POLITECNICO DI TORINO

MASTER's Degree in MECHATRONIC ENGINEERING

MASTER's Degree Thesis

Analysis and implementation of vehicle Platooning through real-case simulations in SUMO.

Supervisors

Prof. Lorenzo ZINO

Prof. Alessandro RIZZO

Company Supervisors

Antonio CAPUANO

Gianluca TOSCANO

Candidate

VALENTINA GULINO

A.A. 2022/2023

Abstract

Today the possibility of self-driving vehicles becoming a common method of transportation becomes more and more real. Autonomous driving represents a significant milestone in the evolution of transportation systems. Currently, the technology is progressing, but its implementation is more complicated due also to ethical and regulatory issues. However, the hope is to reach full vehicle automation in the near future because this technology could bring numerous benefits in terms of human safety and vehicle efficiency.

One potential direction for the advancement of autonomous driving is the concept of Platooning, a cooperative driving approach that involves multiple vehicles traveling in string formation and in close proximity to each other. Platooning is achievable due to the Advanced Driver Assistance Systems and the communication technologies that allow vehicles to cooperate and move as a coordinated group. This technique can help mitigate some problems such as road congestion,environmental pollution, road traffic efficiency, and could also play a key role in improving vehicle safety, efficiency and fuel consumption.

The contribution of this thesis is to analyze, implement and test the platooning driving method through simulations in a real-world framework. The simulations are performed using SUMO software and primarily focuses on study of the lateral dynamics control system for platoons. In the initial phase, the construction of the model for the individual vehicle to be simulated was undertaken. Subsequently, the development of the controller was conducted within Matlab Simulink. Then the Matlab/ Simulink model of the platoon was integrated in the SUMO framework. Finally to assess the model and controller's effectiveness, a simulation of a real-case driving scenario was conducted.

Acknowledgements

I would like to thank Teoresi company, in particular company tutors Antonio Capuano and Gianluca Toscano for their support during these months in the company. The practical experience enriched my studies and gave me a concrete and useful perspective on the professional world. It also gave me the opportunity to compare myself with other graduate students, with whom I established a good relationship of collaboration.
In addition , I would like to express my gratitude to Professor Lorenzo Zino for the assistance and the availability that he has shown over the last few months.

Table of Contents

List of Tables

List of Figures

Acronyms

AV

Autonomous vehicle

ADAS

Advanced Driver Assistance Systems

V2V

Vehicle to Vehicle

V2X

Vehicle to everything

MPC

Model Predictive Controller

LV

Leader vehicle

LV

Follower vehicle

Chapter 1

Thesis goal

This thesis was carried out in collaboration with the company Teoresi s.p.a. The basic idea behind this thesis was to extend a thesis project already carried out in the company[1]. The previous thesis dealt with the development of a model predictive control system for the longitudinal dynamics of a platoon of 4 vehicles.
The starting idea was therefore to advance the thesis by integrating lateral dynamics control for the platoon.
A control over lateral dynamics is necessary to:

- Manage cooperative lane changes.
- Avoid obstacles on the road.
- Manage overtaking.
- Multi-vehicle merging.
- Path following.

Then was also integrated the idea to simulate the vehicles platoon In SUMO, an open source software that allow to simulate real case scenarios.

The final objective of the thesis is therefore to analyse and attempt to implement a system for controlling the lateral dynamics of a platoon of vehicles and, finally, to test the control logic employed in a real traffic scenario.

Chapter 2

Introduction to autonomous driving

Autonomous driving refer to a technology where the vehicle is capable of moving autonomously without the intervention of a human driver.
The self - driving vehicle is able to perceive the environment, make decisions, monitoring important systems, and control its own movements. This is achieved thanks to a combination of some technologies such as sensors, actuators, complex algorithms, artificial intelligence and a lot other systems.
Nowadays autonomous cars have achieved high level in terms of reliability and safety, in the past few years research about self-driving cars has intensified, both in academic and industrial field.
The biggest car manufacturers are investing large amounts of resources on ambitious projects, with the aim of changing the concept of mobility. [2]

2.1 Historical context and milestones.

Self-driving cars have evolved over the years alongside the development of various technologies such as embedded systems, sensors,cameras, artificial intelligence and many others[3].
The history of autonomous driving cars is an intriguing journey that blends technology, innovation, and legal and ethical challenges. Usually, autonomous driving cars are considered a modern technology, but in reality, their history dates back to the early 90s, a period when automobiles were becoming more widespread. As the number of cars on the roads grew, so did the incidence of accidents, largely attributed to human errors. [4] This gave rise to the concept of developing a mechanism to oversee and control vehicles.

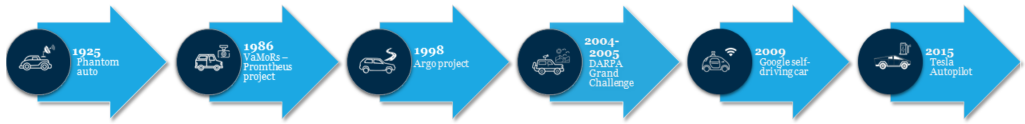

Figure 2.1: Autonomous driving milestones

Phantom Auto - 1925. The first driver-less car was demonstrate in the 1925, the so called "*Phantom Auto*" [2] which was radio controlled through an antenna equipped on the car which captured the signals sent by an operator in another car.
Then during the years a lot of improvements has been made, in 1953 the General Motors in collaboration with RCA Labs develop automated driving control systems where the car was wired guided.

VaMoRs - 1986. We had to wait until the 1980s when, with the advancement of other technologies like microprocessors, computers, and computer vision systems, the wired-control technology was abandoned, and the concept of having onboard processors, cameras, and LiDAR began to develop. VaMoRs was a Mercedes-Benz van that integrated for the first time this type of technology, it was able to drive independently only with the help of the computer-vision system.[5]

Prometheus Project - 1987-1994. During the same years, given the results obtained, the European Union promoted this major project for research in the field of self-driving cars using vision based control. [4]

ARGO Project - 1998. Important Italian project to mention developed during the late 90s, at the Department of Information Engineering of the University of Parma[6]. Argo was a Lancia Thema car modified with vision-based system able to read information from the external environment. The capability of the car to detect obstacle and Lane was tested in the MilleMiglia tour.

Figure 2.2: Argo autonomous car [5]

DARPA Grand Challenge - 2004. Is a prize competition organized by the Defense Advanced Research Projects Agency, with the aim of promoting and accelerating the development of autonomous driving technologies. Participants are required to design autonomous vehicles capable of handling harsh driving conditions. Two events were conducted, one in 2004 where no participant successfully completed the challenge, and another in 2005, which was won by Stanford University. [7]

Figure 2.3: Stanley - Stanford University's robot during the DARPA challenge [8]

Google Self-driving car - 2009. Google's self-driving car project, now known as Waymo. In 2009 Google starts the development of its first car, a Toyota Prius equipped with cameras, lidar , and complex software algorithms. The aim of the project was to develop an autonomous car to improve safety. The car was testing in a lot of scenarios including cities and was able to drive autonomously more miles than others car [9].

Figure 2.4: Google self-driving car [9]

Tesla Autopilot - 2015. Tesla Motors announced its first version of Autopilot. This system integrated in Model S cars allow to autonomously steering, braking, and adjust the speed limit thanks to the images detected by the vision system.[10] Furthermore, it offers self-parking and obstacle avoidance capabilities.

The mentioned examples are just some of the key events in the evolution and development of autonomous vehicles. Nowadays, nearly all major automotive companies are dedicating substantial resources to the advancement of this technology.

2.2 Autonomous Driving Impact on Urban Mobility

Self-driving cars could have a substantially impact in some aspects of human life, transportation system, economy, environment and many more. Has the potential to create a new and improved mobility system. Some of the benefits that can derive from the use of self-driving cars are:

Safety- Autonomous vehicles have the potential to significantly reduce accidents caused by human error, indeed according to Istat in 2022 in Italy almost 38% of car crashes occurs due to human distraction, high speed and the right of way failure. other causes of accidents can be drunk driving, dangerous manoeuvres, failure to respect the safety distance or fatigue.
In all the cases mentioned previously, autonomous vehicles can be a valid solution to reduce the risk of accidents, given that they are able to calculate the actions to be taken to avoid a collision in a few seconds.

Traffic Efficiency- Self-driving cars can communicate with each other and with traffic infrastructure, optimizing traffic flow and potentially reducing congestion.If applied to the supply chain sector this translates into optimized transportation routes, more efficient and low-cost deliveries.

Environmental Impact-These vehicles also employ an efficient fuel-saving mode, achieved by combining traffic simulation, machine learning, and optimization methods, which are used to allocate traffic and manage the movement of autonomous in best possible path. [2]. But also another

Accessibility- AV Could provide mobility to individuals who cannot drive due to age, disability, or people without licence, giving them more independence. In all these cases autonomous vehicles can be a potential transportation solution.

Shared mobility- Nowadays car sharing and car-pooling are very spread and used means of transport and could be more efficient with autonomous vehicles because it would be possible to organize, schedule and program synchronized trips. This could generates a new mobility business models. Another important consequence of this aspect is that could decrease the necessity of

own a vehicle, thus decreasing all the cost relative to maintenance but also could reduce the space needed for parking, especially in urban areas.
Finally, it is necessary to mention one aspect, although not of primary importance, but which could change our way of conceiving a trip. In fact people inside can relax, read, work and do whatever they want while the car accompanies them to the point where they want to go without worry. Obviously until now this is just fantasy but this could be the evolution and the direction of vehicle in the next future.

2.3 Advanced Driving Assistance Systems

2.3.1 Categorization of autonomous driving levels.

SAE (Society of Automotive Engineers) International established a set of standards and definitions for characterizing autonomous driving based on the level of automation present in the vehicle. The classification system includes six levels that help classify and standardize capabilities but also responsibilities in autonomous driving systems.
These levels are often referred to as the "SAE J3016 levels" or simply "SAE levels of automation".

The six levels are[11]:

- **Level 0**: No driving automation.
  The driver has the full control of the vehicle, is responsible for all aspects of driving, with no automation support. If there is any driving automation system it does not acts on the Dynamic driving task but is just needed for warning purpose.

- **Level 1**: Driver Assistance.
  The vehicle can assist with either longitudinal or lateral motion control subtask, but not both simultaneously. The driver must be present all the time, must monitor the environment and remain engaged at all times and has to intervene if necessary. The driver has also the role of deciding when and where to activate or deactivate the driving automation system.

- **Level 2**: Partial Driving Automation.
  In this case the vehicle can performs both the longitudinal and lateral motion control task simultaneously. The driver must be present all the time and has the same responsibilities of the previous level and has the role to immediately take control of the entire autonomous driving system if necessary.

- **Level 3**: Conditional Driving Automation.
  The vehicle can manage almost all the driving task under specific conditions, but the driver may need to take control when requested by the system or if any failures happens.Also the driver has to verify that all the functions

of the automated driving system are functioning and ready to use. The system is also able to detect when some performance relevant issues happens.

- **Level 4**: High Driving Automation.
  In this case the vehicle performs all the driving task without the need of human intervention. Thus the driver becomes a passenger when the systems are active. Nevertheless the driver has the ability to deactivate the guidance systems whenever desired. The system is able to only to performs all the task but is also able to automatically manage the transition conditions to a situation of minimum risk, when for example some emergency occurs.In all the previous level this was managed by human driver. Even if the driver has now switched to the passenger function, still is requested to intervene when the system reaches the limit to operate and has the role of verifying that all the systems work correctly and that exist the conditions to engage the automated system.But the

- **Level 5**: Full Driving Automation.
  In the last level basically the system manged everything and has unlimited ability to operate in every domain of the vehicle and even in the emergency situations. Still the driver must pay attention to the surrounding environment, understand when and if conditions to travel safely are met and check that the automated driving system works correctly.

