POLITECNICO DI MILANO

School of Industrial and Information Engineering
Master of Science Degree in Mechanical Engineering

Advanced NMPC-based Local Motion Planner for Autonomous Vehicles

Supervisor:
Prof. Francesco Braghin (Politecnico di Milano)
Co-Supervisor:
Dr. Laura Ferranti (TU Delft)

Candidate:
Claudio Molteni
Matr. 920010

Academic Year 2019-2020

Acknowledgements

First of all, I would like to warmly thank Dr. Laura Ferranti. She introduced me to this fascinating field of research more than a year ago and constantly provided help in such a wise, kind and timely manner whenever I needed it. Today, I can be proud of this result thanks to her.

Secondly, I would like to thank Professor Francesco Braghin for his advice and for making this work possible (as well as funded by a scholarship).

Thirdly, my family and friends. I thank you all for supporting me and being patience throughout the several months I spent far from home. After all, we realize what we need when it is missing.

Lastly, I apologize for not thanking individually all the people that actually took part in this work. I am sure every single question, exchange of views or expression of interest has shaped the final result.

I definitely do not thank the pandemic that is still threatening society, ruining the future of generations to come and compelled me to mostly work from my room.

This is the end of such an amazing journey across three great universities in three different countries that taught me a lot. I will never forget all the people I have met on the way because they somehow made me the person I am today and allowed me to understand who I want to be.

Milano, 05/04/2021 Claudio

Contents

List of Figures

Abstract

In recent years Autonomous Vehicles have been extensively researched and developed because of their potential socio-economical impact. They can transform both private and public transport sectors in terms of safety, comfort, accessibility and environmental sustainability.

Despite the remarkable amount of work carried out, the Autonomous Driving is still a challenging problem due to the multitude of tasks to execute at the same time, such as Motion Planning. New solutions need to be designed and tested to make it applicable on a large scale.

As joint work between Politecnico di Milano and Delft University of Technology, the aim of this thesis is to develop a novel Local Motion Planner based on Nonlinear Model Predictive Control. It consists of two main modules: (i) a path planning module that generates a collision-free trajectory; (ii) a trajectory tracking controller that ensures the vehicle is able to follow the collision-free trajectory. While the path planning module allows to deal with the dynamic environment surrounding the vehicle, the low-level path-following module allows one to account for the complex nonlinear dynamics of the vehicle when performing avoidance manoeuvres. The proposed method can be easily reconfigured according to the road conditions and the situation faced by the vehicle to maximize handling and comfort.

A tailored simulation environment is created to properly set-up and test the proposed motion planning unit. It allows to simulate different manoeuvres with vary road conditions and agents behaviour prediction strategies, while keeping high fidelity thanks to the Simscape Multibody environment.

The result is an innovative Local Motion Planner that can guide an autonomous vehicle throughout simplified urban traffic while ensuring safety and comfort.

Keywords: Autonomous Driving, Trajectory Planning, Collision Avoidance, Model Predictive Control.

Sommario

Negli ultimi anni i veicoli a Guida Autonoma sono stati soggetti a crescente ricerca e sviluppo per via del loro potenziale impatto socioeconomico. Sono congegni in grado di trasformare il settore dei trasporti privato e pubblico in termini di sicurezza, comfort, accessibilità e sostenibilità ambientale.

Nonostante la notevole quantità di ricerca svolta, la Guida Autonoma rappresenta ancora una sfida per via delle complicate operazioni che richiedono contemporanea esecuzione, come la Pianificazione di Traiettoria. Nuove soluzioni sono necessarie per renderla applicabile su larga scala.

In forma di collaborazione tra il Politecnico di Milano e l'Università Tecnica di Delft, lo scopo di questa tesi è sviluppare un innovativo Local Motion Planner basato su Controllo Predittivo del Modello Non Lineare. L'unità proposta consiste in due moduli principali: (i) un pianificatore di percorso che genera traiettorie prive di collisioni; (ii) un inseguitore di traiettoria che assicura l'effettiva esecuzione delle traiettorie proposte. Mentre il primo permette di destreggiarsi nell'ambiente dinamico circostante il veicolo, il secondo interviene per tenere in considerazione le dinamiche non lineari che scaturiscono in manovre evasive. Tale modulo può essere facilmente riconfigurato al variare delle condizioni stradali e della situazione che si presenta per massimizzare tenuta e comfort.

Un ambiente di simulazione dedicato è stato creato per regolare e testare la suddetta unità di pianificazione del movimento in tutte le sue capacità. In esso è possibile riprodurre diverse manovre con varie condizioni stradali e strategie di predizione del comportamento degli altri utenti, mantenendo fedeltà grazie all'ambiente Simscape Multibody.

Il risultato è un innovativo Local Motion Planner in grado di condurre un veicolo autonomo attraverso traffico urbano semplificato garantendo sicurezza e comfort.

Parole chiave: Guida Autonoma, Pianificazione di Traiettoria, Prevenzione di Collisione, Controllo Predittivo del Modello.

Introduction

Autonomous Driving is a technology that can deeply change modern mobility both at private and public level. It can fundamentally alter transportation systems by reducing deadly crashes, allowing mobility for the elderly and disabled, increasing road capacity and usage, saving energy, and reducing pollution [1]. Autonomous Vehicles (AVs) will lead to new economic trends related to a new mobility paradigm, such as on-demand rides, shared taxis and micro-mobility, basically disrupting many sectors. Last but not least, people daily life will be different since the driving task could be replaced by both work and leisure activities while enhancing safety at the same time. This is confirmed by growing research and investments in all AVs related sectors ranging from Original Equipment Manufacturers (OEMs) to components suppliers, from hi-tech start-ups to new companies active in sharing mobility and similar [2]. However, despite all these premises, there are still many issues regarding technology, costs, insurances, legal and privacy regulation that need to be solved. This has to be done before scaling the technology to avoid major harmful effects for the society as a whole, as well as not to jeopardize its development and adoption.

This chapter outlines the main reasons that will push Autonomous Driving to remarkable development in the coming years, with a brief analysis of the current commercial situation.

Safety

Autonomous Driving is expected to remarkably improve road safety by reducing crashes, which are mainly caused by human errors [3]. According to [4], 1.35 million of people lost their lives in road accidents in 2016, that is 3700 every day worldwide, which were caused by rapid urbanization, poor safety standards, lack of rules enforcement, people driving distracted, fatigued, under the influence of drugs or alcohol, or speeding and not wearing seat-belts or helmets. On top of that, road traffic injury is the leading cause of death for people aged between 5 and 29 years, while it is the eighth one for people of all ages: more human beings now die as a result of road traffic injuries than from HIV/AIDS, tuberculosis and diarrhoeal diseases. Figure 1 reports the whole ranking as well as the change between 2000 and 2016.

A study of crash records occurred between 2004 and 2008 indicated that crash avoidance technologies have considerable potential for preventing crashes of all severities, applying to more than a million crashes in the US annually. Lane departure warning/prevention systems and side view assist/adaptive headlights could display similar but lesser effects, while lane departure warning/prevention was found to be relevant for most fatal crashes. The author notes that simultaneous application of all four technologies could prevent or mitigate up to 1,866,000 crashes each year [3]. As a numerical indication, the estimates for the US are provided: for AV penetration rates of 10%, 50% and 90%, the authors project a corresponding 1.100, 9.600 and 21.700 lives saved (in the US) per year, considering that there were 36.560 deaths on the road in US in 2018 [5]. The cost of reduced accidents is estimated to be worth from 118 to 503 billion per year [6].

Accessibility

Not everyone all over the world can own and drive a car, or even safely share a ride, and this represents a big limitation. There are people renouncing or postponing medical treatments because of the distance from hospitals, basically because they need someone with a vehicle who drives them to the structure [6]. Both drivers and non-drivers are affected. As people age, functional limitations and disorders occur that can increase the crash rate of road users. This is particularly the case in the decline of motor functions like muscle strength, finely tuned coordination, and the ability to adapt to sudden changes in bodily position. There are few indications that a decline in visual and cognitive functions, as part of normal aging, also has road safety consequences [7].

Environmental impact

AVs can significantly reduce the mobility environmental impact. According to the US Department of Energy, automated cars can reduce energy consumption in transportation up to 90%. More than a quarter of greenhouse gas emissions come from automobiles, but with automated cars, that have redesigned computerized systems able to choose the most

