velocity. The built ACC model does not show oscillations, and follows the lead car more smoothly. In Figure 5.29, it is possible to see that the example model brakes much more, and the acceleration reaches higher values.

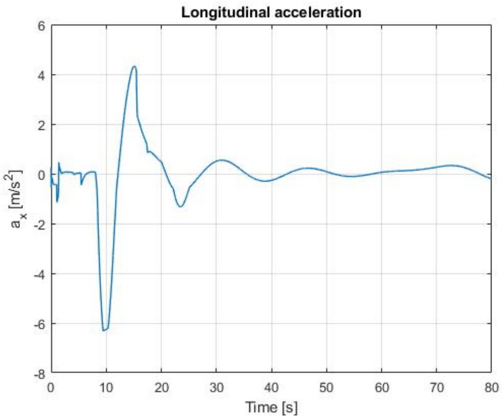


(a) Speed profile using the ACC system from literature.

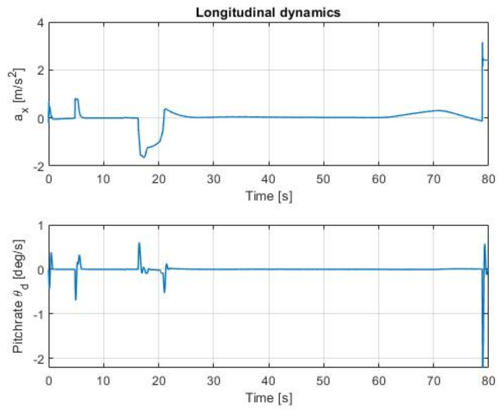


(b) Speed profile using the modeled ACC system.

Figure 5.28: Longitudinal dynamics from ACC system from literature.



(a) Longitudinal acceleration profile using the ACC system from literature.



(b) Longitudinal dynamics using the built ACC system.

Figure 5.29: Longitudinal dynamics comparison.

At this speed the model leaves more or less 80 meters between the two vehicles, an higher value compared to that in the proposed ACC model. It still manages to maintain a constant distance from the lead vehicle, an essential feature for an ACC system.

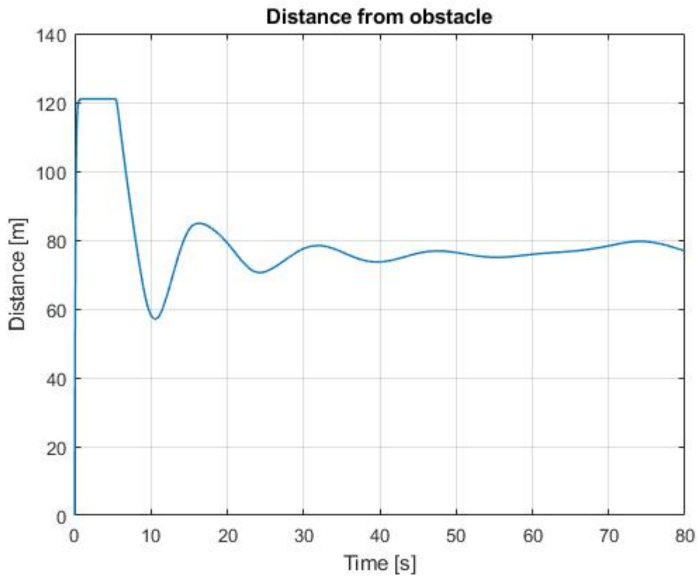
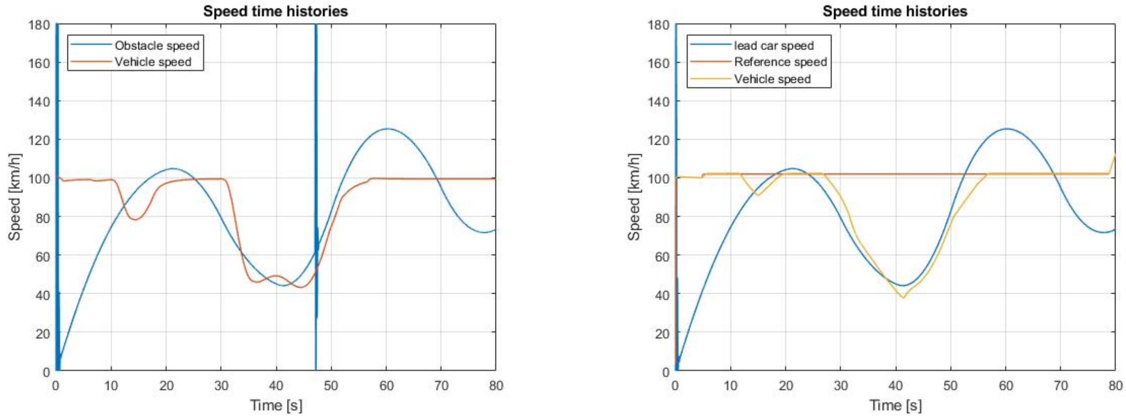


Figure 5.30: Distance from the lead vehicle.