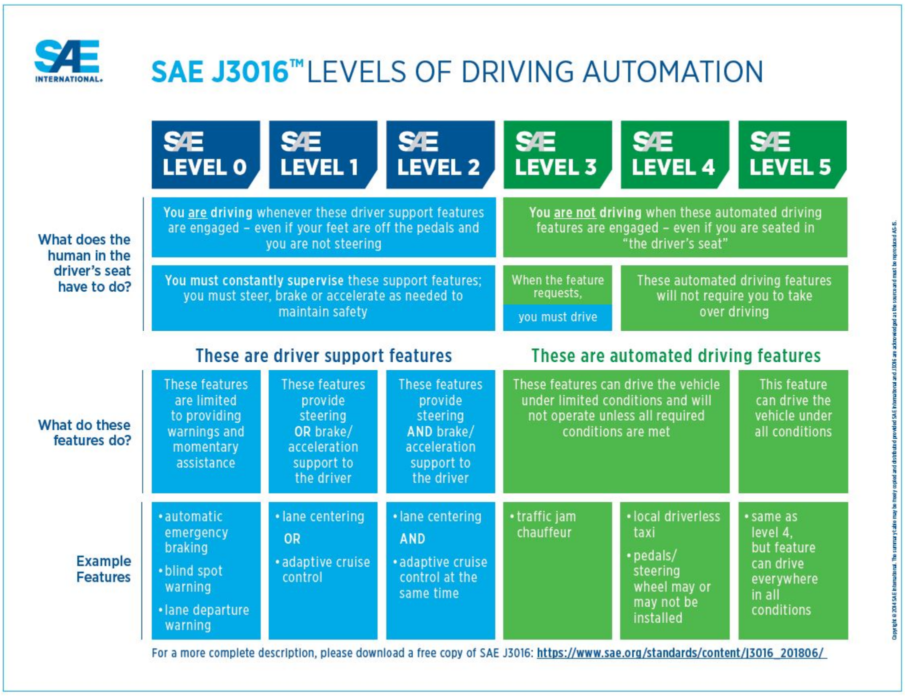

Figure 2.5: SAE J3016 Visual Chart [12]

2.3.2 ADAS and Sensor technology

Below are analyzed some of the most common ADAS, some of which have even become mandatory to have in a modern vehicle, because as this study [13] demonstrate, Driving assistance systems are able to significantly mitigate both fatal and non-fatal accidents. Different systems were analyzed in the study, all of have been shown to be able to reduce, even if in different percentages, the risk of crashes. In particular, the system that proved to be the most efficient from this point of view is AEB, managing to prevent the risk of front-to-rear crashes by a percentage between 40% and 50%.
One of the most known ADAS is the **Adaptive Cruise Control (ACC)**, this system that acts on the longitudinal dynamics of the vehicle as it allows the car to maintain a predefined speed and distance from the previous vehicle. The system controls both the speed and the distance,braking or accelerating autonomously based on the vehicle and thus maintaining a safe following distance. This system is an evolution of the simpler cruise control (CC) which allows the control of only one parameter, the velocity. [14]
Forward Collision avoidance system (FCAS) use a set of sensors and cameras to monitor the road ahead the vehicle to and through algorithms the system evaluates the surrounding environment to detect potential accidents. The system warns the driver via signals or, if necessary, acts directly by activating emergency braking or performing evasive maneuvers. Some systems are also able to detect vehicles approaching from the side, for example when changing lanes.

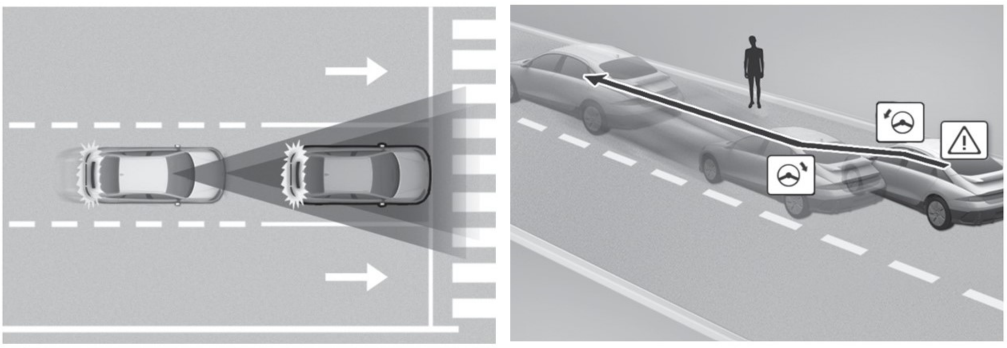

Figure 2.6: Forward collision avoidance, basic function and evasive maneuver [15]

Autonomous Emergency Braking (AEB): this system is often used together with the one described above, thanks to the use of cameras and sensors or a combination of them, the system manages to identify a potential danger for both cars and pedestrians and acts only if the driver does not act in time.
Lane Departure Warning (LDW) alerts the driver, using vibration or audio signals, if the vehicle drifts out of its lane while and **Lane Keeping Assist (LKA)** provides steering assistance to keep the vehicle within the lane.

Both systems uses cameras that detect both road marking lanes and calculate accordingly the steering angle to be applied to correct the trajectory.

For each of the technology described before uses a combination of different sensors to enable specific functionalities. The choice of sensor depends on the requirements of the ADAS feature and the level of accuracy and reliability needed. This paragraph provides an overview of the most used sensors in driving assistance systems described above:

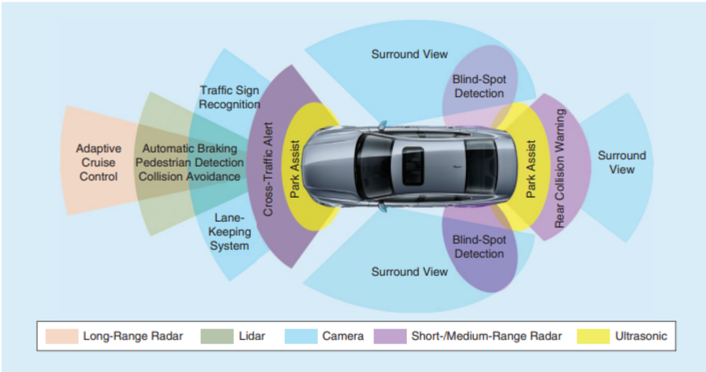

Figure 2.7: Sensors used in ADAS [16]

Exteroceptive sensors

Exteroceptive sensors are sensors that provide information about the external environment or surroundings of the vehicle. Some examples are:
LiDAR (Light Detection and Ranging) is a remote sensing technology used for distance measurement. This technology uses sensors that emit laser rays and then measure the time necessary for the light pulse to be reflected. In this way the Lidar is able to create 3D representations of the surrounding environment, managing to create very detailed maps, being very detailed and accurate sensors. These sensors are capable of measuring up to 250 meters and therefore can be classified as long-range sensors.
Lidar is generally used for implementing anti collision systems, object detection and many more systems.
Radar which stands for Radio Detection and Ranging, is a technology that uses radio waves to detect the presence, distance, speed, and direction of objects by measuring the frequency of the reflected waves. In terms of operation it is very similar to Lidar but in terms of reliability and costs it is significantly superior, as it is not affected by weather conditions and costs less. Also it exist different type of radar, that allow short/medium range detection and other type that performs long/range detection.
Camera are the most use sensors in autonomous vehicles, obliviously there a lot of types of camera but all has the potentiality to see images and specifically to see colors and texture. Cameras are capable of detecting objects thus can

be used for recognition of road signs, traffic lights, pedestrians, objects and much more. They are very versatile systems that can also be used as a security camera for vehicle monitoring [17]. Cameras are a good alternative because their cost is relatively low and offer good performance, however a negative feature is that they require a lot of computational power. In fact the data collected by the cameras must be processed. To deal with this negative side, in recent years the idea of sensor fusion has taken hold, the data received from multiple sensors are mixed. Through this technique it is possible to obtain the best performances from driving assistance systems.

Proprioceptive sensors

Are a class of sensors that provide information about the internal state and conditions of a system or vehicle. These sensors are essential for monitoring and controlling the system's own physical properties, such as position, orientation, movement, and internal conditions.
GPS (Global Positioning System)
IMU, Inertial Measurement Unit, is an electronic device that measures forces, accelerations and angular rate of an object in three-dimensional space. Typically consist of accelerometers and gyroscopes and is a very common device in the field of control of autonomous vehicles because provides real-time measurements of motion, acceleration, and orientation.
Encoders are essential device in the control of an autonomous vehicles, they are able to provide information on position, direction and velocity of an element. For example encoders are used on the steering system, providing information about the steering angle. Thus these devices are widely used because they are inexpensive and are easy to implement.

2.3.3 Vehicular Communication V2X

Is an important technology introduced in the field of autonomous driving, the acronym V2X stands for Vehicle-to-Everything, which account for all the types of communication systems through which the vehicle exchanges information with the surrounding environment. More in the specific in an autonomous vehicle can be implemented Vehicle-to-Vehicle communication (V2V) and also Vehicle-to-Infrastructure communication (V2I).
V2V is a inter vehicle communication [18] and allow to exchange information between near vehicles. This information can include data such as position, speed, acceleration,steering wheel angle, vehicle trajectory and other relevant details. This technologies utilizes a wireless protocol known as Dedicated Short Range Communication.
V2I involves vehicles communicating with infrastructure along the road, such as traffic lights, road signs, and sensors. This permit to have real-time traffic information, data, and other important information that can be uses to regulate the traffic flow[19].

Is important to notice that this technology combined with the Driving assistance system and in particular with the ACC technology enable the development of the so called Cooperative Adaptive Cruise control (CACC). the vehicles perform the same actions described before but in this case they receive more information thanks to vehicular communication, in this way it is possible to implement a much more effective and complex control system.

2.4 Challenges in Autonomous Driving

Autonomous driving faces numerous challenges, ranging from technical and regulatory issues to societal and ethical considerations.

In order to have full autonomous vehicles circulating on the road, some issues need to be resolved. [3] From a technical point of view, there are many problems regarding different aspects of the development of autonomous driving. One and maybe the most important aspect is the reliability, the autonomous vehicle must be working in any weather conditions and in every urban environments therefore all the systems that make up the vehicle must function perfectly in all conditions. All of this systems uses a lot of sensors thus there is a huge amount of data to be processed in real time, this is a very complex situation to manage and requires powerful computing capabilities.
Another important aspect is Safety,in order to have a reliable vehicle it is necessary to establish safety standards that provides a comprehensive rule to be applied also in terms of testing and validation.
As regard ethical and social challenges, one aspect to take into consideration is the fact that human empathy must be replicated in some way, and from a decision-making point of view in this case the programmers will have to decide what the vehicle do in the presence of complex situations, such as a unavoidable accident in which the vehicle must decide whether to save the life of the driver or the pedestrian. Another aspect is that the widespread adoption of autonomous vehicles could lead to job displacement in industries that rely on human drivers, such as trucking and delivery services.
Another scenario to take into consideration when talking about issues related to autonomous driving is cybersecurity. In fact, autonomous vehicles rely on connectivity and accumulate a huge amount of data thus they are potential targets for cyberattacks. Ensuring the cybersecurity of these systems is a crucial feature.

Here some of the issues related to autonomous driving have been mentioned but there are many others. Considering this, it is of fundamental importance for the complete development of autonomous driving to define international laws that ensure safety and reliability, that defines roles and responsibilities in such a way to create a comprehensive framework for the development, testing, and deployment of autonomous driving technology.

Chapter 3

State of Art of Platooning

This chapter aims to provide an in-depth analysis of platooning technology in road transport. This part explores the basic principles of platooning, examines the technological advances and assesses the challenges and societal implications associated with its introduction.

Platooning can be considered an evolution of autonomous driving and is implemented in the context of intelligent roads and the intelligent transportation system. Is an innovative driving method where vehicles essentially travel in formation one behind the other, like a convoy and in close proximity to each other, moving in a coordinated manner.
Platooning is developed through the fusion of two important advanced driving systems that we discussed in Chapter 1, Adaptive cruise control and Vehicular communication. These two technologies used simultaneously result in the Cooperative Adaptive cruise control (CACC)

The analysis of the main components of a platooning system can be divided into two macro areas [20]:

- **Control systems**
  Two types of control systems can be distinguished :

  - Longitudinal - is essentially concerned with controlling the speed of the vehicle to maintain a reference distance from the preceding vehicle. This distance must ensure safety.
  - Lateral - motion control during manoeuvring processes. The purpose of such a control system is to follow a desired path while minimising errors as much as possible.

  In general, the aim in both control systems is to ensure both individual vehicle stability and the stability of the entire platoon.

- **Coordination** This category includes all those features that enable the management and coordination of the entire platoon such as: Platoon formation strategy, route and destination planning strategy to optimise arrival time, fuel optimisation planning strategy . Methods of communication between vehicles and between vehicles and their environment.

Ultimately, all the above aspects must be taken into account in order to have a safe and effective design and management of the platooning system

3.1 Evolution of Platooning

The roots of platoon control can be traced back to the late 1980s in California, at the University of Berkeley, marked by the initiation of the PATH - Program on advanced technology for the Highway - project. This pioneering initiative addressed several fundamental aspects, including the overall objective of platooning, the division of control tasks, the design of the control architecture, and the exploration of technologies encompassing sensing, actuation, and communication.[21] The main focus of the program was to develop a method for reducing congestion and to increase the safety in the highways.
In the project firstly was analyzed and studied separately the longitudinal and lateral motion of the vehicles then the two dynamics control system was integrated to achieve full automation and some public demonstrations had already taken place in 1997, such as the one described in [22].The demonstration took place in San Diego and consisted of a platoon of eight vehicles, which successfully performed manoeuvres such as: starting vehicles automatically from a standstill, holding a cruising speed, automatic steering and many others.

Subsequently, projects were also launched in Europe for the study and research of platooning systems.
Particular mention should be made of SARTRE, Safe Road Trains for the Environment. The project was launched by the European Union in 2009 and was scheduled to last three years.Furthermore, the project was realised in collaboration with 7 partners.
The main aim of the European project was to realise vehicle platooning that would lead to a reduction in fuel consumption and an improvement in safety. [23].