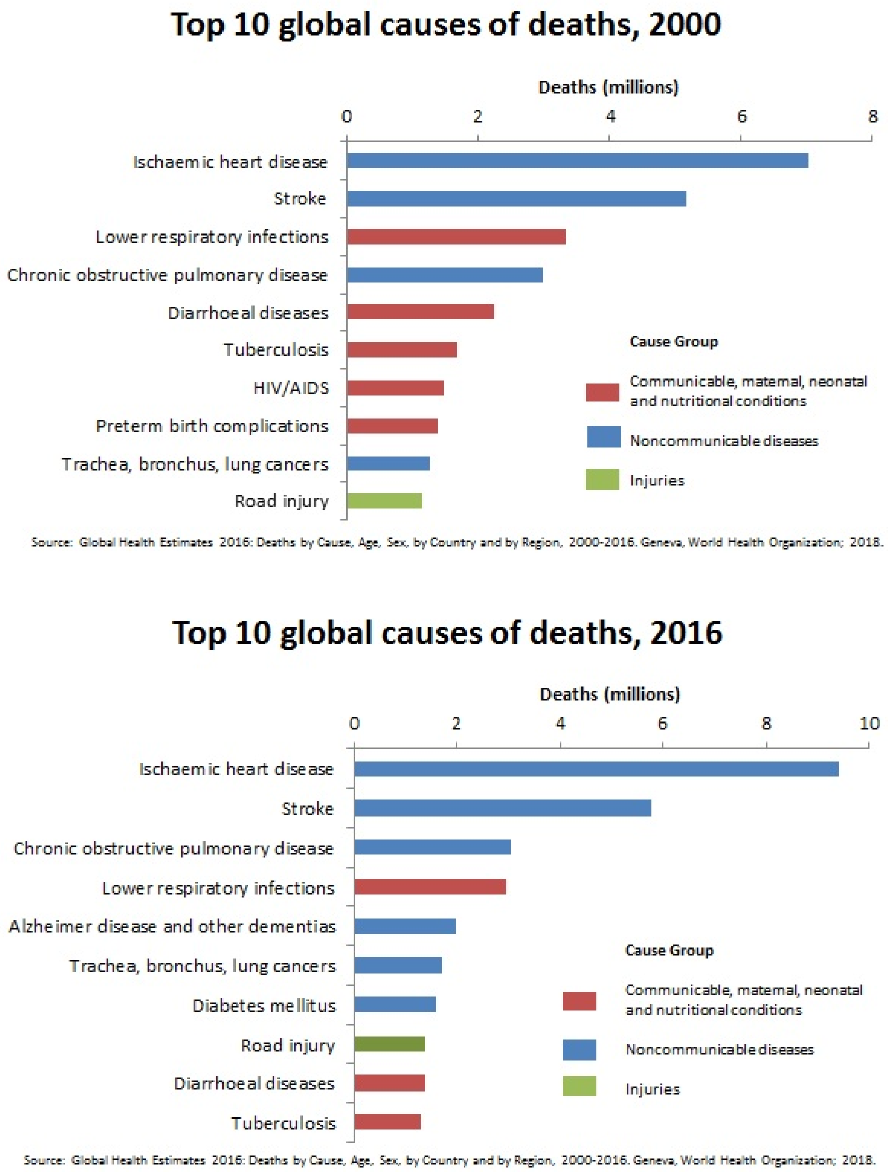


Figure 1: Top 10 death causes variation from 2000 to 2016 according to WHO. Road injuries shifted from 10th to 8th position exceeding diarrhoeal diseases and tuberculosis.

fuel efficient routes and travel faster, this situation can be changed. A self-driving car can move faster and more safely than a human-driven one, hence decreasing traffic congestion. These vehicles have built in adaptive cruise control and can automatically shift into electric mode to save gas. Excess safety equipment that are currently included in regular automobiles would not be needed anymore. Vehicles will then be lighter and able to run faster on the roads. Emissions will be reduced thanks to computerized systems able to accelerate and brake smoothly, and to choose the most fuel efficient route.

Automated cars have the ability to drive 70,000 to 80,000 miles per year which is around four to five more times further than regular privately owned vehicles that typically drive 12,000 to 15,000 miles per year. Self-driving vehicles would then need to be replaced every three to four years, leading to less maintenance and service costs. However, pricing should be carefully handled since it could lead to higher energy consumption (until 200%) because of rides affordability [8].

There are several other benefits coming from AVs diffusion, such as:

- Congestion reduction for which V2X communication is a key point;
- Parking lots saving since AVs are expected to be mostly shared and having high daily usage due to their cost;
- Private money saving as long as AVs usage become cheaper.

Automation Levels and Commercial Situation

As mentioned above, there is a large economic interest in Autonomous Driving: an increasing number of companies is working hard to develop fully self-driving vehicles (cars, robotaxis etc.) and related services (car sharing, e-hailing services, autonomous delivery etc.).

From a technical perspective, depending on the situations that can be handled and how the driver is engaged while the AV is running, different levels of Autonomous Driving can be distinguished. According to the SAE International classification J3016, which was updated in 2016, it is possible to distinguish 6 levels depending on what the driver is required to do. They are outlined in Figure 2.

A growing number of automotive related companies work in this field, and some of them are further along in terms of knowledge and real-world testing than others. However, it has to be pointed out as most companies

SAE level	Name	Narrative Definition	Execution of Steering and Acceleration/ Deceleration	*Monitoring* of Driving Environment	Fallback Performance of *Dynamic Driving Task*	System Capability *(Driving Modes)*
***Human driver* monitors the driving environment**						
0	No Automation	the full-time performance by the *human driver* of all aspects of the *dynamic driving task*, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	Driver Assistance	the *driving mode*-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	Human driver and system	Human driver	Human driver	Some driving modes
2	Partial Automation	the *driving mode*-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	**System**	Human driver	Human driver	Some driving modes
***Automated driving system* ("system") monitors the driving environment**						
3	Conditional Automation	the *driving mode*-specific performance by an *automated driving system* of all aspects of the dynamic driving task with the expectation that the *human driver* will respond appropriately to a *request to intervene*	System	**System**	Human driver	Some driving modes
4	High Automation	the *driving mode*-specific performance by an automated driving system of all aspects of the *dynamic driving task*, even if a *human driver* does not respond appropriately to a *request to intervene*	System	System	**System**	Some driving modes
5	Full Automation	the full-time performance by an *automated driving system* of all aspects of the *dynamic driving task* under all roadway and environmental conditions that can be managed by a *human driver*	System	System	System	**All driving modes**

Figure 2: AV levels of automation according to SAE international classification J3016l.

keep their state of development secret or publish it with a marketing-oriented fashion. Because of this, it is hard to identify the real level reached in terms of driving autonomy. As visible in Figure 3, which provides an overview of the current commercial situation, not only automakers play a key role in this field, but also the companies active in software, sensors and infrastructure commercialization.

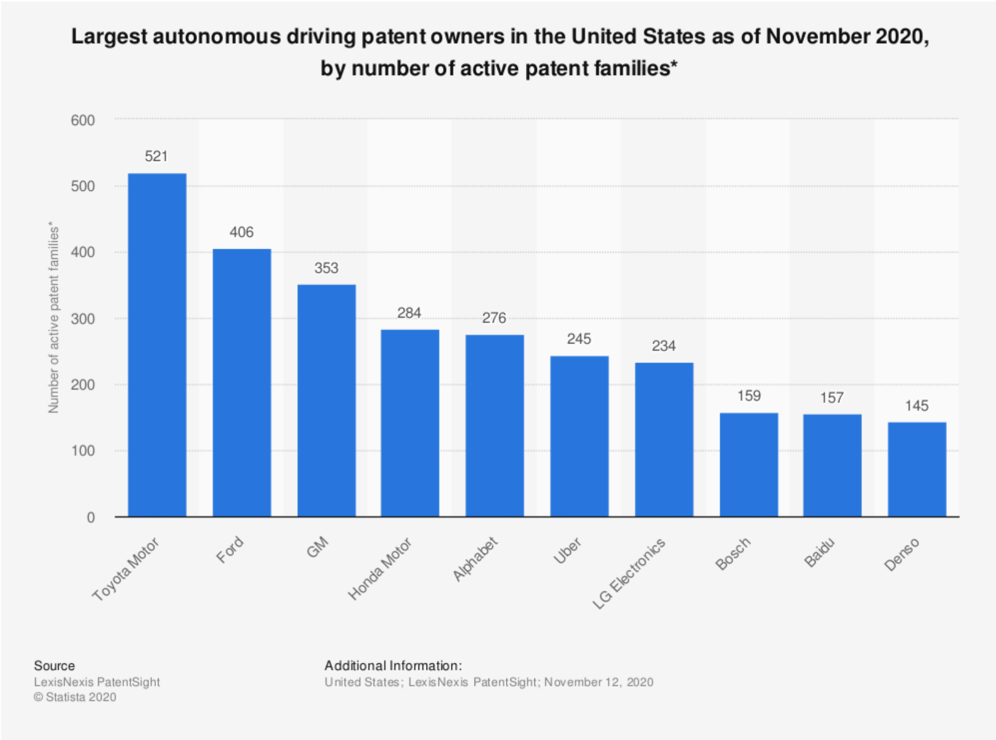


Figure 3: Largest autonomous driving patent owners in the United States as of November 2020, by number of active patent families [9].

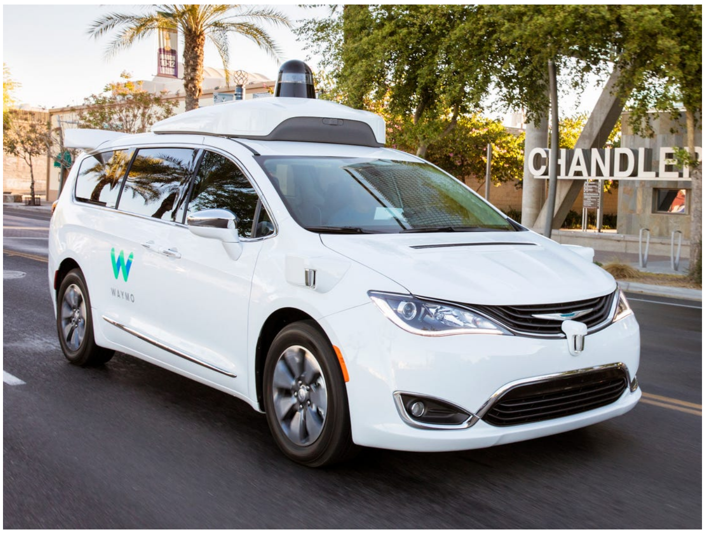


Figure 4: Waymo belongs to the Alphabet group. Currently, it is at the forefront of the Autonomous Driving development. In 2017 it launched the service "Early Rider" with which customers can pay to use its autonomous vehicles [10].

Figure 5: Argo AI, Ford's partner in developing its automated driving system is refreshing its fleet of test vehicles and expanding its reach into Detroit [11].

Figure 6: While automakers have been developing increasing levels of autonomous capability for their vehicles, Toyota has been more focused on safety systems that protect human drivers in extreme situations. But last month (September 2017) it took the wraps off the new autonomous vehicle platform being developed by its special R&D unit, Toyota Research Institute. Platform 2.1 advances the unit's quest for active safety intervention, and eventually a self-driving car, along a parallel development track. [12].