5.6.2. Maneuver B

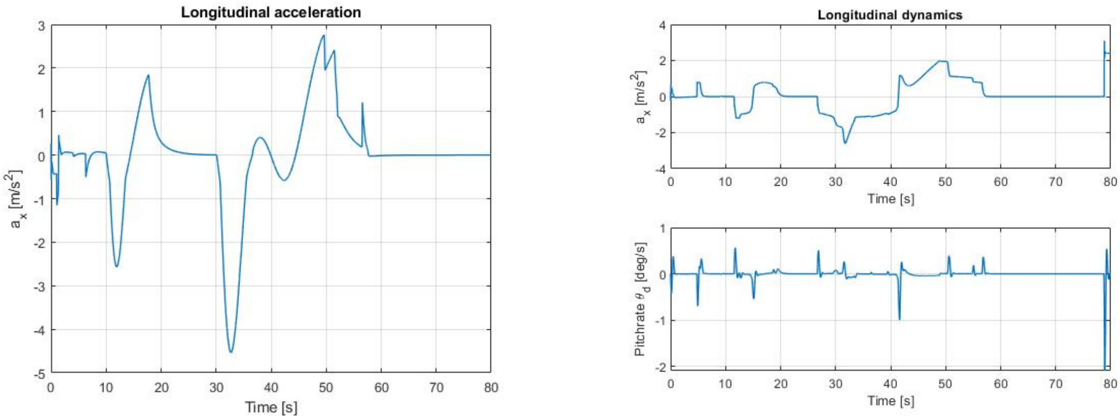
The second simulation shows a strong brake due to a lead vehicle that slows down a lot. The scheme is the same as Maneuver A, but the speed of the lead vehicle is different. Even in this situation the ACC system manages to brake enough and avoid the collision. This test shows that the controller does not follow the lead car anymore if its speed exceeds the set one from the Cruise Control, in this case 100 km/h, reported in Figure 5.31 and 5.31b. In Figure 5.32a, it can be seen that the system from literature reaches higher values in deceleration, while the acceleration and pitch rate for the ACC modeled in this thesis, in Figure 5.32b, show a more gentle braking action.



(a) Speed profile using the ACC system from literature.

(b) Speed profile using the modeled ACC system.

Figure 5.31: Longitudinal dynamics from the simulation.



(a) Longitudinal acceleration profile using the VI-Grade ACC system.

(b) Longitudinal acceleration profile using the ACC system from literature.

Figure 5.32: Longitudinal dynamics from two models in maneuver B.

5.6.3. Maneuver C

The last test is a vehicle that crosses in front of the ego vehicle and then moves away in another lane. Its schematic is represented in Figure 5.33.

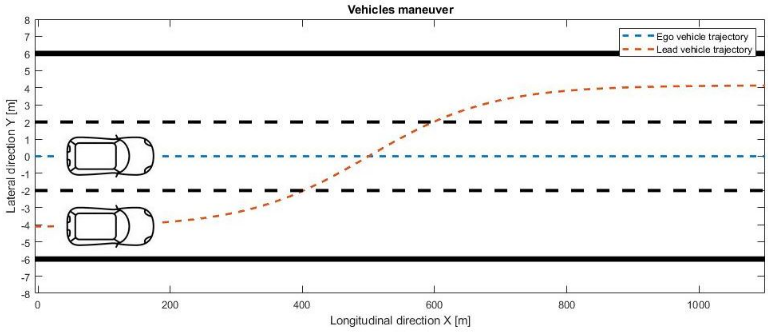
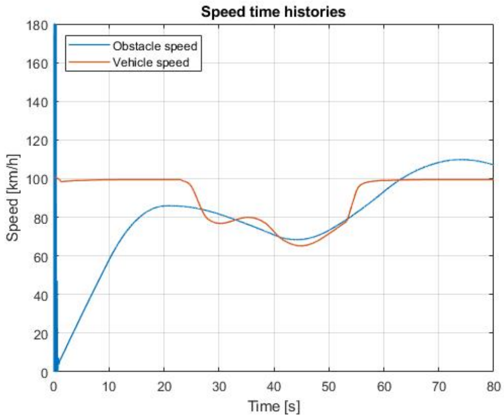
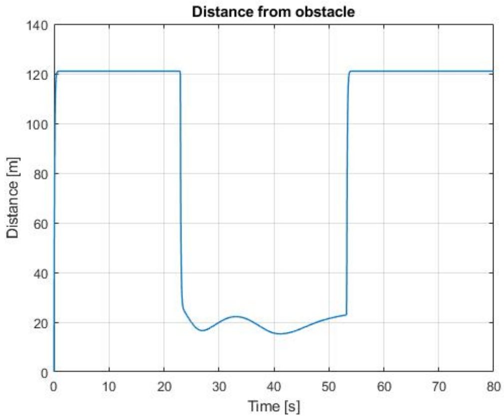


Figure 5.33: Representation of maneuver C.

In this case the system maintains a lower distance from the lead vehicle, reported in Figure 5.34b. Indeed, the controller keeps the two vehicles more or less at 20 meters away from each other. When the lead car leaves the ego vehicle lane the speed gets back at 100 km/h.



(a) Vehicle speeds profile.



(b) Distance from lead vehicle.

Figure 5.34: Literature ACC system performance in maneuver C.

These simulations show that both models satisfy the performance requirements of an ACC system. However, the model from literature has a lower capability to adapt to the situation, and its response seems always stronger than required, showing great deceleration and oscillations, that can bring to passengers discomfort.

5.7. Conclusions

All these tests demonstrate that the ACC model is able to account for the possible situations that a driver could encounter while using such system. The model is able to respond to many unpredictable behaviours of other agents, and react accordingly. The controllers modes and their switching are stable. The data from the virtual camera are reliable, both for obstacle identification and lane detection. Hence the system is verified to work properly.

The final step is to test the model using the Dynamic Driving Simulator. This is crucial so that the driver could understand if the system is comfortable or not. In fact, until now the only way to understand the feedback on the driver was to look at channels such as the longitudinal acceleration or the pitch rate, that do not directly define how the driver is feeling in certain maneuvers. Hence, a testing campaign with real people is necessary to understand if the system is comfortable. With the drivers feedback is the possible to adjust the system parameters to reach high comfort, and ensure safety during the simulations.