Japan also dealt with the concept of platooning with the 'Energy ITS' project. This project is different from the others because the main focus is the energy savings aspect and the global warming prevention through CO_2 reduction methods [24]. This particular project dealt with truck platooning; the system consisted of four trucks, one of which was a light truck. The developed system was finally tested on a motorway at 80km/h, testing various driving states: from manual driving to full automation, which consists of ACC control for longitudinal dynamics and specific control for lateral dynamics.

Figure 3.1: Energy ITS project - test in expressway [24]

Thus, it can be deduced that there has been a global expansion of platooning research and all these projects have evolved from the development of in-vehicle platooning technology to addressing specific technological challenges and the practical use of platooning technology.
The ENSAMBLE project[25], which started in 2018 and ends in 2022, can be mentioned in this regard. The project - ENabling SafE Multi-Brand pLatooning for Europe - was carried out in cooperation between several European truck brands. The main aim was to test the multi-brand truck platoon and thus succeed in defining common standards for communication technologies. Finally, a real road test was carried out in Spain with seven trucks of different brands. The simulation showed that there are still some critical technical issues to be resolved in the area of platooning and that the technologies can only be matured with more tests in real scenarios and more mature implementations of the current technologies.

3.2 Advantages

The projects described above are just some of the various research projects being carried out in the field of platooning. It is clear that this technology is attracting worldwide interest. This is because the improvements it can offer are very advantageous.

Platooning can improve road safety, often the main cause of accidents is the delay in reacting to an unforeseen event or obstacle. The vehicle-to-vehicle communication system offers a solution to this problem, ensuring synchronised and quick that can prevent accidents and ensure the safety of the entire convoy. In the path project, one of the main reasons why platooning was analysed was the possibility of being able to reduce road congestion. In fact, platooning has proven to be a good technique to achieve this, in some projects it has actually been possible to double the road capacity [20].

Another reason was to try out a technique that would allow the reduction of fuel consumption. Platooning allows vehicles to stay very close to each other and thus reduce aerodynamic drag by up to 50%. [26] The impact of the introduction of platooning, thus leads to a reduction in aerodynamic drag, thereby minimising total drag and reducing engine power requirements. This reduction in power demand translates into lower fuel consumption.

Another factor you take into account when talking about platooning or, more generally, self-driving vehicles, is the convenience factor. In fact, by not having to be at the wheel, the driver will benefit from greater comfort during the journey.

Overall, platooning represents a potential future development in transportation that could have significant implications for efficiency, safety, and sustainability, although widespread adoption may take time and depend on various technological, regulatory, and societal factors. The successfully implementation of platooning on road requests a lot of trial and test on the real environment, to asses the feasibility and benefits of this approach.

3.3 Platooning technologies

There are two fundamental technologies that enable platooning: Adaptive cruise control and vehicle communication. Hence, we will analyze both methods in depth.

ACC

Adaptive Cruise Control (ACC) [25]is an advanced vehicle speed control system. It improves on traditional cruise control by allowing the vehicle to maintain a set speed, but also adds the ability to dynamically adapt to the speed of the surrounding traffic.
This system is designed to improve road safety and enhance the driving experience by reducing the need for driver intervention in speed control in variable traffic situations.
Various vehicle speed control strategies have been proposed over the years, based on different approaches to feedback control.
The main ones [1] are Constant Distance, in which a constant distance between to vehicle is set without considering the relative distance of the vehicles or the speed, Constant Time and Constant Safety Factor Criterion.
The constant-distance method is the least efficient because it requires a large relative distance between vehicles to allow for emergency braking.
Instead the constant time method is the most similar to human behaviour, the vehicle's distance in this case is related to their speed. This is a more suitable approach to realize adaptive cruise control.

V2V

This involves direct communication between vehicles in a platoon, allowing them to exchange information such as speed, acceleration, and braking status. V2V communication is essential for maintaining synchronized movement within the platoon.

The communication system is implemented through Vehicular Ad hoc Networks (VANETs), which, via wireless communication, enables the vehicle to communicate with other vehicles and the surrounding environment. Different topologies of communication can be realized, three common types are:

- Predecessor following (PF) :the vehicle receives information only from the preceding vehicle.

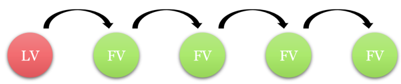

Figure 3.2: PF topology

- Leader following (LF): vehicles communicate only with the leader.

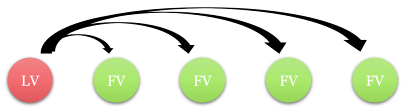

Figure 3.3: LF topology

- predecessor-leader following (PLF): the vehicles can communicate with the leader and with the preceding vehicle.

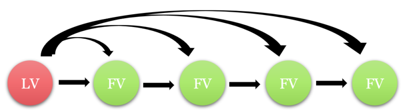

Figure 3.4: PLF topology

There are obviously many other topologies, furthermore communications can be both bidirectional and unilateral.
The communication topology is very important as it influences the success of platooning operations, for example in this paper [27] an MPC controller is analyzed for lateral dynamics under different types of communication. The results show that the best performances are obtained with the PLF topology, the LF topology shows slightly worse results than the PLF and finally the PF is the one that obtains worse results.

Chapter 4

SUMO - Simulation of urban mobility

SUMO stand for **S**imulation of **U**rban **MO**bility [28],is an open source, microscopic, multi-modal traffic simulator.
The software was developed by the Institute of Transportation Systems at the German Aerospace Center in 2001 [29] and allow to model various aspects of urban mobility.
Is a microscopic simulator in the sense that each vehicle in the simulation can be represented in detail, can be defined the physical characteristics such as mass, dimensions and many other parameters but also how the vehicle interacts with the simulation environment. Furthermore, a route can be defined for each vehicle and therefore the precise movements of the vehicle can be simulated.
Actually SUMO is not just a software but rather is a suite of applications that allow to create the traffic simulation.
Some of the key features and capabilities of SUMO are listed below [30]:

- Allow to model complex traffic simulation thanks to the wide range of vehicles that can simulate, including cars, buses, trams, bicycles, and pedestrians, each with its own unique behavior and characteristics.
  This allows to do a very detailed analysis of the traffic.

- In SUMO users can create complex road networks with various road types, intersections, and traffic control elements. It can be designed manually by using netedit but also a real area of a city can be imported using Open Street Map (OSM) and then converted into the netedit elements through netconverter. In this way a real world scenario can be created.

- SUMO can contribute to study pollution reduction since it incorporates emissions models to estimate the environmental impact of transportation systems, helping evaluate sustainability and air quality or helping eco-route planning.

- Simulation of the vehicular communications, indeed SUMO can be coupled with communication network simulator.
- Allows to analyze the impact of autonomous vehicles on urban mobility but also to analyze the capabilities of the new technologies[31]. Microscopic simulations are essential to understand and test the behaviour of the vehicles, because allow to the test the vehicles in a real world situation which introduces random situation that are uncontrollable just as it happens in real traffic conditions.

SUMO is thus a powerful tool for modeling and understanding complex traffic systems because has the ability of replicate real-world traffic scenarios with a variety of possible situations and to to provide a safe and cost-effective platform for studying and improving transportation systems, autonomous vehicle, road congestion, traffic safety and environmental issues.

4.1 How to setup a SUMO simulation

As the simulation application "sumo", which is included in the suite, uses own formats for road networks and traffic demand, both have to be imported or generated from existing sources of different kind

In order to make a simulation 3 principal elements are needed: [28]

- file*.net.xml*: in this element are defined the road network informations.
- file*.rou.xml*: here are described the Demand data, like routes and vehicles.
- file*.sumo.cfg*: this file allows the simulation to run on sumo-gui. Here the information about the simulation, such as time, steps and duration can be specified.

To model additional information to the simulation, like traffic lights, bus stops and many other, can be created a file *.add.xml*.
The files described above can be created using graphic software (netedit) or by creating *.xml* files and manually writing all the elements that are part of the simulation.
Using netedit is much more intuitive because you can immediately see what you are inserting into the simulation. On the other hand, creating an xml file is faster because you just need to write the commands as indicated in the documentation.

Example of simulation

Let's start with the network modelling, to do so the Netedit software was used. In Network mode can be created the road that are defined in terms of junctions and edges:

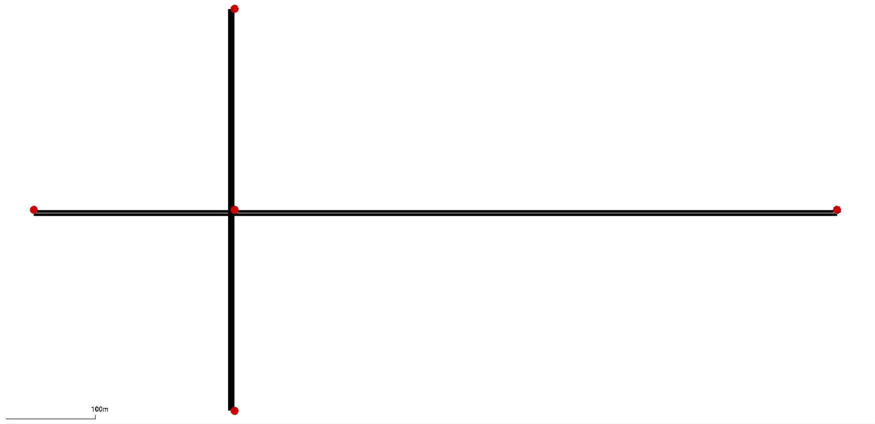

Figure 4.1: Network

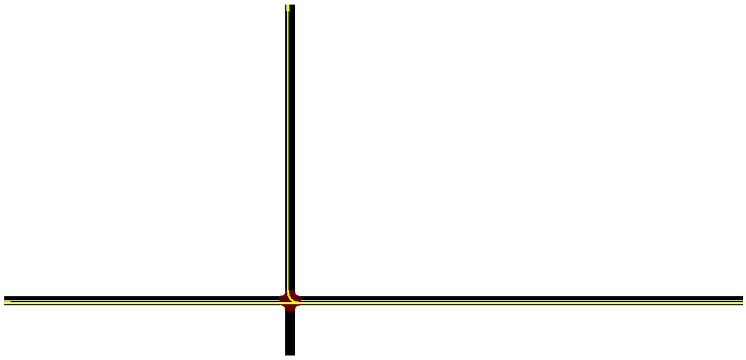

Figure 4.2: Routes definition

Then in Demand mode can be created the routes:

Still in Demand mode, the vehicle can be modeled. In figure 4.3 are described all the characteristic and parameters that are available in software.

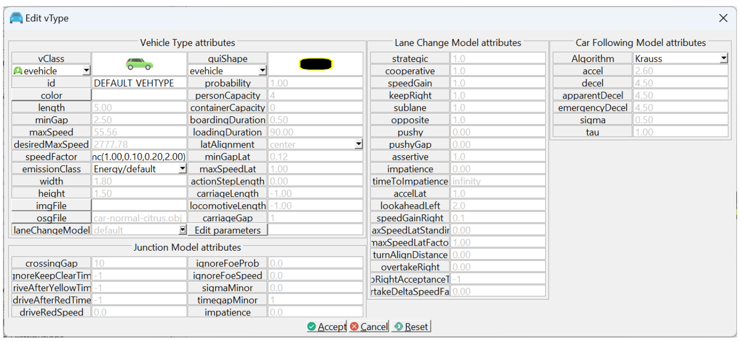

Figure 4.3: Vehicles parameters

When everything is configured, the file *.sumo.cfg* is generated and the simulation can be viewed in the main SUMO application.

4.2 SUMO - Matlab/Simulink integration

4.2.1 SUMO - Matlab

Sumo has an important feature which is that it can be interfaced with many other applications which allow its extension.
n this thesis the aim is to interface Sumo with Matlab/Simulink in order to create the mathematical model of the platoon in Matlab so to have greater accuracy in the description of the model and then carry out the simulations in SUMO thus exploiting its potential in simulating real world scenarios.
The integration of Matlab and SUMO is possible thanks to the **TraCI4Matlab** [32] which is an realize the TraCI (Traffic Control Interface) protocol in which SUMO acts like a server and Matlab acts like a client. Thank to this implementation the user is able to modify and control the object present in the simulation.
In the official manual [32] is explained how to setup a connection between SUMO and Matlab. Essentially 3 main actions has to performed:

- Initializing the connection with SUMO (seen as serve) through the command:

```
    traci.start=('sumo-gui -c ./simulation.sumocfg --start')

```

- Run the simulation specified in the previous command through the following syntax:

```
  traci.simulation.step()

```

- Closing the connection with the server through the command:

```
  traci.close()

```

In the main portion of the code, can an access and modify SUMO objects using the general structure:

```
traci.<domain>.<function>
```

In the manual are listed 15 different domains of functions that can be use in Matlab (real, edge, gui,inductionloop, junction, lane, multientryexit, person, poi, polygon, route, simulation, trafficlights, vehicle and vehicletype.)