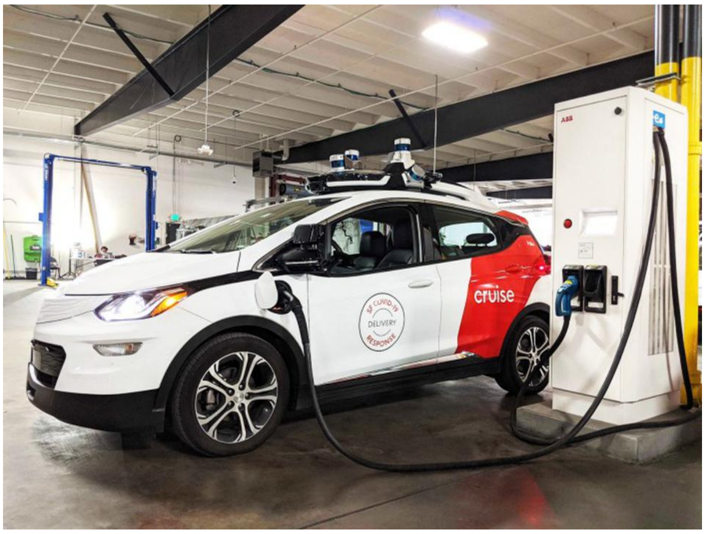


Figure 7: Cruise LLC, the automated driving company owned by General Motors and partners including Honda, Softbank and other investors is moving on to the next stage of its development program. In October 2020, Cruise received a permit from the California department of motor vehicles that allowed it to begin testing some of its vehicles without safety operators [13].

Figure 8: Daimler has already been testing Level 4 cars on public roads in Stuttgart, Germany. The California tests are the first time that occupants who are not employed by the company ride in the vehicles [14].

Thesis Objectives

The thesis work here presented aims to develop a novel NMPC-based Local Motion Planner for autonomous vehicles which is able to effectively operate in urban traffic. It relies on the SafeVRU platform [15] for the features that are required for experimental validation. Nevertheless, it is the result of a collaboration between Politecnico di Milano and Delft University of Technology.

The limits of a Path Planner based on a simple car bicycle model have been found out by testing it in a tailored simulation environment that has been specifically created. Since numerical techniques are defined as very promising for the motion planning task, the here designed Path Planner is based on Model Predictive Control (MPC). Thanks to a prediction model including accelerations and jerks control, a multitude of new features and an optimization problem with parametric structure, the resultant planner reaches really good collision avoidance capabilities by overcoming the previously mentioned limitations and keeps versatility.

Even more accuracy is reached through a combined Path Follower, that operates at high frequency and constantly communicates with the main planning module. Safety in enhanced by enlarging obstacles according to the estimated error between predictions based on different vehicle models. At the same time, by considering the car dynamics, it makes sure the controlled agent will follow the designed trajectories maintaining a good level of comfort.

In order to prove the Motion Planner reliability, it is tested in increasingly complex environments with multiple moving agents and reduced available of time to re-plan. Different prediction strategies for obstacles future behavior are available with corresponding computational burdens.

Thesis Outline

The thesis therein is structured as follows:

Chapter 1 introduces the Motion Planning task. A review of the current state of the art in Path Planning and Path Following techniques with relative limitations is reported, as well as a series of reasons behind the adoption of Model Predictive Control as main planning strategy. Future research directions are also provided.

Chapter 2 presents structure and operating mode of the realized Local Motion Planner. Key features are explained and motivated.

Chapter 3 starts with an introduction to Model Predictive Control, which is then specifically referred to the proposed Path Planner. Vehicle model, constraints, objective function and optimization process design choices are motivated.

Chapter 4 describes the designed Path Follower with the same approach. Design choices are motivated for both the MPC based and the PID-based Path Follower. Performance comparisons are also provided.

Chapter 5 described the created simulation environment in all its aspects. Different plants with variable pros and cons are available; a map reproducing a urban environment allows comprehensible simulations where multiple independent agents are involved; pseudo-detection strategies and additional settings are outlined.

Chapter 6 reports some of the tested scenarios. They are identified as the most representative for urban traffic situations. Numerical and graphical results demonstrate the performances of this Local Motion Planner.

Chapter 1

State of the Art

In recent decades commercial vehicles have become more and more "intelligent" with the main aim of aiding drivers in their tasks and improving safety. This has been possible by employing several additional features, such as Cruise Control, Adaptive Cruise Control, Emergency Braking and Blind Angle Vehicle Detection. They are known as Advanced Driving Assistance Systems (ADAS). Automated Vehicles are basically equipped with numerous more sophisticated extensions of them.

As a matter of fact, AVs have been subject to increasing interests over year, both from public and commercial sector. Companies like Waymo (Google), Ford Motor Company, Toyota, Tesla, Uber Technologies, General Motors and Daimler do not hide their interest in this field (as shown before), while major universities report their latest research progresses at well-known conferences.

The first works in motion planning date back to 1980, such as [16] and [17], and were primarily focused on computing a time-optimal and collision-free trajectory from a given point to another. Since then, many different methods and various successful implementations have been reported in literature.

The technical starting point of this thesis is the SafeVRU platform [15](Figure 1.1) which is able to plan collision-free trajectories in the presence of Vulnerable Road Users (VRUs). A more detailed description is reported in Appendix A.

1.1 Motion Planning

This section focuses on motion planning strategies which were initially developed for robots. It is considered a key aspect for robot navigation

Figure 1.1: TU Delft Vehicle Demo at the IV Conference 2019 [18].

since it provides global and local trajectory planning to command its behavior. It considers dynamic and kinematic models of the robot to go from a starting position to a final one. The main difference between vehicles and robots when executing motion planning is the presence of traffic rules with the first ones, while in the second case it is only necessary to cope with open environments without specific rules to respect [19].

Most of the authors divide the problem into *global* and *local* planning, basically discerning between algorithms which try to search the entire graph in real time and others which work with limited horizon both in terms of time and space. These planning techniques were classified in four groups, according to their implementation in automated driving: graph search, sampling, interpolating and numerical optimization. The module responsible for this task is called Path Planner (PP) or Motion Planner (MP).

Possibly the most popular technique used for on-road autonomous driving in the local search level is the one where a search space contains a certain geometric curve (e.g. clothoids or splines) and several lateral shifts of this curve. Each candidate path is then evaluated through a cost function with several considerations, such as distance and time costs, acceleration and collision checking [20].

In order to improve performance, the above mentioned Path Planner is usually combined with a Path Follower (PF), or Motion Follower, which basically selects appropriate actuators inputs through feedback to track the generated path.

The most relevant path planning algorithms implemented in motion planning for Autonomous Driving are described below, as well as some path following techniques.

To make the next descriptions clearer, here are some definitions:

- **Configuration Vector**: The set of independent attributes which uniquely define the position and orientation of the vehicle according to a fixed coordinate system.
- **Configuration Space**: The set of all vehicle configurations.
- **State**: The set of attribute values describing the condition of an autonomous vehicle at an instance in time and at a particular place during its motion. They can be position (x,y,z), orientation (ϑ_x,ϑ_y,ϑ_z),linear velocities (v_x,v_y,v_z), angular velocities (ω_x,ω_y,ω_z), linear accelerations (a_x,a_y,a_z), angular accelerations ($\dot{\omega_x}$,$\dot{\omega_y}$,$\dot{\omega_z}$) and others.
- **Path**: It is expressed as a continuous sequence of configurations beginning and ending with the boundary configurations, i.e. the initial configuration and the terminating configuration respectively.
- **Maneuver**: It is a high-level characterization of the motion of the vehicle, regarding the position and speed of the vehicle on the road. Examples of maneuvers include 'going straight', 'turning', 'overtaking' etc.
- **Maneuver Planning**: It addresses the problem of taking the best high-level decision for the vehicle while considering the path that should be followed. Within the vehicle architecture, the Behavioral Layer plays a relevant role in this regard.
- **Trajectory**: It is represented as a sequence of states visited by the vehicle, parameterized by time and, possibly, velocity.
- **Trajectory Planning**: It is concerned with the real-time planning of the actual vehicle transition from one feasible state to the next, satisfying the vehicle kinematic limits based on its dynamics and constrained by the navigation comfort, lane boundaries and traffic

rules. At the same time it is necessary to avoid obstacles including other road users, and consider ground roughness and ditches. It is also called Motion Planning.

1.2 Path Planning

This process within the whole Motion Planning task consists in generating a feasible path from a starting point to a final one, although it can also be an intermediate piece of the whole vehicle journey. This is done while taking into account information like current and future obstacles position, road rules and conditions, actuators limitations. The computed path and inputs are then sent to the PF as references.

Below some planning techniques are reported and briefly explained.

1.2.1 Graph Search Based Planners

It is possible to simplify the problem saying that the planning goal is moving the controlled vehicle from an initial point to the final one, while passing through a state space. This state space is often represented as an occupancy grid or lattice that depicts where objects are in the environment. Graph searching algorithms visit the different states in the grid, giving a solution (that not necessarily is the optimal one) or not (there is no possible solution) to the path planning problem.

- **Dijkstra Algorithm**. It is a well known algorithm existing in many variants. The original one simply tries to find out the shortest path overall while selecting the shortest path between the current node (called "source") and all the others in the graph. The configuration space is approximated as a discrete cell-grid space, lattices, among others. It has been successfully implemented in the Little Ben vehicle [21] which took part in the DARPA Grand Challenge [22].

- **A-Star Algorithm (A^*)**. It is really similar to the Dijkstra Algorithm except for the fact that it requires a heuristic, namely a cost function. Instead of simply looking for the closest node, the algorithm, for example, sums the beared cost to the estimated cost necessary to reach the destination. It appears costly in terms of memory and speed for large areas. In several applications it has been used as base for improvements, like in [23] for the DARPA Urban Challenge.

- **State Lattice Algorithms** The algorithm uses a discrete representation of the planning area with a grid of states (often a hyper-dimensional one). This grid is referred as state lattice over of which the motion planning search is applied. The path search in this algorithm is based in local queries from a set of lattices or primitives containing all feasible features, allowing vehicles to travel from an initial state to several others. A cost function decides the best path between the previously computed lattices.