6 | Driving simulator tests and model validation

The last step is to test the model on the DiM400 Driving Simulator. At first, the model is tested to verify its correct functioning, in case some error has not been noticed during offline simulations, being the tested maneuvers quite simple. Then multiple gains setups have been tried as well, to check the driver feedback, and also create multiple settings for the system, some comfort oriented, while others more concerned about safety. Finally, some testers have been asked to test the system and gave their feedback with a questionnaire, to evaluate the system.

The Driving Simulator tests comprehend static and dynamic simulations. The first are simulations where the cockpit does not move, but only the images on the screen move according to the vehicle behaviour. These tests are always performed when a new model is compiled on the simulator, because in this way all the theoretical inputs to the physical actuators are calculated but not directly fed to them. This allows to analyze the loads to which the driver would have been subject during the maneuver. If they show to be excessive, the dynamic simulation is not deployed in order to not cause injuries to the tester. Instead, if the model is considered to function well, the simulator can be set in dynamic mode. In this mode the actuators will move the cable-driven disk-frame and the actuators between it and the cockpit as well, to replicate the vehicle body motion. Even the seat help to better feel maneuvers such as long turns and braking, where long-sustained acceleration cannot be provided by the disk-frame motion. The models are deployed directly from the control room PCs and any modification can be applied while testing. This enables the driver to feel how the system performs with different configurations in a really short period of time.

6.1. Evaluation tests

The final step is to evaluate the system through a series of tests. To do so, a number of testers has been selected to try the ACC system and complete a questionnaire to evaluate the model and the driving simulator itself. This method is commonly used to rate systems like the Adaptive Cruise Control, and it is possible to find many examples in the literature. For instance in [9] an online questionnaire is used to ask to a large number of people an opinion about the ACC system, while in [17] and in [3] some driving tests are organized to let people try the ACC system, then the conclusions are drawn by analyzing both testers answers and simulation data. This feedback is essential for the evaluation, since the ACC system is designed to be used in commercial vehicles, hence its most important quality is to result pleasant and safe to both driver and passengers.

6.1.1. Simulations setup

The test is designed so as to obtain the same maneuver for every tester, to have a valid statistic sample of comparable results. The selected environment is a four lanes highway with multiple agents set in "wander" more. Other two cars are set to move in "Reference Trajectory" mode, defined while creating the scenario in VI-WorldSim, so they follow a determined trajectory during the simulation. The trajectory definition is reported in Figure 6.1. This behaviour is defined using end points, VI-WorldSim then automatically interpolates the trajectory between those points. This process is simpler than defining the position as a time vector, because it is directly defined on the road, so the y and z position are known and automatically set. The other agents which are let wander, not interact with the ego vehicle, but are there to create a more realistic scenario, with multiple cars travelling. When testing the ACC systems, the testers are asked to stay in the second lane throughout the entire simulation, and to reach a speed around 120 km/h before setting the ACC. When using the ACC system, it regulates the speed of the vehicle. Doing so, the ego vehicle behaviour is similar in every test, so that the testers can roughly experience the same maneuvers. Every tester has to complete three different simulations:

- In the first test the driver is asked to drive freely. He can use the Cruise Control system or drive the vehicle manually. This test has two purposes, to let the tester try the driving simulator and get used to it, and to record some information about the tester driving style so as to have a baseline on which to analyze in an objective way the judgement about the ACC system.

- In the second test the simulation is performed with the ACC system on. So the tester activates it, as in a real driving condition, and the system regulates the speed of the ego vehicle.
- In the third and last test the ACC system is still active, but with a different set of gains, to have a stronger controller reaction.

(a) End point definition.

(b) Trigger active on the selected vehicle.

Figure 6.1: Reference trajectory behaviour definition in VI-WorldSim.

To have a clearer idea of the performed simulations, in Figure 6.2 the typical test maneuver is shown. The vehicles with the defined trajectory keep moving in front of the ego vehicle, moving in and out of its lane repeatedly. The ACC regulates the ego vehicle longitudinal dynamics by braking when the lead car moves in its lane, and accelerates when it moves away from it. In the first simulation, the testers have been asked to stay behind these cars, and regulate the distance by braking, before overtaking them after they become inactive for the testing purposes.

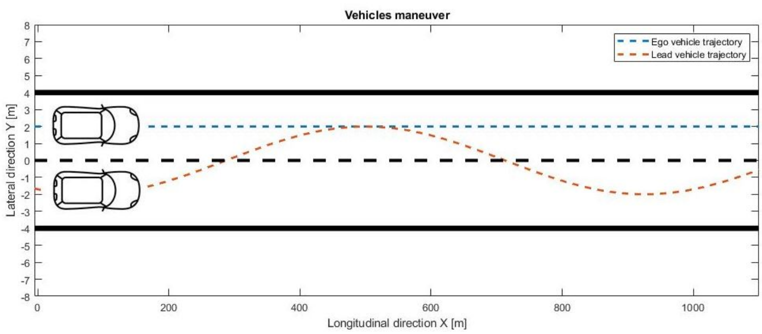


Figure 6.2: Qualitative representation of the ACC DiL simulations.

Before the actual simulations, the testers have been asked to answer to some preliminary questions, to register their age, driving experience, and knowledge and experience about Adaptive Cruise Control systems and driving simulators. The system has been tested by 19 people, both males and females, from 24 to 46 years old. Among these people, seven already used an Adaptive Cruise Control system in a real vehicle, and twelve tried a driving simulator of any type, ranging from desktop to dynamic systems. It is curious to see that of the seven people that tried an ACC, only 2 replied that they commonly use it on the highway, while the other five still prefer to manually drive. The people that never tried an ACC or a driving simulator, have been asked to evaluate from 1 to 5 how willingly they are to try these systems. The answer average regarding the ACC is 4, while for the driving simulator it is 4.86, showing that people are interested and curious about these systems, especially the dynamic driving simulator. These questons are reported in Table 6.1.