4.2.2 SUMO - Simulink

To set up a connection between SUMO and Simulink a different different procedure is needed because Traci4MAtlab functions are only recognized in Matlab but not in Simulink, SUMO do not have a ready-to-use interface for co-simulation with vehicle dynamics and Simulink.
Thus to solve the problem set of embedded Matlab function must be created, in this function is implemented the code to set up the connection. But another problem is that Simulink has know that tracifunctions are only used only to make the simulation and not for code generation purpose.
To do this it can be used the function *coder.extrinsic('function')*.
The statement *coder.extrinsic ('function')* [33] allow to declare a function as extrinsic and execute it in Matlab.
The code generator does not produce code for the body of the extrinsic function and instead uses the MATLAB engine to execute the call.

Considering all this aspects, in order to make the connection with the SUMO server the following function have been created:

```
function start_sumo
coder.extrinsic("system")
system(['sumo-gui' ' -c ' \prova.sumocfg' ' --
remote-port 8813'  ' --start&']);
traci.init();
end
```

```
function close_sumo
traci.close();
end
```

Then in order to execute the simulation that we want we need to create another Matlab embedded function, which is the main part of the code:

```
function simulation_sumo
traci.simulationStep();
end

```

Considering this configuration , in Simulink the configuration framework to establish the connect is:

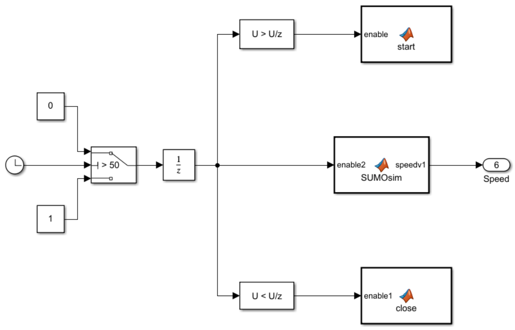

Figure 4.4: Sumo - Simulink integration

Time synchronization

Since we want real time simulations, the two programs must be synchronized in terms of running time.
Let's first analyze the time settings in SUMO:

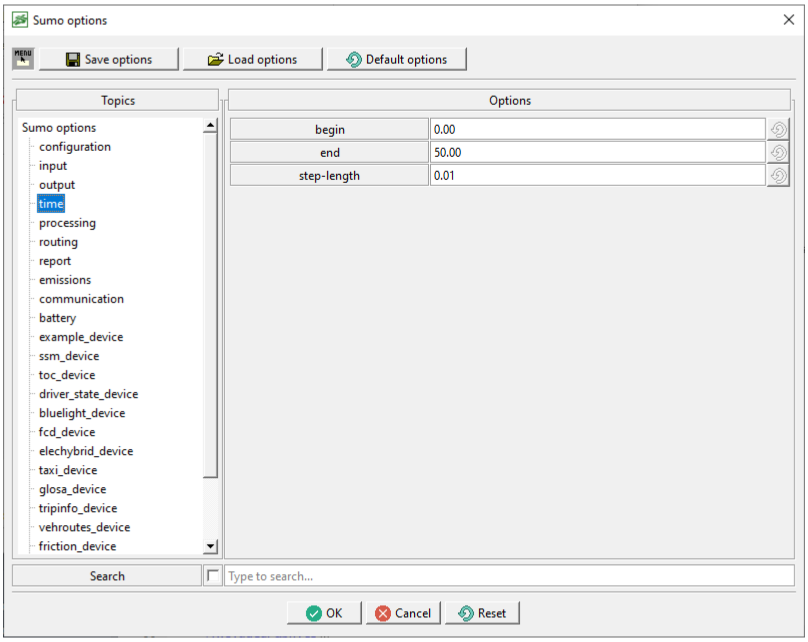

Figure 4.5: Time in SUMO

In the configuration file can be modified the parameters that accounts for the time settings, in general as can be seen in figure 4.5 it requires 3 parameters: Begin time, End time, Step Lenght.
As in the example imposing an End Time of 50s and a Step Lenght of 0.01 means that the simulation counts 5000 steps.

Then in Simulink, in order to match the step size of SUMO, the solver parameters must be modified:

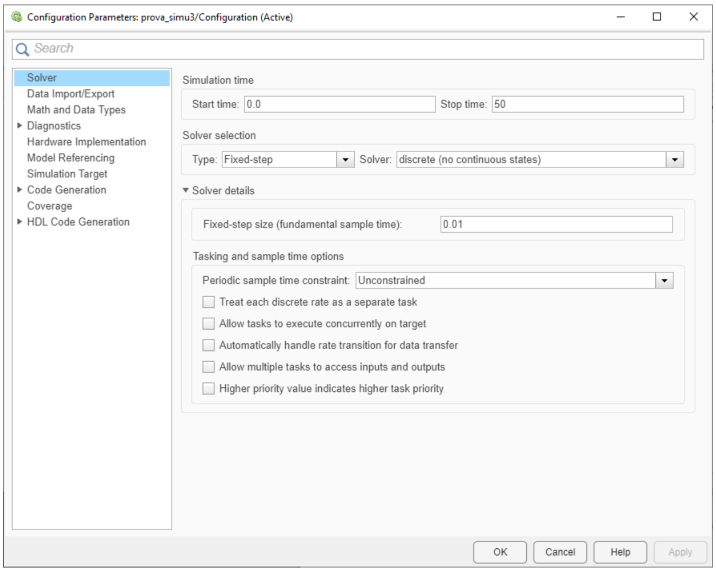

Figure 4.6: Solver configuration

Imposing as type a Discrete time solver, since SUMO runs a discrete simulation and as step length the same imposed in SUMO options.

Using this settings SUMO can be controlled through the Simulink model. The most important aspect is that each simulation step in Simulink corresponds to a step in SUMO. So the time in the 2 software run simultaneously, as we can see in the following picture:

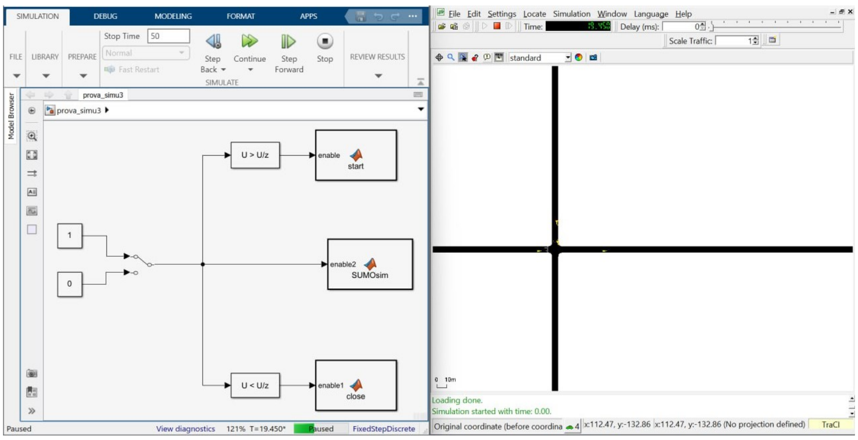

Figure 4.7: SUMO- Simulink time synchronization

SUMO commands in Simulink

Now that the connection is established, the next step is to understand how to implement the Traci commands in the Matlab embedded function block and how the commands sent through the model acts on the simulation.
Essentially in SUMO there are two main type of commands that allow to change the objects in SUMO:

- *.get* which permit to read information from the simulation.
- *.set* that allow to impose some specified parameter to the simulation objects.

To use this commands simply have to use the syntax described above in the main part of the code, specifying the domain and the type of function you want to use. As we can see from the figures

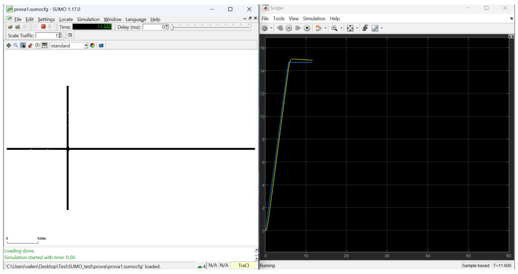

Figure 4.8: *.get* command

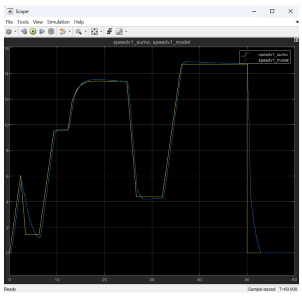

Figure 4.9: *.set* command

In Figure 4.8 the Simulink implementation model is able to read correctly the speed of the vehicle specified and in the Figure 4.9 SUMO receives correctly the commands and the model is able to follow the imposed speed.

The generated framework in Simulink is able to :

- Initialize correctly the connection with SUMO server.
- Run the selected simulation.
- Read correctly the speed in SUMO.
- Impose some values for the objects in SUMO
- Close correctly the connection with the server.

Chapter 5

Model of the vehicle

This chapter aims to described the vehicle model used for controller implementation. Indeed, as will be explained better later, to achieve the control of the lateral dynamics a Model Predictive Control is chosen. To be able to implement the controller in Simulink it is necessary to describe a predictive model, based on which the controller will calculate the optimal control input to realized the desired steering strategy.
In this first part of the chapter a general summary was given regarding the coordinate systems used to describe the motion of a vehicle.
Then in the second part of the chapter are found the equations that describe the vehicle dynamics.

5.1 Vehicle coordinates systems

To accurately define the equations of the vehicle's dynamic model, it is necessary to first establish the reference systems used in order to have a clear understanding of the situation. To do so I take as reference the chapter 8 of [34].
According to this, 4 reference frame can be identified:

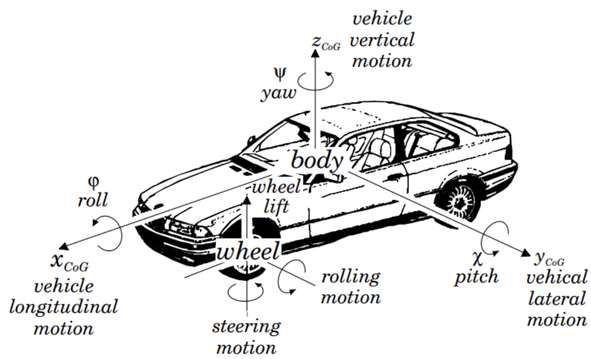

Figure 5.1: Coordinates systems [34]

- Fixed Inertial system
- Center of gravity coordinate system - CoG
- Undercarriage system
- Wheel coordinate system

The inertial system as name suggest is fixed in the space, while all the others frame move with the vehicle. The axis of the inertial frame are indicated with bold letter : X, Y, Z
The CoG frame is the most important, in our case, since represents the physical center of gravity of the vehicle and all the movements that it can have are referred in this frame.
Thus let's analyze the movements of the vehicle:

- Rotation around z_{CoG} is defined through the Yaw angle ψ.
- Rotation around y_{CoG} is defined through the Pitch angle φ.
- Rotation around x_{CoG} is defined through the Roll angle α.

5.2 Vehicle model

In general to model the lateral dynamic of a vehicle is used the 2 Degree of Freedom model also called Bicyle model. This model is used in the majority of the analyzed papers [35],[36],[37] [38].
Other uses a more complicated model, for example in [39] is used a 3-Degree-of-Freedom vehicle dynamics model and also they take into account a Four wheel independent driving vehicle,since there are more degree of freedom the vehicle manoeuvrability is improved.
In [40] the dynamical model of the vehicle is described using the bicycle model but they also made a mathematical description of the geometric interaction between two following vehicles. Putting together the two description they obtain the dynamical equations that describe the model of a vehicle which follow a target.
In this thesis, the two-degree-of-freedom (2 DOF) model was initially taken into account, then modifications were made to also take into account the dynamics between the vehicles of the platoon.
The 2 DOF vehicle model which is a simplified representation of a vehicle's dynamic behavior, focusing on two primary degrees of freedom : lateral motion y and yaw motion ψ.
Thus the basic notation are now defined, the next step is to define the vehicle parameters.
As main reference to build the model, in this thesis, is considered [41].