1.2.2 Sampling Based Planners

The approach consists in randomly sampling the configuration space, or state space, looking for connectivity inside it. This kind of planners can work out a solution in a reasonable time in high dimensional spaces, but the outcome is sub-optimal. The most commonly used methods in robotics are the Probabilistic Roadmap Method (PRM) [24] and the Rapidly-exploring Random Tree (RRT) [25]. The latter has been extensively tested for Autonomous Vehicles.

1.2.3 Interpolating Curve Planners

Interpolation is defined as the process of constructing and inserting a new set of data within the range of a previously known sets (reference points). This means that these algorithms take a previously set of knots (like a given set of way-points describing a global road map), generating a new set of data (a smoother path) in benefit of the trajectory continuity, vehicle constraints and the dynamic environment the vehicle navigates. In the presence of obstacles, it suffices to generate a new path to overcome the event and then reentry the previously planned path. This allows the motion planners to fit a given description of the road by considering feasibility, comfort, vehicle dynamics and other parameters in order to draw the trajectory. The most used planners of this type are:

- **Lines and Circles**: Different segment road network can be represented by interpolating known way-points with straight and circular shapes.

- **Clothoid Curves**: Using clothoid curves is possible to define trajectories with linear changes in curvature, since their curvature is equal to their arc-length, as well as make smooth transitions between straight segments to curved ones and vice versa.

This technique is here used to interpolate the given way-points and create a reference for the PP.

- **Polynomial Curves**: These curves are commonly implemented to meet the constraints needed in the points they interpolate, i.e. they are useful in terms of fitting position, angle and curvature constraints, among others. The desired values, or constraints, in the beginning and ending segment will determine the coefficients of the curve.
- **Bézier Curves**: These are parametric curves that rely on control points to define their shape. The advantage of this kind of curves is their low computational cost, since the curve behavior is defined by control points.
- **Spline Curves**: A spline is a piece-wise polynomial parametric curve divided in sub-intervals that can be defined as polynomial curves, b-splines (that can also be represented in Bézier curves) or Clothoid Curves. The junction between each sub-segment is called "knot" and they commonly possess a high degree of smoothness constraint at the joints between the pieces of the spline.

See [19] for a more detailed explanation.

1.2.4 Numerical Optimization

In path planning, these methods aim to design trajectories, or smooth previously computed ones, by minimizing or maximizing a function subject to different constrained variables. This allows to effectively control safety and comfort with the downside of a not negligible computational burden. However, in recent years a lot of work has been done on software and hardware to improve computational performances.

The proposed Local Motion Planner belongs to this category, since it is based on Model Predictive Control (MPC) (explained in Section 3.1), which basically incorporates optimization concepts and aspects of control engineering. Within MPC, a model of the vehicle is used to compute optimal inputs by predicting its future behavior, which here means finding the best trajectory for the vehicle (PP in this work) or effectively following a given one (PF in this work).

This control strategy was born and has been widely used in the process industry for its interesting features. Nevertheless, it seems able to tackle most of the planning limitations (see Section 1.2.5) too, and its application in the automotive sector is indeed increasing. Despite the different scopes and sectors, the same theory lies behind this control technique. It means that multiple players can benefit from growing research.

In [15] the motion planning task relies on MPCC (Model Predictive Contouring Control) as done in [26], [27], and in [28]. Basically, the planning problem is formulated as a multi-objective constrained non-convex optimization problem, and its output is a collision-free path for the vehicle over a predefined time window. In [29] a similar approach is implemented but in simpler scenarios, with linearized dynamics and to test emergency maneuvers.

Model Predictive Control is subjected to growing interest in the automotive sector not only as planning tool. In [30] it is employed as active steering system to perform double lane change on a slippery road (the trajectory is given). In [31] some MPC applications at Ford Motor Company are described, with the underlying idea that this technology well matches the increasing target of performance, safety and emissions. For the same reasons it could be successfully employed with a driver model and help with the driving task, as described in [31].

As reported in [20], Model Predictive Control seems to be the most promising way to deal with the dynamic and uncertain nature of vehicle planning. Specifically, it looks interesting for the following reasons:

- Its finite horizon matches the sensors limited range;
- Its finite horizon reduces the computational effort which then allows to implement more detailed models and extensive algorithms;
- Its solution re-calculus at each time step satisfies the necessity of re-planning due to the fact that reality continuously changes (i.e. road users continuously move).

For this reasons and the growing interest in it (as reported in Section 1.2.4), this control technique is at the base of the designed Local Motion Planner.

1.2.5 Local Motion Planning Limitations

The above presented techniques can perform quite well but still have some limitations, as outlined in [20], that makes Autonomous Driving hard to achieve.

1. **Obstacles management**. All the previous approaches need to take into account other road users' intentions, basically by making predictions, since agents communication (V2X) is rarely present. As soon

as the obtained path is periodically re-computed, the same needs to be done with the just mentioned predictions, and the computational burden quickly compounds. Some approaches assume absence of uncertainty in agents movements (like in this thesis) but it is not possible in real world.

Another limitation in obstacles handling is the fact that, in order to reduce the required computational power, they are often represented as rectangles, circles or ellipses. Because of this, close proximity motion is hard to achieve. This simplification has also been applied in this work, but it is reasonable since close movements should be avoided for a safer collision avoidance.

Another major limitation in terms of obstacle handling is the inability to see around corners and detect obstacles, such as pedestrians and bicycles, approaching from blind spots. This disadvantage makes the planning algorithm take a 'cautious' and hence inefficient approach, like slowing down even in the absence of any obstacle.

2. **Vehicle dynamics**. Most planners use a simplified kinematic vehicle model (the so called Bicycle Model), which neglects aspects like tire forces to make the computation lighter. The downside of this choice is the inability to capture actual maneuver capabilities of a car, eventually resulting in a lack of precision. However, dynamics can be considered within the follower (like in this work) to compensate for it without making the computation too heavy.

3. **Risk indicators**. Risk indicators are metrics aiming to assess a collision risk in driving situations. Time to Collision (TTC), Distance to Collision (DTC) or Time to React (TTR) are the most common ones but they usually rely on constant velocity and constant acceleration, which is a limitation. More detailed strategies should be developed.

4. **Sensing and perception**. The vehicle is sometimes modeled as an isolated identity and with more knowledge of the surrounding than what it can have. Blind spots, bad weather conditions and unexpected obstacles can easily lead to a scenario with really limited visibility and high chance of failure in collision avoidance. Effective and expensive sensors together with agents communication could solve this issue, but proper testing is definitely necessary.

5. **Testing environment**. It is roughly possible to recognize three types of testing environments:

- Simulations
- Experiments with vehicle model
- Experiments in real-world

The first two are relatively economic and easy to implement but it is not possible to expect high accuracy. Because of that, experiments in real world with real traffic and variable weather conditions are necessary in order to get consistent improvements, which would be quicker to get if collected data was shared.

The designed Local Motion Planner can deal quite well with the above mentioned limitations, since it continuously updates obstacle predictions and keeps track of them, limits maneuvers aggressiveness (kinematic and dynamic behavior becomes similar), and has been tested through simulations in several scenarios. On top of that, thanks to its modular structure it can be easily interfaced with more advanced perception modules.

1.3 Path Following

The path following task consists in making the car follow the path generated by the PP. It is done by selecting appropriate actuators inputs while continuously taking the actual vehicle state (position, velocities, accelerations etc.) into account in form of feedback.

Among the possible solutions, the adoption of PIDs as PF is probably the simplest one. Basically, the input actions (steering and acceleration) coming from the PP constitute the references and are continuously compared with the actual inputs to determine the errors. These errors are then minimized according to the selected coefficients. This kind of control comes with quick tuning and good performances in most situations, and is part of the designed PF.

In [32] several Path Following techniques are presented with relative advantages and disadvantages, such as the "Path Stabilization and Trajectory Tracking Control for the Kinematic Model". It is also pointed out how MPC can better operate on slippery roads and emergency maneuvers thanks to a more accurate model. This translates into growing interest from the automotive sector for this task too, especially now that powerful optimization solvers allow real-time implementation.

In [33] a semi-closed form solution with minimal computational requirements is adopted. In [34] the performances of a NMPC PF incorporating the steering system dynamics and tested with a CarSim model are shown, while a predefined path, tracking errors and constant vehicle speed are provided. In [35] MPC is employed to control an Active Front Steering System and best follow the desired trajectory on a slippery road at the highest possible entry speed. Successive online linearizations of the nonlinear vehicle model are carried out, and an additional stability condition, in form of *ad-hoc* convex constraint for the LTV MPC problem, in formulated. The result is a PF fast enough to be effective up to 20 m/s on an icy road. A similar approach to operate in almost the same conditions is implemented in [36] with empathizes on the computational burden as reason behind the choice.

Among other techniques, there are more classical although elaborated ones, like *Pure Pursuit* ([37], [38]) which consists in an algorithm aiming to design a curve joining the current vehicle position to another one ahead on the path. [38] also emphasizes how MPC is able to provide very good results at the price of a (previously) large computational burden. A Liapunov based method with kinematic vehicle model is proposed in [39]. In [40] an innovative approach employing Spatial Dual Global Positioning System (GPS) system is tested to track paths (way-points and velocities) generated by a PP in real-time. The reached tracking error is of 16 cm, which is compatible with the 13 cm precision of the employed Spatial Dual GPS, whereas the longitudinal and lateral acceleration are within comfort levels as defined by available experimental studies. In [41] two different approaches based on Neural Network are presented and, according to the experiments carried out, the best of them can reach a lateral error smaller than 35 cm.

Compared to the above mentioned MPC based PFs, the designed one is precise, fast (solution in less than 5 ms in case of convergence), within the same car speed range (up to 20 m/s), on variable road conditions (such as sudden drop of friction), while keeping the dynamic model nonlinear and with realistic tire forces (Dugoff model). At the same time, future estimated distance from agents errors, are fed back to the PP. This allows to enlarge obstacles and improve safety. On top of this, robustness is enhanced thanks to the parallel PID based PF which does not fail.