ID	Question
1	**Have you ever used an Adaptive Cruise Control (ACC) system?**
2	**If yes, do you normally use it while driving on the highway?**
3	**If no, how willing are you to try an Adaptive Cruise Control?**
4	**Have you ever used a driving simulator? (Specify the type)**
5	**If no, how willing are you to try a driving simulator?**

Table 6.1: Questions on tester experience.

6.1.2. Questionnaire

After the simulations, the tester is asked to reply to a series of questions. The first ones regarding the driving simulator and simulation realism. These questions address both the physical components and the virtual ones of the simulator, the questions are reported in Table 6.2. The testers are asked to give a score on an increasing scale from 1 to 5.

ID	Question
6	**Rate how realistic the simulation is**
7	**Rate the driving simulator realism (cockpit, pedals, steering...)**
8	**Rate the driving simulator scenario realism (speed and distance perception, sounds...)**

Table 6.2: Questions on driving simulator realism.

The next two groups of questions, shown in Table 6.3, are about the evaluation of the Adaptive Cruise Control systems, both from safety and comfort point of view. This series of questions is asked for both models, first for the softer ACC and then for the stronger one. This is the reason why every question has two IDs, because these questions are asked twice. The testers answer these parts after having tried both models, so that they are able to tell the differences and to evaluate each system knowing the comparison with the other one. The testers are asked once again to give a mark, increasing from 1 to 5, except for questions 10 and 19, where the grades meaning is specified in the question.

The last group of questions are reported in Table 6.4. These are meant to compare the three simulations, so instead of giving a score on a 1-5 scale, the testers are asked to chose the model they prefer.

ID	Question
9-18	**Rate the ACC model from a safety point of view (distance from other vehicles, sounds...)**
10/19	**How do you consider the distance to the leading vehicle kept by the ACC model? (1 too close, 3 adequate, 5 too far)**
11/20	**Rate the ACC model from a comfort point of view (accelerations, speed. . .)**
12/21	**Rate the ACC model from a comfort point of view during braking**
13/22	**Rate the ACC model from a comfort point of view during accelerations**
14/23	**How much does the system reduce the driving effort?**
15/24	**If you normally use an ACC, rate how similar this model is**
16/25	**Would you use the ACC model in the proposed scenario instead of manually driving?**
17/26	**Rate how willingly you would use the model in a real driving situation**

Table 6.3: Questions on ACC system evaluation.

ID	Question
27	**Chose the model you consider more comfortable (accelerations, speed. . .)**
28	**Chose the model you consider safer (distance from other vehicles, speed. . .)**
29	**Which ACC system represents more your driving style (aggressiveness, speed. . .)**
30	**Which ACC system you would prefer to have in your car**

Table 6.4: Questions on the three simulations comparison.

6.1.3. Questionnaire results

At first, the questionnaire answers are analysed, to have the drivers feedback on the simulations. Then, some significant data are reported as well, to give an objective interpretation to the subjective evaluations.

In tables from Table 6.5 to Table 6.6, the average score for each question is reported. Of course, for questions where the answer is YES/NO, reported score is the percentage of YES answers over the total.

The answers in Table 6.5 are the first to be analyzed and are those about the driving simulator itself. Before looking at the responses from the testers, it is essential to specify that all the simulations have been performed in static mode. Hence, the simulator lower body and cockpit does not move during the tests. The feeling of movement and acceleration is given by belts tension and seat inflation. This decision comes from the fact that during the tests, the temperature in the simulator room was too high. Therefore, for safety reasons, the simulator could not be used in dynamic mode. For this reason, all the drivers are asked to firmly tighten their belts, so that they could better feel the motion of the vehicle. The results from the tests show high scores anyway, especially regarding the simulator components (Q7). The score of question 6 are a bit lower, probably due to the sensation given by the belts, that for many testers result too strong and not realistic. This may be due to the fact that the belts have been regulated to let the driver feel also lower decelerations, due to the simulation conditions.

ID	Question	Score
6	**Rate how realistic the simulation is**	3.84
7	**Rate the driving simulator realism (cockpit, pedals, steering...)**	4.11
8	**Rate the driving simulator scenario realism (speed and distance perception, sounds...)**	4.00

Table 6.5: Answers on driving simulator realism.

The results about the ACC models evaluation are reported in Table 6.6. To compare the two systems, the scores for the softer controller are reported under the "Test 2" column, while those for the stronger one are reported under column "Test 3", so they will be called accordingly during the discussion. From the results it is clearly visible that, on average, the testers prefer the system with lower gains, that controls the vehicle less aggressively. This system shows higher scores from a comfort point of view, as expected, both for acceleration and braking conditions. A curious result is that the system in Test 2 is evaluated better from a safety point of view as well. This means that the drivers feel stronger braking as more dangerous, rather than considering the system capable to brake more to avoid a collision. Both systems are evaluated similarly when looking at the relative distance and driving effort reduction. This result makes sense, since both systems work in the same way, and have the same reference distance definition, that is considered adequate. The Test 2 system is also considered more similar to commercial Adaptive Cruise Control systems, from the testers that have already tried it. When the testers have been asked if they would use the ACC model instead of manually driving, so Q16 and Q25, the results show once again a preference for the Test 2 model. Among the 19 testers, 7 replied that they would prefer to drive manually rather than use the Test 3 system, while for Test 2 model only 2. These two people replied NO to both questions 16 and 25, showing that they prefer to driver manually rather than using an assistance system like the Adaptive Cruise Control.