In the bicycle model the 2 front wheels are considered as a single wheel, the same for the 2 rear wheels. The model has 2 degree of freedom: the lateral position y and the yaw angle ψ.
To outline the vehicle in the bike model, some keys parameters must be considered:
Kinematic analysis

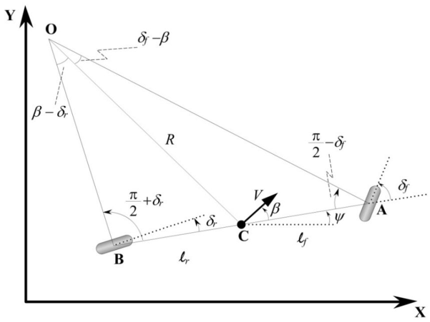

Figure 5.2: Kinematics lateral motion [41]

Table 5.1: Geometric model parameters

Parameters	
C	Center of gravity
l_f	Distance from the front axle to the CoG
l_r	Distance from the rear axle to the CoG
$L = l_f + l_r$	Wheelbase
δ_f	Steering angle relative to the front wheel
δ_r	Steering angle relative to the rear wheel
V	Velocity of the CoG
ψ	yaw angle, orientation of the vehicle wrt X global axis.
β	Body slip angle:angle between the direction of the speed longitudinal axis of the vehicle. V

Then in order to analyse the dynamics of the 2 DOF model, we need to consider :

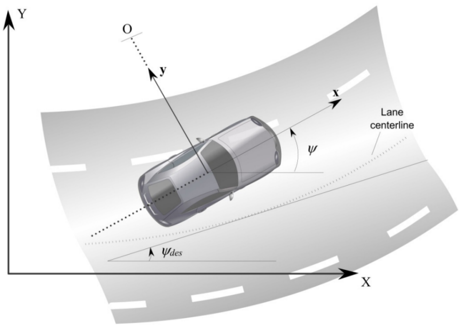

Figure 5.3: Dynamic model [41]

Applying Newton's second law for motion along the y axis :

$$ma_y = F_{yf} + F_{yr} \tag{5.1}$$

F_{yf}: lateral tyre force-front wheels
F_{yr}: lateral tyre force-rear wheel
a_y: Inertial acceleration of the vehicle.

$$a_y = \ddot{y} + V_x\dot{\psi} \tag{5.2}$$

$\ddot{y}$: lateral acceleration.
$V_x\dot{\psi}$: centripetal acceleration.
Replacing 5.2 in 5.1:

$$m(\ddot{y} + V_x\dot{\psi}) = F_{yf} + F_{yr} \tag{5.3}$$

Moment balance about z axis:

$$I_z\ddot{\psi} = l_f F_{yf} + l_r Fyr \tag{5.4}$$

m: mass of the vehicle.
I_z: yaw moment of inertia.

Computation of the lateral tyre forces:
The lateral tyre forces are proportional to the slip angle α,

- $\alpha_f = \delta_f - \theta_{vf}$: Front wheel slip angle

- $\alpha_r = -\theta_{vr}$: Rear wheel slip angle

θ_v angle between the wheel velocity and longitudinal axis of the vehicle.

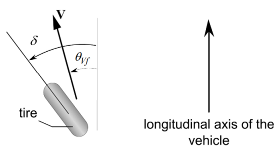

Figure 5.4: Wheel detail [41]

These angles are computed in the following way:

$$\theta_{vf} = \frac{\dot{y} + l_f\dot{\psi}}{V_x} \tag{5.5}$$

$$\theta vr = \frac{\dot{y} - l_r\dot{\psi}}{V_x} \tag{5.6}$$

Lateral tyre forces:

$$F_{yf} = 2C_f(\delta_f - \frac{\dot{y} + l_f\dot{\psi}}{V_x}) \tag{5.7}$$

$$F_{yr} = 2C_r(-\frac{\dot{y} - l_r\dot{\psi}}{V_x}) \tag{5.8}$$

where:
C_f : cornering stiffness front wheel.
C_r : cornering stiffness rear wheel.

Substituting all the values in 5.3 and 5.4:

$$m(\ddot{y} + V_x\dot{\psi}) = 2C_f(\delta_f - \frac{\dot{y} + l_f\dot{\psi}}{V_x}) + 2C_r(-\frac{\dot{y} - l_r\dot{\psi}}{V_x}) \tag{5.9}$$

$$I_z\ddot{\psi} = l_f 2C_f(\delta_f - \frac{\dot{y} + l_f\dot{\psi}}{V_x}) + l_r 2C_r(-\frac{\dot{y} - l_r\dot{\psi}}{V_x}) \tag{5.10}$$

Rewriting in state space form:

State vector:

$$x = \begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix};$$

The input vector is:

$$u = \delta_f$$

The matrices are:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-2C_f - 2C_r}{mV_X} & 0 & -V_x + \frac{-2C_f l_f + 2C_r l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-2C_f l_f + 2C_r l_r}{I_z V_X} & 0 & \frac{-2C_f l_f^2 + 2C_r l_r^2}{I_z V_x} \end{bmatrix}; B = \begin{bmatrix} 0 \\ \frac{2C_f}{m} \\ 0 \\ \frac{2l_f C_f}{Iz} \end{bmatrix}$$

Considering:

$$\dot{x} = Ax + Bu$$

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-2C_f - 2C_r}{mV_X} & 0 & -V_x + \frac{-2C_f l_f + 2C_r l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-2C_f l_f + 2C_r l_r}{I_z V_X} & 0 & \frac{-2C_f l_f^2 + 2C_r l_r^2}{I_z V_x} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{2C_f}{m} \\ 0 \\ \frac{2l_f C_f}{Iz} \end{bmatrix} u \quad (5.11)$$

This model describe the vehicle in function of geometric parameters, vehicle characteristic and longitudinal speed.

5.2.1 Inter-vehicles dynamics

Now that we have established the vehicle dynamics as a function of typical vehicle parameters, geometry, and longitudinal velocity, we need to define the dynamics that describe the interaction between the vehicles of the platoon.
To start, let's consider a Leader vehicle (LV) and just one follower vehicle (FV).

To do so,we need to define the concept of error within the platoon, in this thesis was considered as reference the following [38], [42].

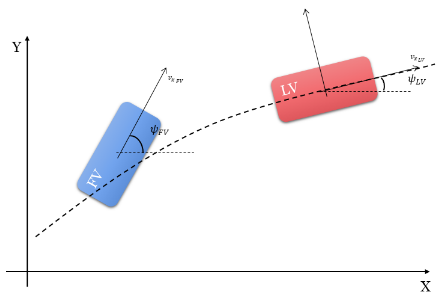

Figure 5.5: Inter-vehicles parameters

Defining:

- ψ_{FV}: angle between v_x and global axis X
- ψ_{LV}: angle between v_x and global axis X

Error Definition

- **Heading error** ψ_e, error between the Yaw angle of the FV and the Yaw angle of the LV.

$$\psi_e = \psi_{FV} - \psi_{LV}$$

  Derivative:

$$\dot{\psi}_e = \dot{\psi}_{FV} - \dot{\psi}_{LV} \tag{5.12}$$

- **Lateral error** y_e, error between the actual position and the expected position of the vehicle, can be defined using the heading error.

$$\dot{y_e} = v_x sin(\psi_e) + v_y cos(\psi_e)$$

Considering small heading error:

$$\dot{y_e} = v_x \psi_e + v_y \tag{5.13}$$

In the end putting everything together,the model that describe the single vehicle dynamics and the model that accounts for the inter-vehicles dynamics and considering as system states and input:

$$x = \begin{bmatrix} \dot{y}_{Fv} \\ \dot{\psi}_{Fv} \\ y_e \\ \psi_e \end{bmatrix}$$

$$u = \delta$$

$$\dot{x} = Ax + Bu + C\dot{\psi}_{Lv}$$

Is obtained the final model that describe the system.
Then rewriting in state-space form:

$$\frac{d}{dt}\begin{bmatrix}\dot{y}_{Fv}\\ \dot{\psi}_{Fv}\\ y_e\\ \psi_e\end{bmatrix} = \begin{bmatrix}\frac{-2C_f-2C_r}{mV_X} & -V_x + \frac{-2C_f l_f+2C_r l_r}{mV_x} & 0 & 0\\ \frac{-2C_f l_f+2C_r l_r}{I_z V_X} & \frac{-2C_f l_f^2+2C_r l_r^2}{I_z V_x} & 0 & 0\\ 1 & 0 & 0 & V_x\\ 0 & 1 & 0 & 0\end{bmatrix} x + \begin{bmatrix}\frac{2C_f}{m}\\ \frac{2l_f C_f}{Iz}\\ 0\\ 0\end{bmatrix} u + \begin{bmatrix}0\\ 0\\ 0\\ -1\end{bmatrix} \dot{\psi}_{Lv} \tag{5.14}$$

Let's now consider the whole platoon, in general the state equation can be written as:

$$\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t) + C_i d_i(t) \tag{5.15}$$

where:

$$x_i(t) = \begin{bmatrix}\dot{yi}_{Fv}(t)\\ \dot{\psi}_i(t)\\ y_{ei}(t)\\ \psi_{ei}(t)\end{bmatrix}; u_i(t) = \delta_(t); d_i(t) = \dot{\psi}_{i-1}(t - \Delta t)$$

Chapter 6

Model Predictive Control design

This chapter is about the analysis and implementation of a controller for realizing the control of the lateral dynamics of a single vehicle.
With regard to the choice of controller to be used, several studies were analysed and according to the literature MPC has emerged, in the last years as a major tool for control of autonomous vehicle. The MPC (Model Predictive Controller), implements a feedback control algorithm that uses the model to make predictions about future optimal outputs. Predictions are made by solving an online optimization problem at each time step. In [37] MPC offers some interesting features:

- Is able to handle MIMO system, is a multi variable controller that computes the outputs considering the interaction between the the inputs and the outputs of the system but also the interaction between all the variables in the problem.
- Allow to set multiple hard constraints that most of the time are in conflict with each other.
- Has the capability of predict the future control action.
- Is able to handle complex systems and offers a powerful control strategy

On the other hand some of the disadvantages of using MPC with respect to other controller are:

- Computation complexity, since the MPC runs in real time the optimization problems this can create some issues in terms of computational power and time.
- The effectiveness of the MPC is strongly related to the prediction model used, if is inaccurate the control strategy could not work properly.

6.1 MPC theory

Model Predictive Control method uses the dynamical model of the Plant to predict the future behaviour of the variables of interest computing at the end the optimal control input.
From a mathematical point of view is obtained using a constrained finite horizon control problem and the Receding Horizon principle.

Considering a Discrete Time, Linear Time Invariant dynamic model:

$$x(k+1) = Ax(k) + Bu(k)$$

where:
$x(k)$: state at time k
$u(k)$: control input at time k
At generic time k the MPC optimization problem is defined as:

$$\begin{aligned} \min_{U(k|k)} \quad & J(x(k), U(k)) \\ \text{s.t.} \quad & x(k+1) = Ax(k) + Bu(k) \\ & u_{min} \leq u(k+1|k) \geq u_{max} \\ & y_{min} \leq y(k+1|k) \geq y_{max} \end{aligned}$$

- The cost function can be written in the simplest case as:

$$\begin{aligned} J(x(k), u(k)) = \sum_{i=0}^{H_p-1} & x^T(k+1|k)Qx(k+1|k)+ \\ & + u^T(k+1|k)Ru(k+1|k) \\ & + x^T(k+H_p|k)Sx(k+H_p|k) \end{aligned}$$

  In this case the first term accounts for the state, the second term accounts for the control input and the last term considers the behaviour of the state at the end of the prediction horizon.
  The cost function can be modified if needed, for example can be added a term that accounts for the output tracking.

- $U(k|k)$ identify the sequence of future control inputs:

$$U(k|k) = [u(k|k), u(k+1|k), ..., u(k+H_p|k)]^T$$

- H_p is the prediction horizon.

- MPC problem is able to handle both constrained on the input and on the output.

The Receding horizon principle defines an iterative procedure to solve this problem:

1. Consider the state $x(k) = x(k|k)$
2. Solve the QP problem wrt the control input sequence $U(k|k)$
3. Find the minimizer:

$$U^*(k|k) = [u^*(k|k), u^*(k+1|k), ..., u^*(k+H_p|k)]^T$$

4. Apply only the first element of the sequence $u^*(k|k)$
5. Repeat the process for all the k

In this way the MPC computes the optimal control input $u(k) = u^*(k|k)$ to be fed to the plant that we want to control.

6.2 Control problem formulation

The state space plant model has the following formulation:

$$x(k+1) = Ax(k) + Bu(k) + Ed(k)$$

$$y(k) = Cx(k) + Du(k)$$

Considering this state space vehicle model, the controller finds the optimal input sequence minimizing the cost function,:

$$\begin{aligned} J(k) = \sum_{i=0}^{H_p-1} & \{\{w_{ye} y_e(k+1|k) - 0]\}^2 + \\ & + \{w_{\psi_e}[\psi_e(k+1|k) - 0]\}^2 \\ & + \{w_u[u(k+1|k) - u_{target}(k+1|k)]\}\} \end{aligned}$$

where:

- w_{ye} is the weight assigned to the lateral error
- w_{ψ_e} is the weight assigned to the heading error
- w_u is the weight assigned to the control variable.

Stability

In platooning, the concept of stability is based on two different notions:

- Individual vehicle stability: stabilize its lateral movement.
- String stability: when the error between two following vehicles do not propagate along the platoon.

In particular the focus will be on the string stability. In the paper [13], the concept of string stability is analysed in the case of combined longitudinal and lateral control. A further distinction is also made between LF and PF string stability.

- Leader- following string stability:

$$max|q_{e},_{j}\,| \leq \alpha_{j} max|q_{e},_{leader}\,|\alpha \in (0{,}1)$$

  Thus the position error of the j-th vehicle is compared with the position error of the leader.