1.4 Suggested Research Directions

Considering the previously mentioned limitations, some guidelines for future research in the Autonomous Driving field are:

- More work need to be done on Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I) (e.g. traffic lights) and Vehicle-to-Device (V2D) communication (for non-motorized traffic participants) because they can provide remarkable performance improvements. Indeed, relying on statistical predictions of other road users would not be so important anymore, although it could be used to compensate for sensors failures, for example, with bad weather conditions. All together, they are called V2X communication. On the other hand, high performance but rather expensive sensors are necessary to make autonomous vehicles reliable enough.

- New ways to include vehicle dynamics into the motion planning task should be found, especially when working on collision avoidance, although a kinematic bicycle model is computationally convenient and sometimes accurate enough. This because the resultant computed trajectories are more realistic and lead to smaller errors.

- Experiments in real-world with real traffic and road users are necessary for an effective validation.

- Multiple object tracking and movement anticipation is another important field of research basically because there are several factors that increase uncertainty in motion planning: road signals, appearance and disappearance of road users, different surroundings and high speed just to name a few.

In the next chapter my solution to overcome these limitations is presented and motivated while in the following ones its modules are thoroughly described.

Chapter 2

Proposed Local Motion Planner

In this chapter the overall structure of the proposed Local Motion Planner (LMP), or Motion Planning Module (MPM), is presented and motivated, its operating mode and safety features are explained, while its two main components (Path Planner and Path Follower) are thoroughly described in the following chapters.

The main task of this unit consists in planning collision free trajectories while considering vehicle control limits and comfort targets in a continuously changing environment.

2.1 Architecture

As visible in Figure 2.1, the designed LPM has a two-levels structure:

- **Path Planner (first level)** which is responsible for collision avoidance though long distance planning using a kinematic model of the car. It also takes care of passengers' comfort;

- **Path Follower (second level)** which aims to make the car follow the planned path considering its short distance dynamics (MPC-based) or smoothing the inputs (PID-based). In doing so, it considers road conditions and communicates with the PP.

The working principle of this structure is outlined in the next section.

2.2 Operating Mode

This Local Motion Planner goal is to design paths ensuring avoidance of both standstill and moving obstacles. This is done by applying optimization

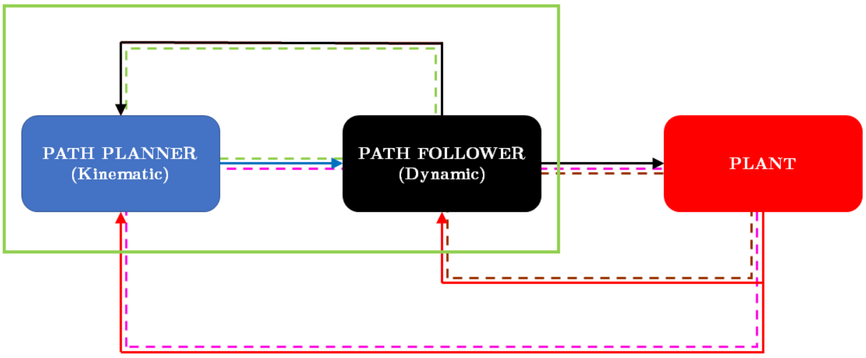


Figure 2.1: The Local Motion Planner two-levels structure: the PP represents the first level while the PF the second one. The Plant is next to the PF, that is connected to the LMP.

procedures to complex problems, for which optimal values can not always be found in the available time span (or can not be found at all). In Section 3.5 this topic is explained in detail.

Current commercial solvers, like ForcePro, can provide information regarding the executed optimization process at each time step. By using the solver diagnostic message (i.e., the solver exitflag), that is the convergence indicator, it is possible to take actions, which means sending inputs to the vehicle actuators, or to the PP itself, according to the optimization outcome. The main idea is that when an optimal path is reached (i.e., the optimizer converged to a locally optimum solution), applying the computed inputs is the best choice by definition. As shown in Figure 2.2, PP convergence is utilized to:

1. Decide which PF to use;

2. Take safety actions.

These two cases are explained in detail in the coming subsections.

2.2.1 Combined Path Follower

As previously anticipated, this LMP is equipped with a combined PF or, more precisely, a primary one which works by default and a backup

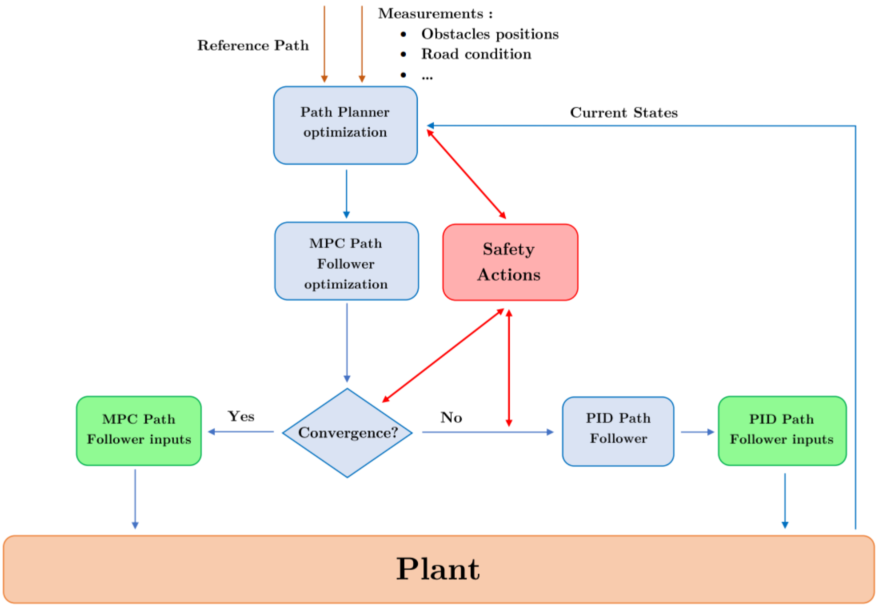


Figure 2.2: This flowchart shows how the designed Local Motion Planner works.

one that is activated in certain situations only (despite it always runs in parallel). In Chapter 4 they are exhaustively explained.

- A **MPC-based Path Follower** (primary)
- A **PID-based Path Follower** (backup)

Basically, both of them run but the inputs coming from the MPC-based PF are injected into the plant only when the its optimization process terminates with an optimal value, which means it converged (the solver provides this information). This is done for a limited time span even though the PP does not converge, basically because the trajectories it returns, that is the references for the MPC-based PF, might still be good enough.

When both the PP and the MPC-based PF do not converge, the same inputs of the last time step are applied once smoothed by the PID-based PF until any Safety Actions is triggered. This choice is motivated by the fact that the MPC PF performances are better overall, and it returns a feedback to the PP (in case of convergence). However, the PIDs always show good performances. Overall this Local Motion Planner results effective and robust.

2.2.2 Fail-Safe Architecture

Another key feature of this LPM is the capacity of taking additional measures to ensure safety when the situation faced is not under optimal control. Basically, this corresponds to repeated lack of convergence at the end of the optimization process. According to the PP solver outputs and the persistence of that output over time, a certain measure is taken, as shown in Figure 2.3.

In case of convergence to an optimal value, the optimal inputs and states are sent to the PF and the process is exactly the one explained in Section 2.2. Otherwise, in case of missed PP convergence, two different cases can be distinguished:

1. The problem is feasible but an optimum has not been found for some reasons. For example, the maximum number of iterations has been reached.
2. The problem is not feasible (with the current parameters, like the reference speed).

If the first situation shows up, the default operating mode goes on (see Figure 2.2) because the MPC PF can still handle the situation. If this situation persists for some time (0.7 s with the current setting), a reduced velocity is set as reference for the PP (like a fraction of the original one), in order to try to retrieve convergence and gain time. It is like a real driver would do in case of uncertain situation. If this attempt is not successful for a longer time (2 s with the current setting), the vehicle is forced to stop, again by setting a really low reference velocity. This is actually what happens at the end of the simulation in Section 6.1.2.

In case of unfeasible problem, the car is slowed down right after a short time span (0.1 s with the current setting) by imposing negative longitudinal jerk (until the minimum acceleration is reached), and this goes on for some time to avoid a jerky behavior. If the convergence is not retrieved, the vehicle stops in the same way described before.

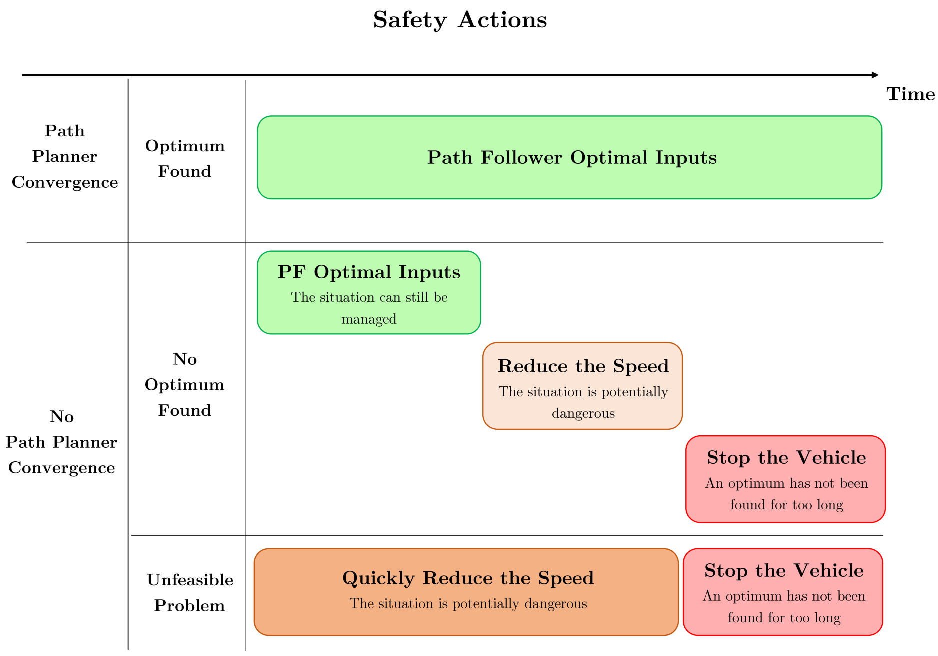


Figure 2.3: Adopted measures according to the solver output and its persistence over time.