ID	Question	Test 2	Test3
9/18	**Rate the ACC model from a safety point of view (distance from other vehicles, sounds...)**	4.42	4
10/19	**How do you consider the distance to the leading vehicle kept by the ACC model? (1 too close, 3 adequate, 5 too far)**	2.95	2.89
11/20	**Rate the ACC model from a comfort point of view (accelerations, speed. . .)**	4.42	3.74
12/21	**Rate the ACC model from a comfort point of view during braking**	3.95	3.11
13/22	**Rate the ACC model from a comfort point of view during accelerations**	4.47	4.05
14/23	**How much does the system reduce the driving effort?**	4.11	4
15/24	**If you normally use an ACC, rate how similar this model is**	3.67	3.2
16/25	**Would you use the ACC model in the proposed scenario instead of manually driving?**	89.47%	63.16%
17/26	**Rate how willingly you would use the model in a real driving situation**	3.79	3.42

Table 6.6: Answers to ACC systems evaluation.

Finally, the answers to the questions regarding the three simulations comparison are reported. In Figure 6.3, the number of preferences for each system is represented. 1, 2 and 3 represent the three tests, so normal driving, softer ACC and stronger ACC respectively. Regarding comfort, it is visible in question 27 that most of the people preferred the softer ACC, while some still finds more comfortable to drive autonomously. When looking at question 28, the system considered safer by the testers is the second one. Between the drivers, 13 recognize the softer ACC as the more similar to their own driving style, while 6 the stronger one. However, only one person voted for the third model as its favorite, confirming that the testers preferred the model with lower gains.

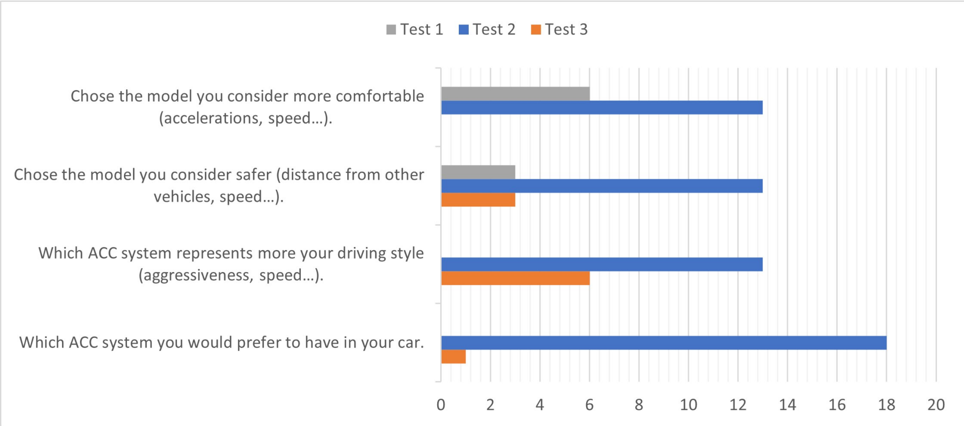


Figure 6.3: Answers to questions from 27 to 30.

6.1.4. Simulation results

At this point the data from the simulations are analysed, firstly the maximum positive and negative longitudinal accelerations. These data are the principal indicators for the comfort level, since the simulations have been static, hence other quantities like the pitch rate are not perceived by the driver and thus cannot influence his comfort. In Table 6.7 the average and maximum acceleration values from the three simulations are reported. The average is not computed on the complete driving, but the considered data are only those for which the ego vehicle has a car in front of it. Hence, these results represent the vehicle dynamics while following another car, first being regulated by the driver, then by the Adaptive Cruise Control systems. The acceleration phase is quite similar in all the three simulations, with the ACC that accelerates more at first but then the average is lower than the first drive. The ACC brakes much more than the driver in both configurations, which is evident from both the average and the maximum value. This because the Adaptive Cruise Control forces the two vehicles to stay at a defined distance, so when the lead vehicle moves in front of the ego vehicle it brakes quite a lot. Instead, the driver is not forcing itself to keep a predefined distance form the vehicle in front but regulates it in a wide range and thus decelerates more slowly.

	Test 1	Test 2	Test 3
Average acceleration $[m/s^2]$	0.32	0.28	0.30
Maximum acceleration $[m/s^2]$	1.27	1.43	1.45
Average deceleration $[m/s^2]$	0.43	0.53	0.58
Maximum deceleration $[m/s^2]$	1.79	4.52	4.90

Table 6.7: Average results during the simulations.

In Figure 6.4 and 6.5, the maximum acceleration and deceleration for each tester are reported. Both in manual driving mode and with the two ACC models, the behavior is similar for almost all testers because they reached the limitation of the vehicle during the initial speed buildup in a predefined gear and starting at a predefined speed. Regarding Test 1 results it is possible to see different driver behaviours, some more aggressive and some less, especially in the braking phase. The maximum deceleration results actually show that Test 2 system, with softer gains, brakes more than Test 3 in some simulations. This means that in those simulations the lead vehicle changes lane when it is closer to the ego vehicle, so the ACC brakes more. However, the average values and the questionnaire answers show that this system feels more comfortable.