- Predecessor-following string stability:

$$max|q_{e,j}| \leq \alpha_{j} max|q_{e,j-1}|$$

  the position error of the j-th vehicle is compared with the position error of the preceding vehicle.

Considering the second case, reformulating for our case, the string stability problem can be written as:

$$\max|y_{e}, i| \leq \alpha|y_{e},_{i-1}\,|$$

Constraints

The vehicle steering angle is limited due to physical limitation.

$$u_{i,min} \leq u_{i} \leq u_{i,min}$$

Constraints on the outputs:

$$y_{e,i,min} \leq y_{ei} \leq y_{e,i,min}$$

$$\psi_{e,i,min} \leq \psi_{e,i} \leq \psi_{e,i,min}$$

In the end, putting everything together the the control problem formulation is:

$$\begin{aligned} u^{*}(k) = \min_{U(k|k)} \quad & J(k) \\ \text{s.t.} \quad & u_{min} \leq u(k+1|k) \leq u_{max} \\ & y_{e,i,min} \leq y_{ei}(k+1|k) \leq y_{e,i,min} \\ & \psi_{e,i,min} \leq \psi_{e,i}(k+1|k) \leq \psi_{e,i,min} \end{aligned} \tag{6.1}$$

6.3 Controller Design in Simulink

In order to realize the MPC controller in Simulink can be used the Model Predictive Control Toolbox.

Basically, the controller needs as input a model of the plant to be controlled and reference values. Using these elements, the controller calculates a sequence of control actions such that the cost function is minimised over a defined time. The sequence is found by solving an optimisation problem In order to function, the MPC block requires a number of fundamental elements such as the model of the plant to be controlled, the basic parameters, cost function weights and constraints.
The development was conducted in several stages.
The first stage was the evaluation of the dynamical model of the platoon, which is covered in detail in Chapter 5, the final result is the state space representation of the model, 5.15.
To fully define the system,the vehicle physical parameters must be specified We decided to use the vehicle defined in the previous project[1]. It is a Fiat 500e with the following characteristic:

Mass of the vehicle	$m = 1474[kg]$
Cornering stiffness front wheel	$C_f = 100000[kN/rad]$
distance from front wheel to CoG	$l_f = 1520[mm]$
Cornering stiffness rear wheel	$Cr = 80000; [kN/rad]$
distance from rear wheel to CoG	$lr = 1400; [mm]$
inertia	$Iz = 2700; [kgm^2]$

Table 6.1: Vehicle parameters

Once the model and all its parameters and characteristics have been defined, the next step is the definition of the controller parameters.
Basic parameters[43]:
Scale factor- is a parameter used to approximate the range of the variable. It is therefore recommended to change this parameter if there are significant differences between the input and output variables of the plant. In particular, according to the documentation, span means the difference between the maximum and minimum value of a variable in engineering units. It is therefore recommended to change the values of the scaling factors so that all signals have the same unit order. This should make it easier to tune the MPC weights later. In order to choose the right value, it is recommended to check the upper and lower bounds of the value and choose the scale factor value based on the difference.
Sample time - Rate at which the controller executes the control algorithm. If it is too high, the controller will not be able to follow disturbances fast enough. If it is too low, the controller will react well to any disturbances, but the computational load will be high. So it is important to find the right trade-off,

generally, as the matlab documentation suggests is common to have between 10 and 20 samples in the rise time open loop.
Prediction horizon- number of predicted future time steps, essentially tells us how far the controller is able to see in terms of time steps. Here too, it is necessary to choose the right value by making a compromise between a low and high value. A value that is too low does not allow the controller to 'see' the full dynamics, but a value that is too high wastes computational power.
Control Horizon-parameter defined in time steps. It indicates the number of control moves to be optimised at each k interval. So the controller only uses the first m inputs, the others are discarded. So essentially it tells us the value

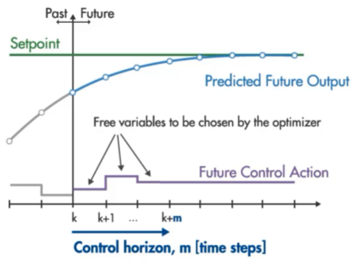

Figure 6.1: Control horizon

of variables to be optimised at each step k. It is suggested to keep the value between 10% ÷ 20% of the prediction horizon.

Chosen parameters after trial and error procedure.

Sample time	0.1
Prediction horizon	10
Control Horizon	2

Table 6.2: MPC basic settings

The next step was to tune the controller.
First the weight and relative weights was assigned to each variables. The weights of the variables are essential to the functioning of the mpc because in the cost function several parameters are indicated which correspond to the variables to be optimised, often these parameters are also in conflict with each other so by assigning the right weights to the various components, a balance can be found between the various performances to be optimised.
In this case, i used the following values: for the lateral error 0.5 and for the heading error 1. And for the steering a rate weight of 0.1 was used.

Finally, constraints are defined.
Two types of constraints can be defined Hard and Soft.

In general, according to the matlab documentation, it is recommended not to use hard constraints on both input and output variables because this can lead to infeasibility of the optimization problem. In general is specified to use Soft constraints on the output. Relaxing constraints is possible, through the dialog, by modifying the Equal Concern for Relaxation (ECR). In particular using a positive value. In fact, the higher the value, the greater the relaxation imposed on that constraint. Therefore, the greater the possibility that the controller will violate the constraint to achieve the desired performance.
Constraint values :
Hard constraint on the input:

$$-15deg \le u_i \le 20deg$$

Soft constraint on the lateral error:

$$-1m \le y_e \le 1m$$

Soft constraint of the Heading error:

$$-2deg \le \psi_e \le 2deg$$

As first attempt in order to become familiar with the use of the designer in Simulink, a simple case was implemented without the use of SUMO, so the reference values were obtained through the Driving scenario generator.
The developed controller showed good results. So the next step was the integration of SUMO and the evaluation of the controller's effectiveness by means of graphical simulation.
In particular, the first task was to replace the reference block. The control flow of the MPC block in simulink is as follows:

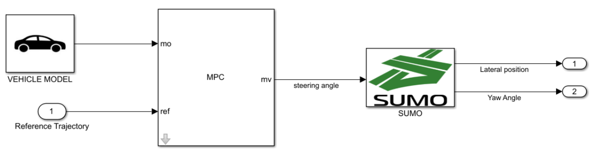

Figure 6.2: MPC control flow using SUMO

In this case, the variables relating to the leader vehicle obtained via SUMO were inserted as reference, which simulate the v2v communication technology and in particular the Leader following topology.
Having tested the correct operation of this block, the next task was to send the

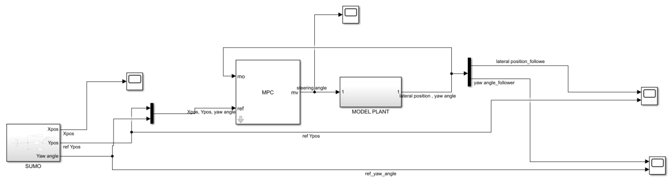

Figure 6.3: Simulink framework - SUMO

control input, i.e. the optimum steering angle value calculated by the MPC controller, to the follower vehicle. First, a simulation was carried out with only two vehicles:

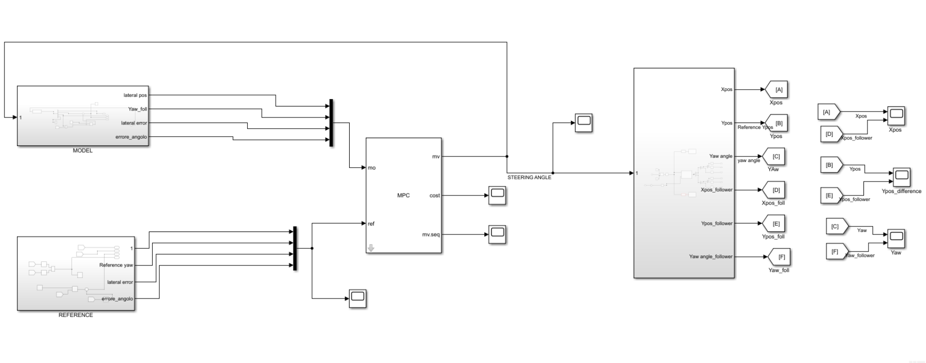

Figure 6.4: Simulink framework - 2 vehicle platoon

The optimum angle value was then imposed using Traci commands.
After testing that the controller was working properly, an additional vehicle was added to simulate a 3-vehicle platoon, with 1 leader and 2 follower.

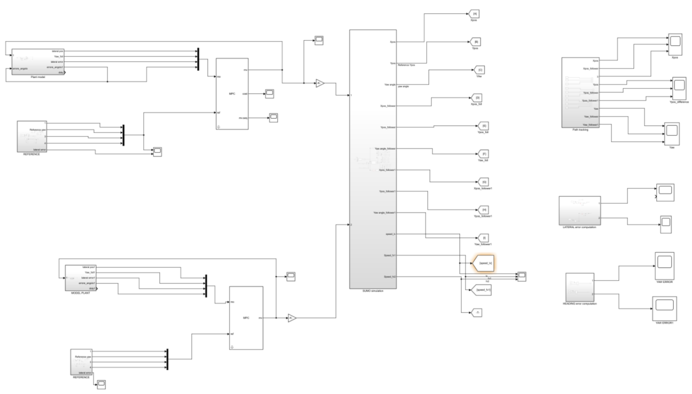

Figure 6.5: Simulink framework - 3 vehicle platoon

Chapter 7

Real world Simulation in SUMO

7.1 SUMO setup

As far as the final simulation is concerned, a suburban area was chosen as the area to be treated, which is simpler and also more suitable for a platooning situation, and unlike an urban area there are less complex stretches of road from the point of view of studying the platoon characteristics. In particular, the selected segment of road is very relevant for the Teoresi company in the context of a European project.
The specific route chosen is part of the Naples ring road.

Figure 7.1: Naples ring road scenario

The configuration files for the road section and vehicular traffic were provided to me by the company. Before carrying out the simulations it was necessary to make some changes and in particular it was decided to keep traffic flowing and insert other vehicles with the characteristics mentioned in 6.1, to simulate the platoon.
Two different simulations were carried out. In both simulation, since the focus in on the lateral dynamics, a constant longitudinal gap of $5m$ between the vehicles is imposed.

The first simulation analyses the entry of the platoon into the motorway from a side road. The second simulation instead analyses how the platoon behaves in a real traffic situation.

7.2 Results

First simulation: motorway entry This first simulation focuses on entering the motorway from a side road. T

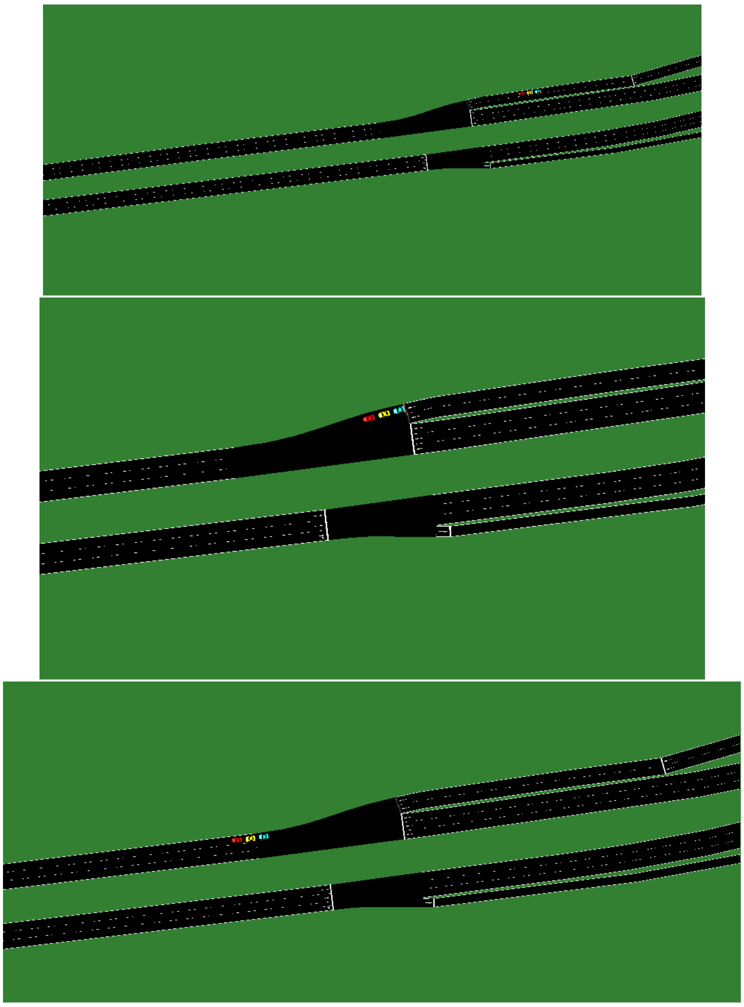

Figure 7.2: Snapshots from the simulation

The graphic simulation shows that the platoon can follow the curvature of the road perfectly. It can merge into the main road without any problems and without compromising the formation of the platoon.