Chapter 3

Proposed Path Planner

In this Chapter the designed Path Planner is described starting with a brief explanation of the general Model Predictive Control framework. The actual characteristics of this controller: system model, constraints, objective function and the optimization set-up are outlined and motivated. More emphasis is given to the system model and operating constraints due to their remarkable contribute. Overall this set-up is able to ensure collision avoidance as well as good comfort while taking handling limits into account. In Figure 2.1 the blue box represents the Path Planner.

3.1 Model Predictive Control

The first examples of model based control systems date back to 1970 and were realized mainly for the process industry, since it often deals with multi-variable processes, or Multi-Input Multi-Output systems. It then evolved to a technique able to ensure the optimal working within defined constraints. There are several reasons for increasing popularity of MPC since 1985, but constraints handling capabilities, easy extension to multi-variable processes and the possibility of maximizing the profit are definitely the main ones [42].

Nowadays, this approach is also applied in Autonomous Driving thanks to its ability to include non-linearities, inequality constraints and whatever objective function. on top of this, the growth of computational power enabled its usage like a feedback control (basically an optimal solution can be computed quickly enough to capture the system dynamics).

The MPC iteration, i.e. the set of operations executed at each time step are represented in Figure 3.0 and can be summarized as follows:

- The controlled vehicle actual state is measured (position, orientation, velocity etc.);

- The so called Optimal Control Problem with finite horizon is solved. In this specific case, by considering constraints and the vehicle model, as well as state predictions of other road users, the objective function is optimized and the respective car inputs for the whole horizon are worked out (Eq 3.43);

- From the above mentioned set of inputs, the ones associated to the first time step are sent to the vehicle actuators (in absence of PF) while the others are basically wasted;

- The optimization horizon if shifted forward by one step (in the sense that it now starts from the previous second step).

Five items which are always present in an MPC-based controller are:

1. Prediction model (and disturbance model)

2. Constraints

3. Performance index

4. Optimization

5. Receding horizon principle

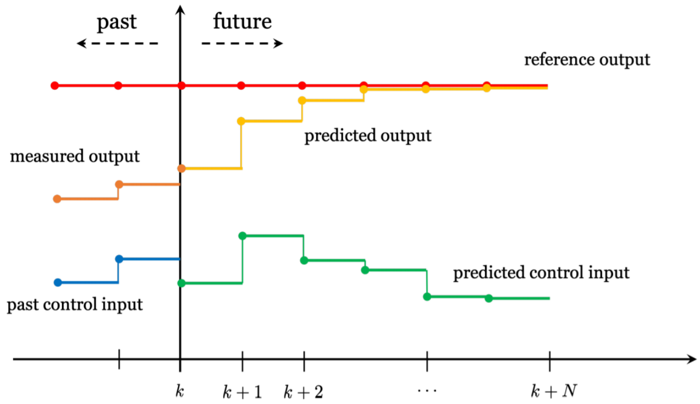
The proposed PP is inspired by the SafeVRU one [15] but brings improvements and changes to each of the just mentioned items. In the following sections they are analyzed and explained one by one for this specific project. A disturbance model has not been included in this work.

3.2 Prediction Model

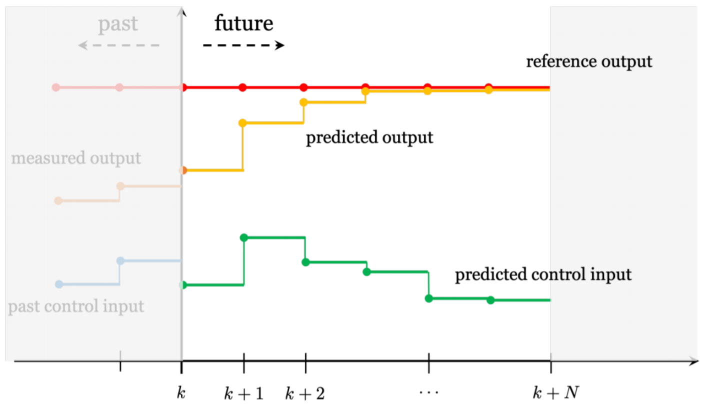
In general, the chosen plant model purpose is twofold:

- Prediction of future behavior (outputs) on the basis of inputs and known disturbances;

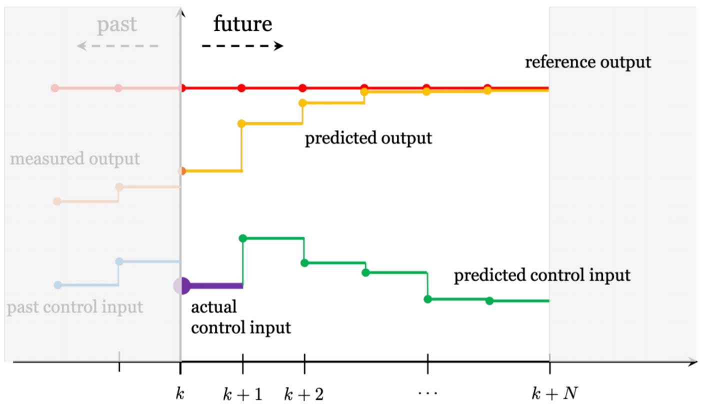
- Computation of the sets of inputs which minimizes the given objective function.



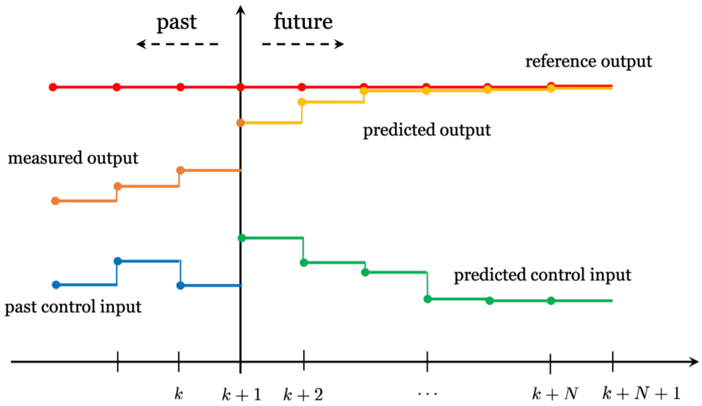
(a) MPC step 1



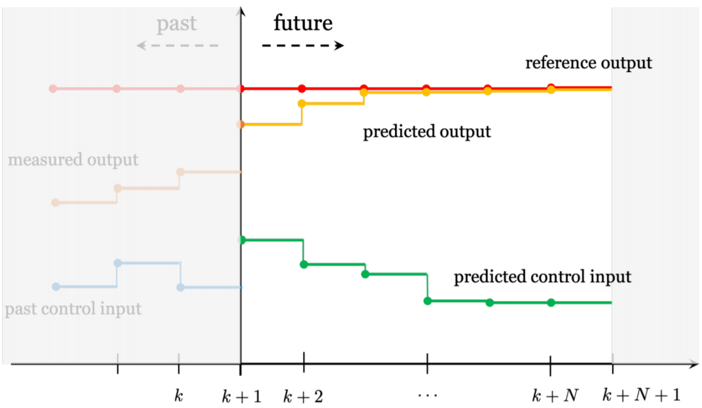
(b) MPC step 2



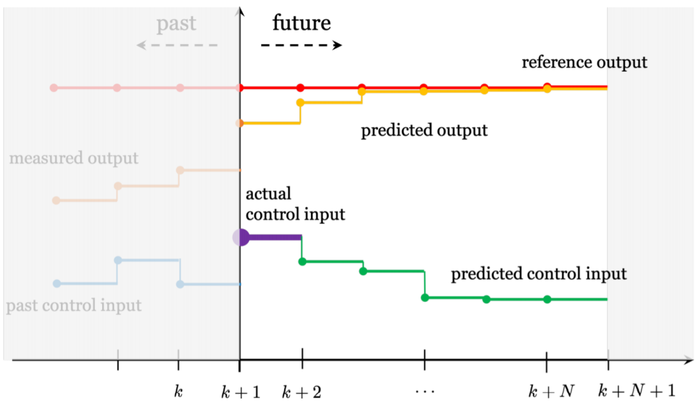
(c) MPC step 3



(d) MPC step 4



(e) MPC step 5



(f) MPC step 6

Figure 3.0: MPC iterations [43].

Theoretically, it is possible to adopt different models for these two tasks, but in practice they are often the same.

When choosing the model to implement, the main trade-off that arises is between *accuracy* and *computational cost*. A very detailed model (i.e. mathematically complex) can describe the system future behavior quite well when subjected to known inputs, and this brings effectiveness. On the other hand, this accuracy comes at the cost of time (and energy) necessary to perform calculations since many more operations need to be carried out. The computational effort is also influenced by the *time step* and *horizon length* which are later analyzed.

Specifically for this work, that deals with car motion planning, the main crossroads when choosing the vehicle model is between a kinematic model, that can be described as potentially less accurate but computationally light, and a dynamic one that brings high fidelity and larger computational burden.

The current motion planner is based on a car kinematic model: this reduces accuracy but also computational cost, and the risk of having singularities at low speed. This choice was motivated by the overall good performances that can be obtained in the analyzed scenarios, as proved in [44], since the car speed is rarely greater than $17\,m/s$, and the fact of having two parameters (l_f and l_r) makes adaptations to other vehicles with different sized wheelbases really easy.

In order to limit maneuvers aggressiveness and enhance comfort, the implemented model is inspired by [44] with considerations from [45] and further improvements. The first one is a classical bicycle model for cars with different frontal and rear axles length, which is proved to be accurate enough for AVs motion planning. The second one is modified to guarantee a better comfort, with the following considerations:

- Limiting the *Longitudinal Acceleration* allows to limit the longitudinal tire forces without having them explicitly present in the model;

- Limiting the *Longitudinal Jerk* improve comfort and consider actuators limitations;

- Limiting the *Lateral Acceleration* allows to limit the lateral tire forces without having them explicitly present in the model, and make maneuvers smoother (better comfort). If a small time step is considered, this can be done defining the path radius $R = L/\delta$ with $L = l_f + l_r$, that is the sum of car frontal and rear length. The lateral acceleration can then be approximated as $a_y = v_x^2/R$ or $a_y = v_x^2\delta/L$.