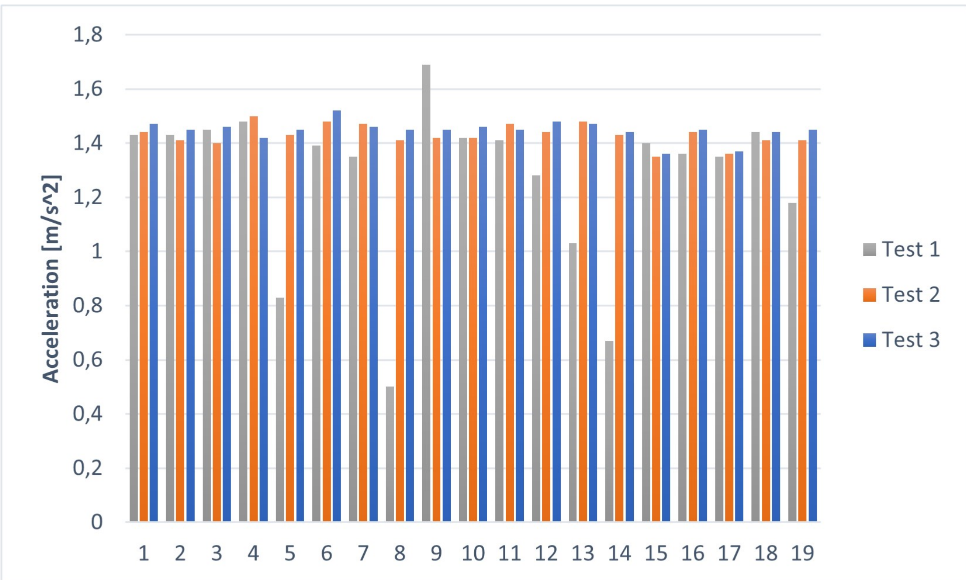


Figure 6.4: Maximum longitudinal acceleration for the different tests.

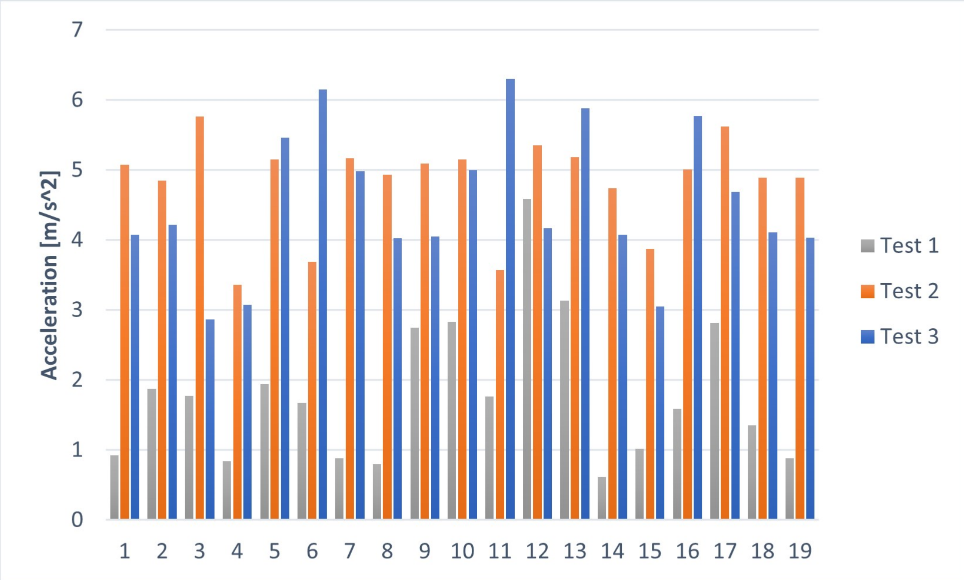
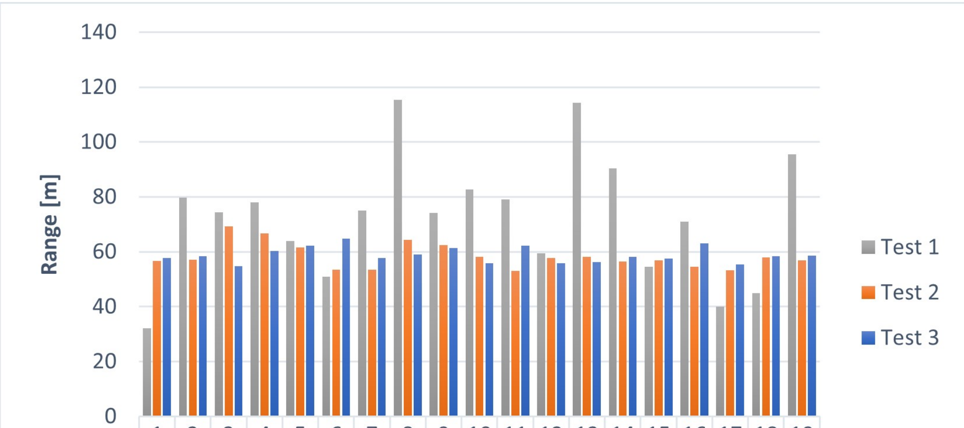


Figure 6.5: Maximum longitudinal deceleration for the different tests.

The average range is computed from the moments when the ego vehicle approaches the lead car. Therefore, the values in Figure 6.6 represent, more or less, how far the ego vehicle stays behind the lead one. As expected, the ACC systems show similar results, because the reference distance is defined identically in the two models. On the other hand, the results from Test 1 show how different drivers consider adequate the relative distance.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Test 1	32	80	74	78	64	51	75	11	74	83	79	59	11	90	54	71	40	45	95
Test 2	57	57	69	67	62	53	54	64	62	58	53	58	58	56	57	54	53	58	57
Test 3	58	58	55	60	62	65	58	59	61	56	62	56	56	58	58	63	55	58	59

Figure 6.6: Average ego vehicle distance to the leading car for the different tests.