Let's now analyze in the detail the experimental measurements.

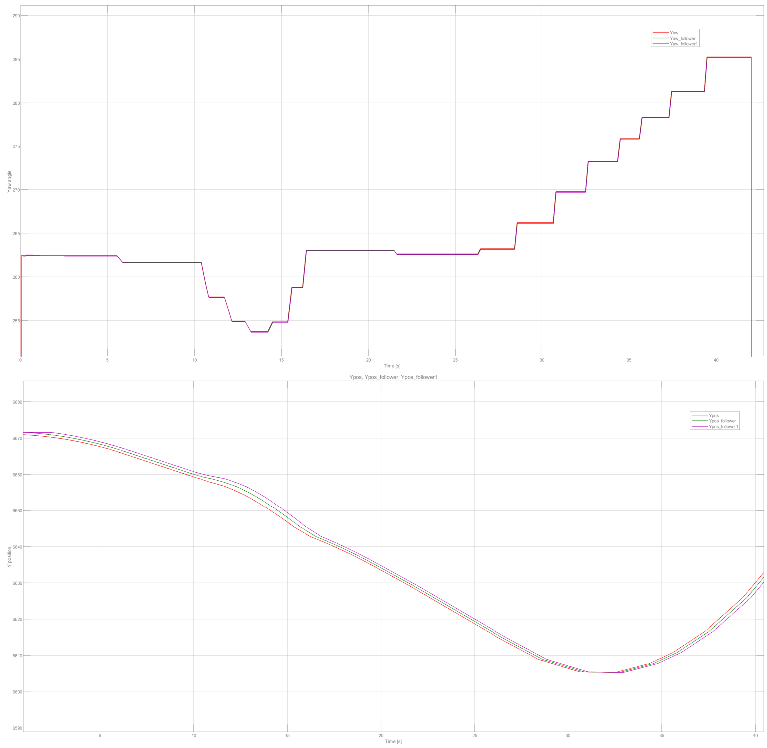

Figure 7.3: Yaw angle and Lateral position of the vehicles along the route

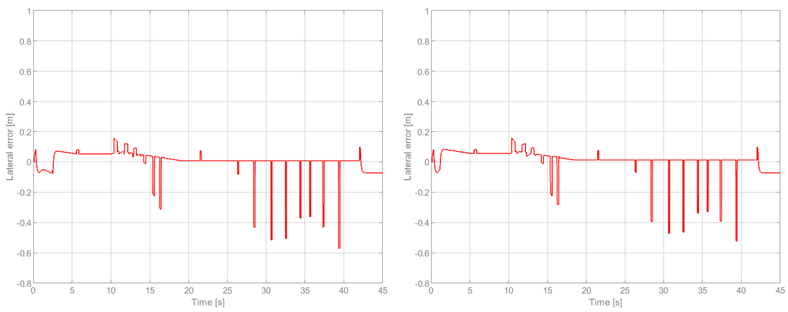

Figure 7.4: On the right: lateral error between LV and FV1, on the left: lateral error between FV1 and FV2

The lateral error as we can see is maintained at fairly low values between $0.2m$ and $-0.55m$ for the first two vehicles. While the error between the second and third vehicles is between $0.2m$ and $-0.5m$. So the vehicle can maintain lateral stability as the error decreases.

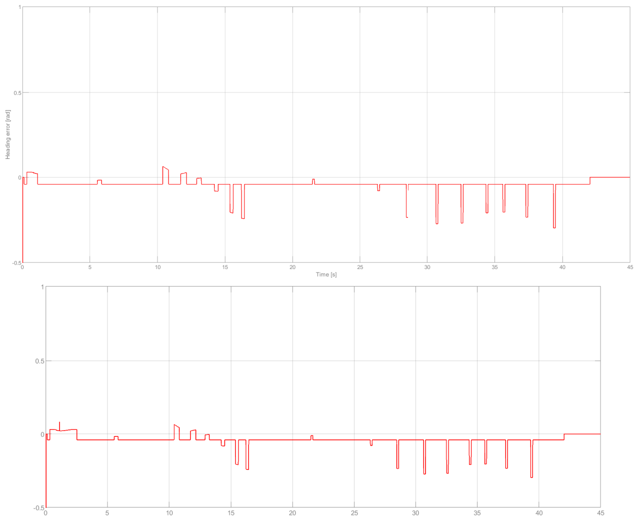

Figure 7.5: Heading errors

As far as the heading error is concerned, more than sufficient results were obtained, the error remained very low, the maximum value was 0.3 degree. These errors considering the analysed papers can be considered acceptable. Although there is room for improvement especially for the lateral error.

Second simulation: Platoon in traffic scenario

In this second simulation, a real traffic flow on the Naples road was added to the scenario. The aim is to analyse whether the platoon is able to follow the trajectory but also interact correctly with other vehicles in a more complex scenario.

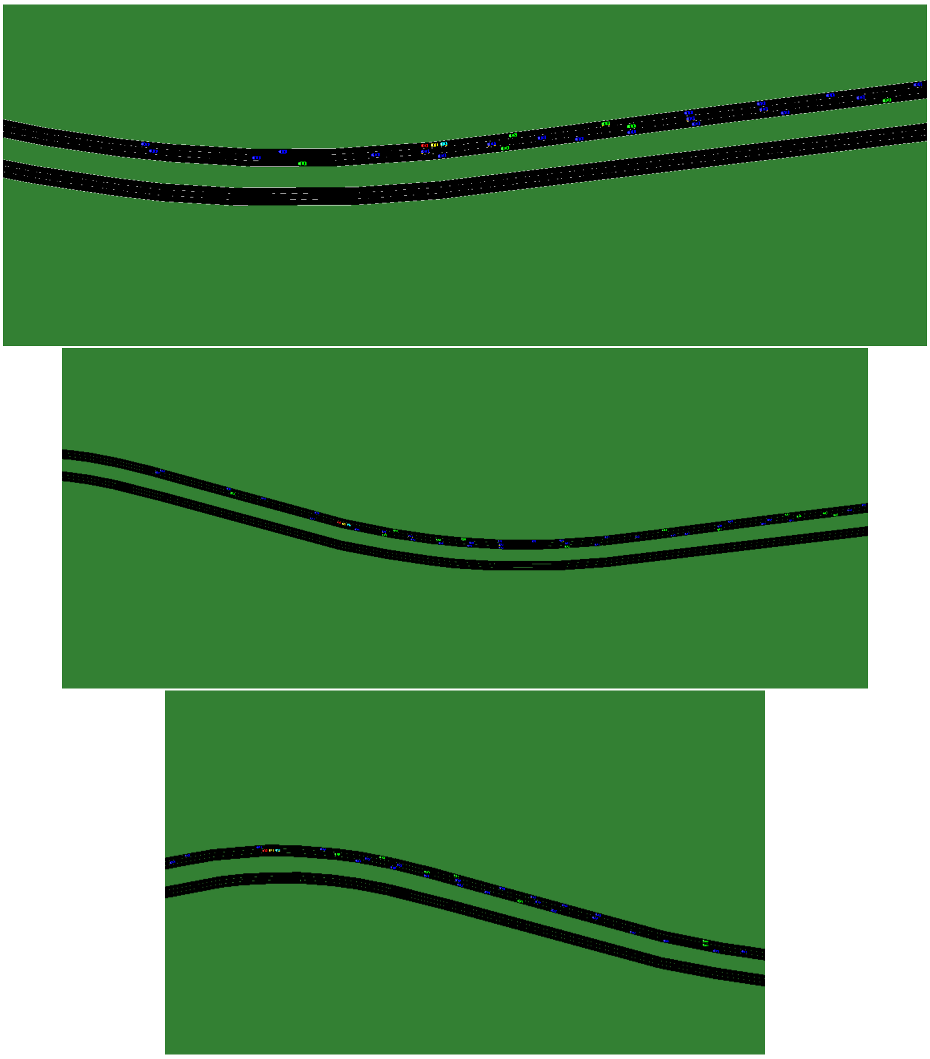

Figure 7.6: Simulation

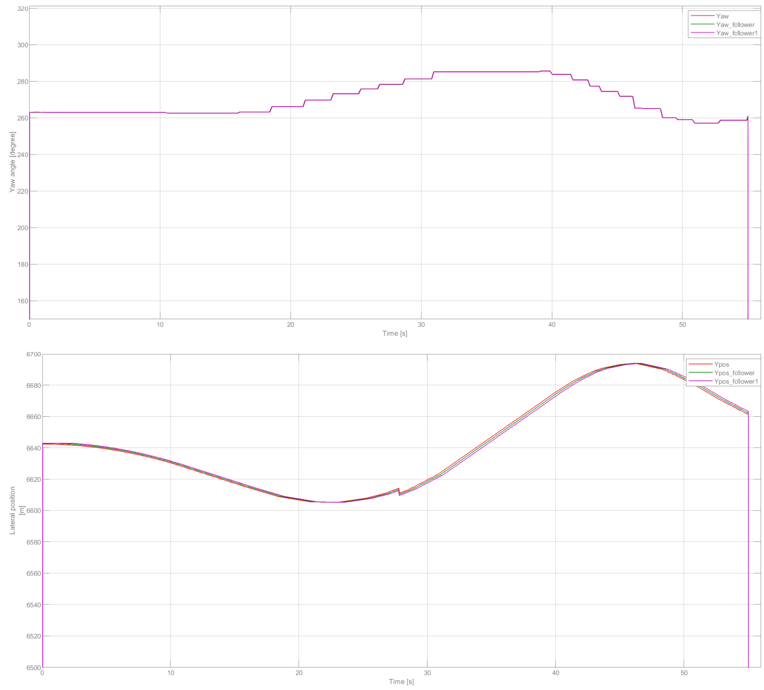

Figure 7.7: Yaw angle and lateral positions of the vehicles during the simulation

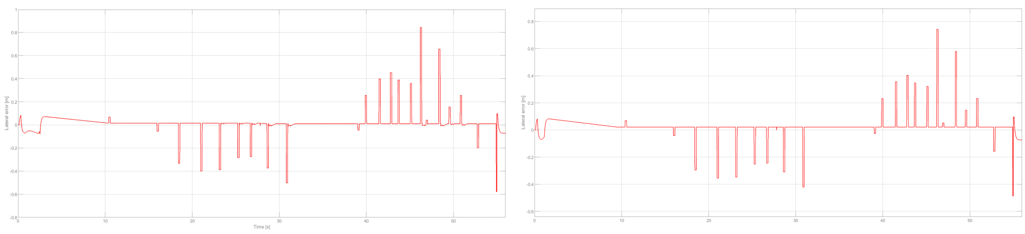

Figure 7.8: Later error

It is immediately noticeable that in this case the lateral errors are slightly higher than in the previous simulation. In particular, the maximum peak of the lateral error is $0.8m$, which is a slightly high value, but this may be due to the resolution of Sumo with respect to the network representation. In fact, some very small curvatures are not well represented. However, it is always important to note that the error decreases, even if only slightly, as you move forward in the platoon. This is a sign that the stability line is being respected.

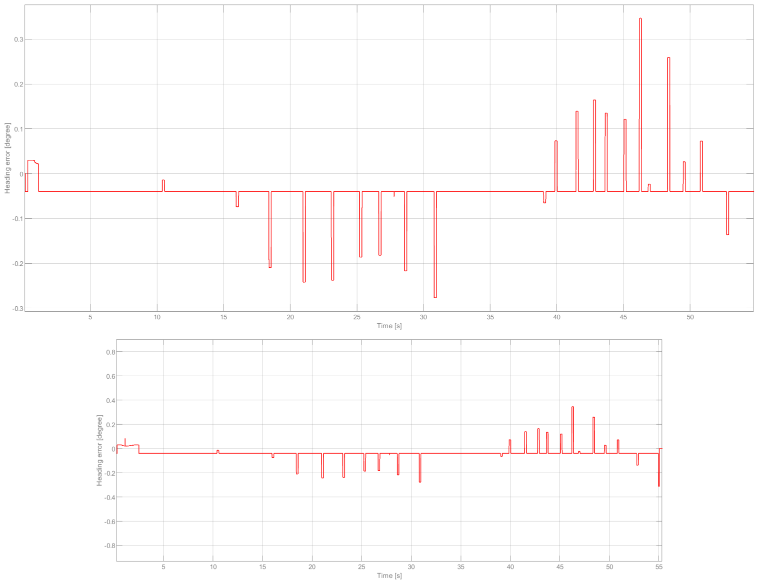

Figure 7.9: Heading error

The same considerations also apply with regard to heading error. The error remains very low, the maximum peak is 0.34 degrees. But as for the lateral error, compared to the first simulation there are generally higher values, the maximum peak indeed was around 0.3 degree. This effect may be due to the fact that it is a more complex simulation, with greater changes in curvature. Therefore, having slightly higher values can be considered acceptable.