According to [46], this additional constraints decrease the chance of obtaining unfeasible solutions;

- Limiting the *Steering Change Rate* allows to improve comfort and consider actuators limitations.

On top of this, *Lateral Jerk* control has been introduced to reduce jerky feeling due to continuous variance of lateral acceleration. Besides, it further improves handling in case of slippery roads. This imposed to adopt steering acceleration as second input.

The vehicle is assumed to have a planar motion. The proposed car model is:

$$\dot{X} = v_x \cos(\Psi + \beta) \tag{3.1}$$

$$\dot{Y} = v_x \sin(\Psi + \beta) \tag{3.2}$$

$$\dot{\Psi} = \frac{v_x}{l_r} \sin(\beta) \tag{3.3}$$

$$\dot{v_x} = a_x \tag{3.4}$$

$$\dot{a_x} = j_x \tag{3.5}$$

$$\dot{a_y} = j_y \tag{3.6}$$

$$\dot{j_y} = ({a_x}^2 \, \delta + v_x \, j_x \, \delta + 2 \, v_x \, a_x \, \omega + \frac{1}{2} \, {v_x}^2 \, \Delta\omega) \, \frac{2}{l_f + l_r} \tag{3.7}$$

$$\dot{\delta} = \omega \tag{3.8}$$

$$\dot{\omega} = \Delta\omega \tag{3.9}$$

$$\beta = \tan^{-1}\left(\frac{l_r}{l_f + l_r} \, \tan(\delta)\right) \tag{3.10}$$

in which X, Y and Ψ represent the car absolute position on the map and its orientation. The longitudinal speed v_x and acceleration a_x are parallel to the longitudinal axis of the vehicle, while the lateral acceleration a_y is perpendicular to it. j_y represents the lateral jerk, δ is the steering and ω the steering change rate. The inputs to this model are the longitudinal jerk j_x and the steering acceleration $\Delta\omega$.

The effect of lateral acceleration control is evident when comparing planned trajectories in the same scenario with exactly the same other constraints (Figure 3.2): the maneuver shape and consecutive registered acceleration values are different. On top of that, the lateral jerk control makes its variation much smoother as emerges from the comparison in Figure 3.3. It turns out that this additional features can play a remarkable

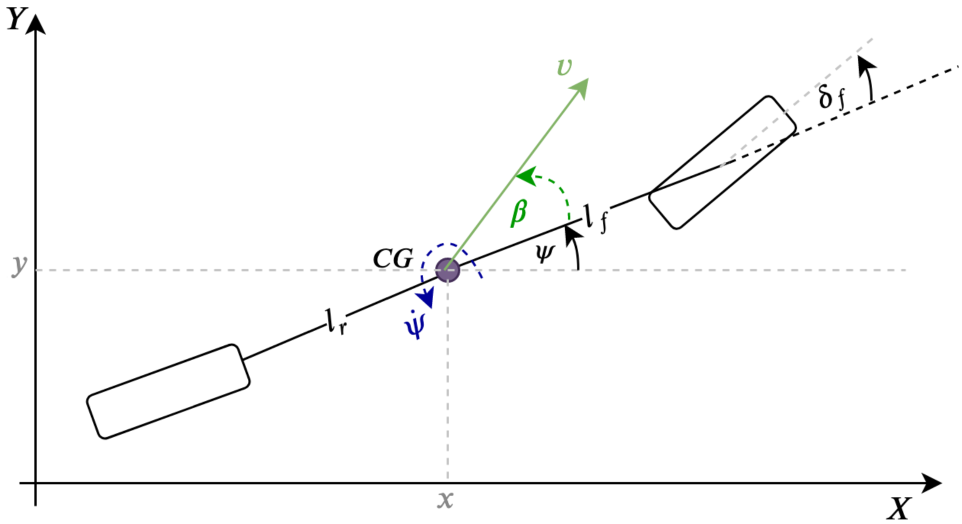


Figure 3.1: Reference system of a car kinematic bicycle model.

role in terms of comfort and road handling, as well as provide an additional customization for the passenger who can choose the driving style [47].

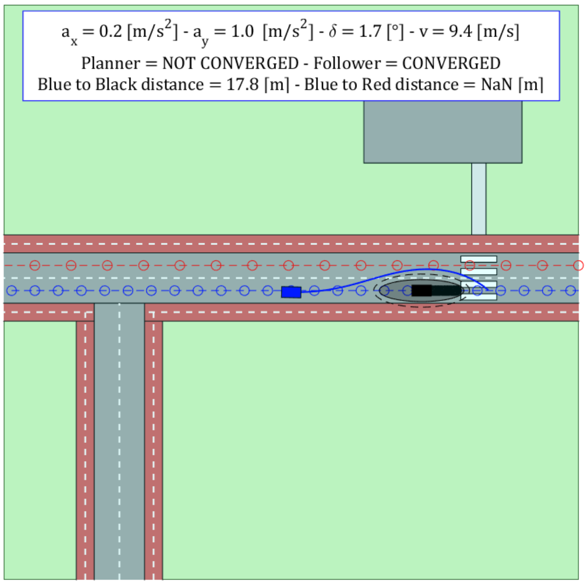
Without the control of lateral jerk and acceleration, but limiting longitudinal jerk and steering change rate, it would be possible to slightly mitigate the vehicle aggressive and jerky behavior, and this would come with a smaller computational effort. However, the current performances are great with reasonable solving time spans.

3.3 Constraints

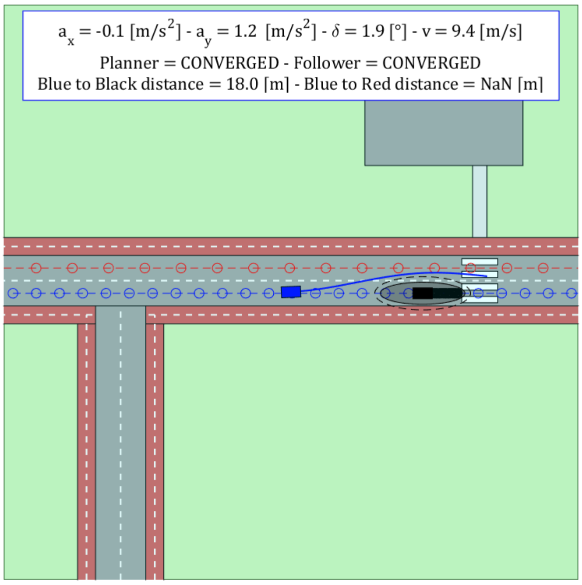
In reality all systems are subjected to constraints due to safety limitations, environmental regulations, consumer specifications and physical restrictions such as power and operating speed. It is possible to distinguish two main types of constraints:

- **Hard constraints** which cannot be violated for any reason. They define the solutions feasible domain;
- **Soft constraints** which can be violated (ideally for a short time). They are useful to deal with extreme initial conditions and guarantee continuity between consecutive states.

All the implemented constraints in the realized PP are of *hard* type.

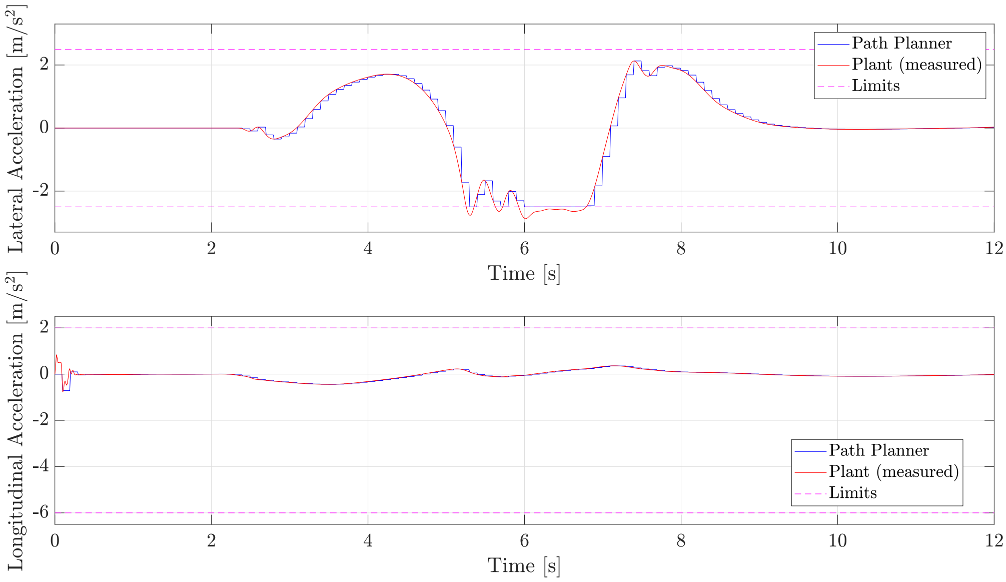


(a) Lateral acceleration neglected in both constraints and objective function: it reaches values around 6 m/s^2.