An interesting comparison is given by looking at the answers to questions 10 and 19, where two the testers are asked to rate the relative distance between the two vehicles, left by the two ACC models. It is interesting to see that driver 1 and 18 are the only ones that consider the lead vehicle too far, while using both ACC systems. Indeed, these two testers, on average, leave two of the lowest distances between them and the leading car when they are required to drive the vehicle on their own. Also looking at driver 14 it is possible to find a confirmation of the results. In fact this driver finds the lead car too close, and leaves 90 meters while freely driving.

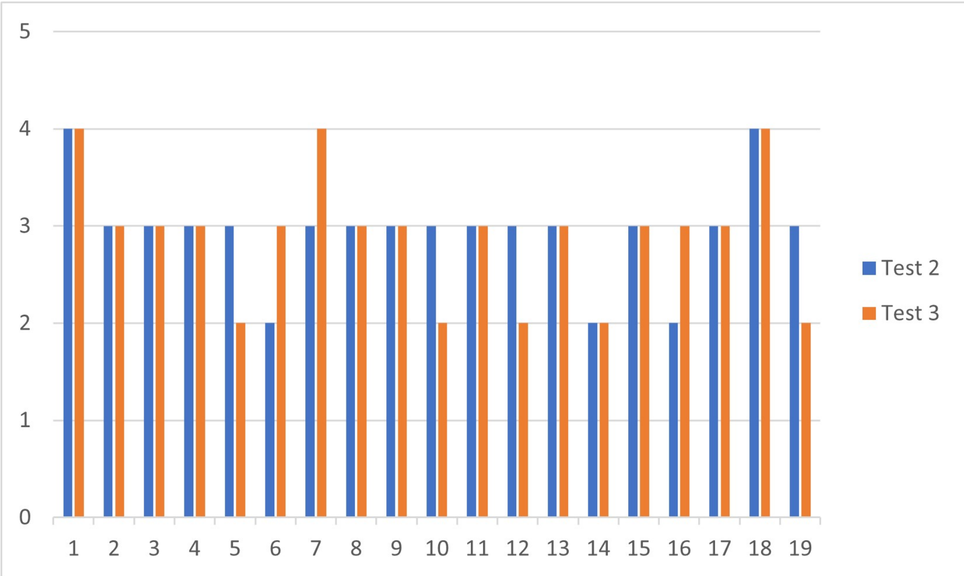


Figure 6.7: Answers regarding the relative distance between ego and lead vehicle.

Lastly, the speed data are presented in Table 6.8. The average and the standard deviation are computed again when the ego vehicle is approaching the leading one. The average speed is similar in all three tests since the vehicles are forced to travel at a certain speed to stay behind the lead car, independently from the relative distance left. The difference lays in the standard deviation. This shows that when the driver has to stay behind the lead vehicle, tends to accelerate and brake more often, resulting in speed oscillations. While the standard deviations form tests 2 and 3 show that the ACC models brake and then follow the lead car at the same speed, without oscillations.

	Test 1	**Test 2**	**Test 3**
Average speed [km/h]	112.36	111.69	111.33
Standard deviation [km/h]	10.33	8.46	8.78

Table 6.8: Average speed and standard deviation for the different tests.

7 | Conclusions and future developments

Cruise Control (CC) and Adaptive Cruise Control (ACC) have been designed by several car makers to regulate the speed of the car under different traffic situations. However, these systems still present some issues, regarding safety, comfort and stability. This thesis presents a control scheme allowing to have smooth and stable transitions among the different control modes these systems have. A system with three different modes is presented:

- *Cruise*: a speed controller to maintain the desired speed by the driver.
- *Follow*: a relative speed controller to follow the lead vehicle, if present in the same lane as the ego vehicle.
- *Safety-critical*: a distance controller to brake in case a vehicle, or any other obstacle, is detected in the same lane as the ego vehicle, and to keep a safety distance as well.

These modes are defined by three controllers, that take into account all possible situations. A feedback controller featuring a fuzzy gain scheduling is designed to work together with a feedforward contribution whose aim is to fast up the system response. An Emergency Braking (EB) system has been added to the controller, with its activation that is triggered only in safety critical situations. An extension to the conventional ACC is presented as well, being the CACC system a promising research for future driving development.

New developments in commercial simulation softwares enable to obtain results that are as accurate and reliable as ever. Therefore, the whole ACC logic is built to co-simulate with VI-CarRealTime and VI-WorldSim, the software from VI-Grade to define the vehicle model and simulation scenario respectively. From offline simulations the mode switches result stable and smooth, and the system reacts properly in all the tested scenarios. To do so, the ACC system has been tested in a series of maneuvers to verify its functioning. At first, some simple maneuvers where the ego vehicle encounters another vehicle ahead, in its same lane, have been performed. Then, situations where some vehicles move and change lane ahead of the ego have been tested as well. These maneuvers are meant to verify the reaction of the ACC system with lead vehicles, thus the functioning of the logic and sensor included in the ACC. The system is then compared with a state of the art model. The presented model shows a more comfortable behaviour, with softer braking and less oscillations.

To prove the comfort of the designed ACC system, a high fidelity driving simulator is used. Testers have provided their feedback about two ACC systems tuning variants and also about the driving simulator itself. The results show that the systems satisfy safety and comfort requirements of a state of the art Adaptive Cruise Control system.