Chapter 8

Conclusion

The initial aim of the thesis was to develop the work carried out in the previous thesis, mentioned above, by developing a control for lateral dynamics and carrying out simulations in a real scenario.
Ideally, it would have been possible to integrate the previous work on longitudinal dynamics with the lateral control system realized, but primarily due to a lack of time, but also because the computational effort required by this simulation would have been too high for the available hardware, it was not possible to integrate the longitudinal dynamics control realised in the thesis work. Instead, it was decided in the end to use the standard ACC model already implemented in SUMO as longitudinal control. Despite this, the results obtained are more than satisfactory: the follower vehicles are able to correctly follow the trajectory imposed by the leader vehicle, managing to maintain a very low heading error and lateral error. The proposed control strategy also ensure string stability of the entire platoon.
Another important result of this work has been the successful integration of Sumo and Simulink, bringing together the capabilities of the two software packages. Simulink makes it possible to create highly accurate vehicle models and platooning systems, while SUMO has great potential for real-world traffic simulation.

Possible future developments of this thesis could be numerous, also because the field of study is constantly evolving, as are the technologies available.
An interesting future analysis could be to implement different communication topologies and see the effect it has on the final control.
An in-depth study and analysis of platoon coordination techniques to optimize travel time or fuel reduction could also be interesting.
Or, still remaining in the control field, it would be interesting to try other maneuvers or implement a different type of controller perhaps also exploiting neural network technology.

Bibliography

[1] Antonio Capuano. «Development of an Adaptive Model Predictive Control for platooning safety in Battery Electric Vehicles». MA thesis. Politecnico di Torino, 2020/2021 (cit. on pp. 1, 16, 39).

[2] B Padmaja, CH VKNSN Moorthy, N Venkateswarulu, and Myneni Madhu Bala. «Exploration of issues, challenges and latest developments in autonomous cars». In: *Journal of Big Data* 10.1 (2023) (cit. on pp. 3, 4, 6).

[3] Rasheed Hussain and Sherali Zeadally. «Autonomous Cars: Research Results, Issues, and Future Challenges». In: *IEEE Communications Surveys Tutorials* 21.2 (2019). DOI: `10.1109/COMST.2018.2869360` (cit. on pp. 3, 12).

[4] Markus Maurer, J Christian Gerdes, Barbara Lenz, and Hermann Winner. *Autonomous driving: technical, legal and social aspects*. Springer Nature, 2016 (cit. on pp. 3, 4).

[5] *La vera storia dell'auto autonoma*. `https://www.wired.it/gadget/motori/2016/01/20/vera-storia-auto-autonoma` (cit. on p. 4).

[6] A. Broggi, M. Bertozzi, and A. Fascioli. «ARGO and the MilleMiglia in Automatico Tour». In: *IEEE Intelligent Systems and their Applications* 14.1 (1999), pp. 55–64. DOI: `10.1109/5254.747906` (cit. on p. 4).

[7] Long Chen et al. «Milestones in Autonomous Driving and Intelligent Vehicles—Part I: Control, Computing System Design, Communication, HD Map, Testing, and Human Behaviors». In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 53.9 (2023), pp. 5831–5847. DOI: `10.1109/TSMC.2023.3276218` (cit. on p. 5).

[8] Sebastian Thrun et al. «Stanley: The robot that won the DARPA Grand Challenge». In: *Journal of Field Robotics* (2006). DOI: `https://doi.org/10.1002/rob.20147`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20147` (cit. on p. 5).

[9] `https://waymo.com/about/story` (cit. on p. 5).

[10] *Your Autopilot has arrived*. ://www.tesla.com/blog/your-autopilot-has-arrived (cit. on p. 6).

[11] On-Road Automated Driving (ORAD) Committee. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. Apr. 2021. URL: https://doi.org/10.4271/J3016_202104 (cit. on p. 7).

[12] *SAE Standards News: J3016 automated-driving graphic update*. URL: https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic (cit. on p. 8).

[13] Rebecca Spicer, Amin Vahabaghaie, Dennis Murakhovsky, George Bahouth, Becca Drayer, and Schuyler St. Lawrence. «Effectiveness of Advanced Driver Assistance Systems in Preventing System-Relevant Crashes». In: *SAE International Journal of Advances and Current Practices in Mobility* 3.4 (Apr. 2021), pp. 1697–1701. DOI: https://doi.org/10.4271/2021-01-0869. URL: https://doi.org/10.4271/2021-01-0869 (cit. on pp. 9, 38).

[14] Worrawut Pananurak, Somphong Thanok, and Manukid Parnichkun. «Adaptive cruise control for an intelligent vehicle». In: *2008 IEEE International Conference on Robotics and Biomimetics*. 2009. DOI: 10.1109/ROBIO.2009.4913274 (cit. on p. 9).

[15] *Forward Collision-Avoidance Assist (FCA)*. URL: https://digitalownersmanual.hyundai.com/manual/CE/2023/en_US/idf2c50b37faa.html (cit. on p. 9).

[16] Vipin Kumar Kukkala, Jordan Tunnell, Sudeep Pasricha, and Thomas Bradley. «Advanced Driver-Assistance Systems: A Path Toward Autonomous Vehicles». In: *IEEE Consumer Electronics Magazine* (2018). DOI: 10.1109/MCE.2018.2828440 (cit. on p. 10).

[17] Sean Campbell, Niall O'Mahony, Lenka Krpalcova, Daniel Riordan, Joseph Walsh, Aidan Murphy, and Conor Ryan. «Sensor Technology in Autonomous Vehicles : A review». In: *2018 29th Irish Signals and Systems Conference (ISSC)*. 2018. DOI: 10.1109/ISSC.2018.8585340 (cit. on p. 11).

[18] Albert Demba and Dietmar P. F. Möller. «Vehicle-to-Vehicle Communication Technology». In: *2018 IEEE International Conference on Electro/Information Technology (EIT)*. 2018. DOI: 10.1109/EIT.2018.8500189 (cit. on p. 11).

[19] Vicente Milanes, Jorge Villagra, Jorge Godoy, Javier Simo, Joshué Perez, and Enrique Onieva. «An Intelligent V2I-Based Traffic Management System». In: *IEEE Transactions on Intelligent Transportation Systems* (2012) (cit. on p. 11).

[20] Veronika Lesch, Martin Breitbach, Michele Segata, Christian Becker, Samuel Kounev, and Christian Krupitzer. «An Overview on Approaches for Coordination of Platoons». In: *IEEE Transactions on Intelligent Transportation Systems* 23.8 (2022), pp. 10049–10065. DOI: 10.1109/TITS.2021.3115908 (cit. on pp. 13, 15).

[21] Steven E Shladover et al. «Automated vehicle control developments in the PATH program». In: *IEEE Transactions on vehicular technology* 40.1 (1991), pp. 114–130 (cit. on p. 14).

[22] Rajesh Rajamani, Han-Shue Tan, Boon Kait Law, and Wei-Bin Zhang. «Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons». In: *IEEE Transactions on Control Systems Technology* 8.4 (2000), pp. 695–708 (cit. on p. 14).

[23] Tom Robinson, Eric Chan, and Erik Coelingh. «Operating platoons on public motorways: An introduction to the sartre platooning programme». In: *17th world congress on intelligent transport systems*. Vol. 1. 2010, p. 12 (cit. on p. 14).

[24] Sadayuki Tsugawa. «An overview on an automated truck platoon within the energy ITS project». In: *IFAC Proceedings Volumes* 46.21 (2013), pp. 41–46 (cit. on pp. 14, 15).

[25] Antonius J. C. Schmeitz, Dehlia M. C. Willemsen, and Simon Ellwanger. «EU ENSEMBLE Project: Reference Design and Implementation of the Platooning Support Function». In: *IEEE Transactions on Intelligent Transportation Systems* (2023), pp. 1–16. DOI: 10.1109/TITS.2023.3276482 (cit. on pp. 15, 16).

[26] Arturo Davila and Mario Nombela. «Platooning - Safe and Eco-Friendly Mobility». In: *SAE 2012 World Congress Exhibition*. SAE International, Apr. 2012. DOI: https://doi.org/10.4271/2012-01-0488. URL: https://doi.org/10.4271/2012-01-0488 (cit. on p. 15).

[27] Hao Long, Arash Khalatbarisoltani, and Xiaosong Hu. «MPC-based Eco-Platooning for Homogeneous Connected Trucks Under Different Communication Topologies». In: *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2022, pp. 241–246 (cit. on p. 17).

[28] Pablo Alvarez Lopez et al. «Microscopic Traffic Simulation using SUMO». In: *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. URL: https://elib.dlr.de/124092/ (cit. on pp. 19, 20).

[29] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. «SUMO – Simulation of Urban MObility: An Overview». In: *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. Ed. by SINTEF & University of Oslo Aida Omerovic, RTI International - Research Triangle Park Diglio A. Simoni, and RTI

International - Research Triangle Park Georgiy Bobashev. ThinkMind, Oct. 2011. URL: https://elib.dlr.de/71460/ (cit. on p. 19).

[30] *SUMO Documentation.* https://sumo.dlr.de/docs/index.html (cit. on p. 19).

[31] Hani S Mahmassani. «50th anniversary invited article—Autonomous vehicles and connected vehicle systems: Flow and operations considerations». In: *Transportation Science* 50.4 (2016), pp. 1140–1162 (cit. on p. 20).

[32] Andres F. Acosta, Jorge E. Espinosa, and Jairo Espinosa. «TraCI4Matlab : Enabling the Integration of the SUMO Road Traffic Simulator and Matlab Through a Software Reengineering Process». In: *Modeling Mobility with Open Data.* Springer, 2015, 155170. URL: http://link.springer.com%20/10.1007/978%E2%88%923%E2%88%92319%E2%88%9215024%E2%88%926%209 (cit. on p. 22).

[33] URL: https://it.mathworks.com/help/simulink/slref/coder.extrinsic.html (cit. on p. 23).

[34] Uwe Kiencke. *Automotive Control Systems: For Engine, Driveline, and Vehicle.* eng. Netherlands: Springer Nature, 2005. ISBN: 9783540264842 (cit. on p. 27).

[35] Hanyang Zhuang, Xiaofeng Chen, Chunxiang Wang, and Ming Yang. «Lateral Control Method of the Follower Vehicle in Local Decentralized Platooning». In: *2021 5th CAA International Conference on Vehicular Control and Intelligence (CVCI).* 2021, pp. 1–5. DOI: 10.1109/CVCI54083.2021.9661166 (cit. on p. 28).

[36] Yassine Kebbati, Vicenç Puig, Naima Ait-Oufroukh, Vincent Vigneron, and Dalil Ichalal. «Optimized adaptive MPC for lateral control of autonomous vehicles». In: *2021 9th International Conference on Control, Mechatronics and Automation (ICCMA).* 2021. DOI: 10.1109/ICCMA54375.2021.9646218 (cit. on p. 28).

[37] Vivek K, Milankumar Ambalal Sheta, and Veershetty Gumtapure. «A Comparative Study of Stanley, LQR and MPC Controllers for Path Tracking Application (ADAS/AD)». In: *2019 IEEE International Conference on Intelligent Systems and Green Technology (ICISGT).* 2019. DOI: 10.1109/ICISGT44072.2019.00030 (cit. on pp. 28, 35).

[38] Zhixiang Weng, Panshuo Li, and Xingyan Mao. «Distributed Robust MPC Based Lateral Control of Vehicle Platoons». In: *2023 42nd Chinese Control Conference (CCC).* IEEE. 2023, pp. 6459–6466 (cit. on pp. 28, 32).

[39] Yang Liu, Changfu Zong, and Dong Zhang. «Lateral control system for vehicle platoon considering vehicle dynamic characteristics». In: *IET Intelligent Transport Systems* (2019). DOI: https://doi.org/10.1049/iet-its.2018.5504 (cit. on p. 28).

[40] Stefan Solyom, Arash Idelchi, and Badr Bin Salamah. «Lateral Control of Vehicle Platoons». In: *2013 IEEE International Conference on Systems, Man, and Cybernetics*. 2013. DOI: 10.1109/SMC.2013.776 (cit. on p. 28).

[41] Rajesh Rajamani. *Vehicle Dynamics and Control*. eng. Netherlands: Springer Nature, 2011. ISBN: 9781461414339 (cit. on pp. 28–31).

[42] Roozbeh Kianfar, Paolo Falcone, and Jonas Fredriksson. «A Distributed Model Predictive Control Approach to Active Steering Control of String Stable Cooperative Vehicle Platoon». In: *IFAC Proceedings Volumes* 46.21 (2013). 7th IFAC Symposium on Advances in Automotive Control, pp. 750–755. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20130904-4-JP-2042.00040. URL: https://www.sciencedirect.com/science/article/pii/S1474667016384646 (cit. on p. 32).

[43] *Understanding Model Predictive Control*. https://www.youtube.com/playlist?list=PLn8PRpmsu08ozoeoXgxPSBKLyd4YEHww8 (cit. on p. 39).