7.1. Future works

Possible further avenues for research in this area include:

- Implementation and validation of the CACC system on the driving simulator. Implementing V2V communication, using throttle and brake signals as well.
- Determining an optimal value for vehicle acceleration with respect to various constraints, such as the relative distance between the host and lead vehicle, fuel economy, passenger comfort, traffic congestion, etc. This procedure can be done by modelling an optimization problem while defining the requirements as objectives.
- Extending the ACC model with systems such as Stop&Go, to account for urban environment usage.

Bibliography

[1] M. A. A. Abdulbaqi Jumaa B., Mousa Abdulhassan A. Advanced driver assistance system (adas): A review of systems and technologies. *International Journal of Advanced Research in Computer Engineering and Technology (IJARCET)*, pages 231–234, 2019.

[2] N. H. T. S. Administration. The national motor vehicle crash causation survey. *Computer Science*, 2008.

[3] K. V. C. Analysis of the effects of adaptive cruise control (acc) on driver behavior and awareness using a driving simulator. Master's thesis, University of Cape Town, Cape Town, 12 2017.

[4] P. N. P. G. Prabhakar, S. Selvaperumal. Fuzzy pd plus i control-based adaptive cruise control system in simulation and real-time environment. *IETE Journal of Research*, pages 1–11, 2018.

[5] H. H. Hallouzi R., Verdult V. Experimental evaluation of a co-operative driving setup based on inter-vehicle communication. *Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV 2004)*, pages 1–6, 2004.

[6] A. N. S. Institute. Sae levels of driving automation. *SAE International*, 2018.

[7] H. l. D. I. Insurance Institute for Highway Safety. Advanded driver assistance, 2022. URL `https://www.iihs.org/topics/advanced-driver-assistance`.

[8] C. C. C. Ioannou P. Intelligent cruise control. *IEEE Transactions on Vehicular Technology 42.4*, pages 657–672, 1993.

[9] C. G. J.de Winter. Pleasure in using adaptive cruise control: A questionnaire study in the netherlands. *Traffic Injury Prevention*, pages 216–224, 2017.

[10] S. Kim. *Design of the Adaptive Cruise Control Systems: An optimal Control Approach.* PhD thesis, University of California, Berkeley, 4 2012.

[11] M. T. Kim S. G. Mode switching and smooth motion generation for adaptive cruise

control systems by a virtual lead vehicle. *IFAC Symposium on Transportation Systems*, pages 490–496, 2009.

[12] C. Kreuzen. Cooperative adaptive cruise control using information from multiple predecessors in combination with mpc. Master's thesis, Delft University of Technology, Delft, 5 2012.

[13] E. H. Mamdani. Application of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Transaction on Computers*, pages 1182–1191, 1977.

[14] E. NCAP. Euro NCAP for safer cars, 2022. URL `https://www.euroncap.com/it`.

[15] Z. X. P. Ioannou. Throttle and brake control systems for automatic vehicle following. *IVHS Journal*, pages 345–377, 1994.

[16] S. B. Ploeg J. Design and experimental evaluation of cooperative adaptive cruise control. *International IEEE Conference on Intelligent Transportation Systems*, pages 260–265, 2011.

[17] G. H. S. F. Varotto. Empirical longitudinal driving behavior in authority transitions between adaptive cruise control and manual driving. *Journal of the Transportation Research Board*, pages 105–114, 2015.

[18] V. Sasan. Design and formal verification of an adaptive cruise control plus (acc+) system. Master's thesis, McMaster University, Hamilton, Ontario, Canada, 2014.

[19] N. Simonini. Validation of a coupled lane keeping and adaptive cruise control inside a high fidelity driving simulator scenario. Master's thesis, Politecnico di Torino, Torino, 7 2021.

[20] H. S.-M. T. Ching-Chih. Fuzzy longitudinal controller design and experimentation for adaptive cruise control and stop & go. *Intelligent & Robotic Systems*, pages 167–189, 2010.

[21] e. a. V. Milanés, D. F. Llorca. Intelligent automatic overtaking system using vision for vehicle detection. *Expert Systems with Applications 39*, pages 3362–3373, 2012.

[22] E. A. Vahidi A. Research advances in intelligent collision avoidance and adaptive cruise control. *IEEE Transactions on Intelligent Transportation System 4.3*, pages 143–153, 2003.

[23] VI-Grade. Cable-driven DiM DYNAMIC Simulator, 2021. URL `https://www.vi-grade.com/en/products/cable-driven_dim_dynamic_simulator/`.

[24] VI-Grade. VI-WolrdSim, 2021. URL `https://www.vi-grade.com/en/products/vi-worldsim/`.

[25] *VI-CarRealTime 2021.3 Documentation*. VI-Grade GmbH, Darmstadt, Germany, 11 2021.

[26] *VI-WorldSim 2021.3 Documentation*. VI-Grade GmbH, Darmstadt, Germany, 11 2021.

[27] L. Y. Wu C., Xu Z. Spacing policies for adaptive cruise control: A survey. *IEEE Access*, pages 50149–50162, 2020.

List of Figures

List of Tables

Acknowledgements

First of all, I would like to thank my supervisor Prof. Edoardo Sabbioni for his support, trust, and advice during my studies. A great thank also goes to Michele Asperti, that greatly helped me during the whole thesis, during the modelling and testing phases of the work.

Special thanks go as well to all the people that helped me during my academic journey. To my colleagues during my courses in Polytechnic of Milan and Oxford Brookes. To my teammates in Oxford Brookes Racing and Physis PEB, two of the greatest and most beautiful experiences in my life, that greatly helped me to grow as an engineer. My great appreciation goes to all my friends outside university, some new and some old.

Finally, my deepest thanks go to my family, that always supported and encouraged me. Without them I would not be here now.