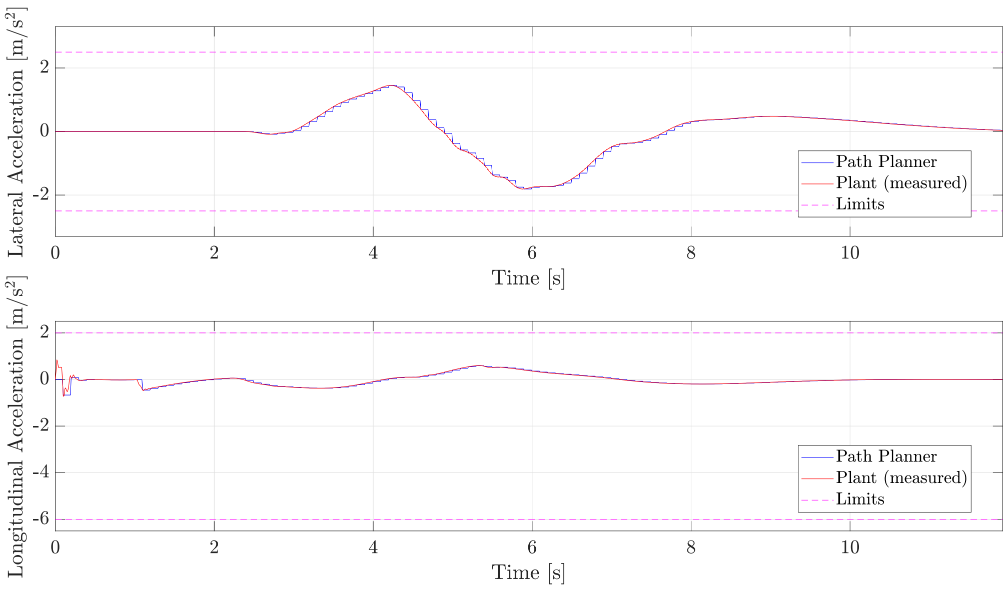


(b) Lateral acceleration considered in both constraints and objective function.

Figure 3.2: Overtaking paths generated with and without lateral acceleration control.



(a) Acceleration profiles without lateral jerk control.



(b) Acceleration profiles with lateral jerk control.

Figure 3.3: Influence of lateral jerk control on vehicle accelerations in Scenario 1 (Section 6.1).

Distinction can also be done between bounds for model inputs and states, and other constraints necessary to reproduce real-world limitations, like the road boundaries. Key design choices for both of them are represented by how the *vehicle ego* and *obstacles* are modeled when planning paths.

Vehicle ego

In order to consider the whole area occupied by the car (actually a bit more for safety reasons) and keep its representation simple, the vehicle ego is modeled as a set of four circles with radius R and placed along its longitudinal axis (see Figure 3.4).

$$R = \frac{\sqrt{(l_f + l_r)^2 + w^2}}{2} \tag{3.11}$$

in which l_f is the longitudinal distance between the COG and the front axle, l_r the one respect to the rear axle and w is the car width.

A totally different definition of R can be used to improve safety or other reasons.

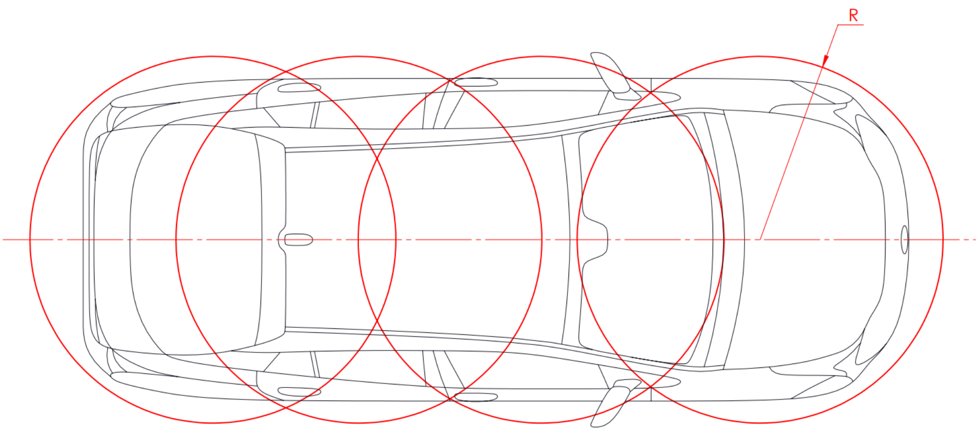


Figure 3.4: Car ego represented as a set of four circles.

Obstacles modeling

Following up on the work done in [28], moving obstacles are modeled as ellipses whose semi-axis dimensions (a and b) vary, respect to the base value (obstacle dimensions), according to different factors:

1. a increases with increasing velocity of the controlled car. This is inspired by the safety distance concept present in road regulations. In Figure 3.5 it is showed with two different velocities (black ellipse and green ellipse);

2. both a and b are increased (like an offset) of the absolute value of estimated maximum error between Path Planner and Path Follower (explained below). This is possible thanks to an additional feedback between the two;

3. a and b become equal when the relative angle between the controlled car and the obstacle is 90° (the obstacle becomes a circle).

Regarding the second contribute, the mentioned error can be of two types:

1. The maximum distance between the paths, for example comparing waypoints one by one;

2. The maximum variance between the controlled agent to obstacles distances predicted by Path Planner and Path Follower along the whole horizon.

The current set-up adopts the second type. A successful application based on something similar to the first type of error is outlined in [48].

3.3.1 Limits on Inputs and States

Inputs

$$-4 \leq j_x \leq 1 \ \frac{m}{s^3} \tag{3.12}$$

$$0.5 \leq \Delta\omega \leq 0.5 \ \frac{rad}{s^2} \tag{3.13}$$

They respectively represent longitudinal jerk (or variation of longitudinal acceleration) and steering acceleration. The main trade-off when deciding these values is between vehicle promptness (better with large limits), which can also influence safety especially in case of evasive maneuvers (see Appendix B), and actuators limitations. Values have been decided considering the work reported in [49].

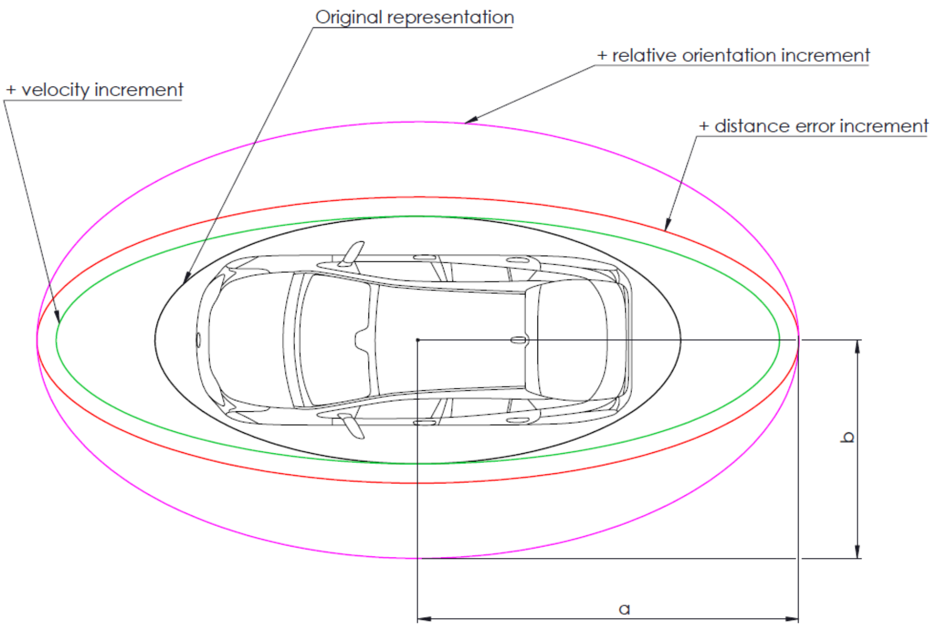


Figure 3.5: Influence of distinct increments on moving obstacles representation.

Position and Orientation

$$x_{min} \leq x \leq x_{max} \tag{3.14}$$
$$y_{min} \leq y \leq x_{max} \tag{3.15}$$
$$\psi_{min} \leq \psi \leq \psi_{max} \tag{3.16}$$

Together they represent the area where the vehicle can move and its possible orientations. They are valid for both COG and Ego (see Section 5.2) and can theoretically be infinite.

Velocity

$$0 \leq v \leq 50 \ \frac{m}{s} \tag{3.17}$$

The lower bound actually needs to be set a bit greater than zero to avoid numerical problems with the high-fidelity plant (Section 5.1), and the upper one only needs to meet traffic laws.

Longitudinal Acceleration

$$-6 \leq a_x \leq 2 \ \frac{m}{s^2} \tag{3.18}$$

These bounds also aims to ensure handling, safety (emergency braking), passengers' comfort and take actuators limits into account. Values have been decided considering the experiments results reported in [49].

Lateral Acceleration

$$-2.5 \leq a_y \leq 2.5 \ \frac{m}{s^2} \tag{3.19}$$

These bounds ensure handling with good road conditions, which means small deviations between the planned path and the executed one. Moreover, the passengers' comfort is expected to improve.

Lateral Jerk

$$-3 \leq j_y \leq 3 \ \frac{m}{s^3} \tag{3.20}$$

These bounds ensure limited lateral acceleration variation and are expected to improve comfort. These values are a compromise between comfort indications and maneuvers feasibility: on the one hand, in case of relatively high speed the vehicle might not have enough time to change lane once the obstacle is detected for example; on the other hand a large limit would lead to jerky behavior.

Steering

$$-0.52 \leq \delta \leq 0.52 \ rad \tag{3.21}$$

These bounds also aims to take actuators limits into account.

Steering Change Rate

$$-0.5 \leq \omega \leq 0.5 \ \frac{rad}{s} \tag{3.22}$$

These bounds also aims to take actuators limits into account, which could theoretically be wider than the specified ones. As before, values have been decided considering the work reported in [49].

3.3.2 Operation Constraints

These constraints reproduces limitations the vehicle faces when operating in real life. Both of them involve the concept of *vehicle ego* and are inspired to the work done in [27].

Road boundaries

In each scenario a certain reference trajectory is provided, which does not take into account moving obstacles and is assumed to be exactly in the middle of the lane. Therefore, this constraints allow asymmetric lateral deviations, as the car can run on the other lane to overtake or avoid obstacles (when possible), while keeping COG and ego within road boundaries.

At each time step the reference trajectory closest point $(x^P(\vartheta), y^P(\vartheta)$ is found and the minimum distance to the momentary vehicle position $\mathbf{z}_k = (x_k, y_k)$ is determined (ϑ is the path parameter). Following the demonstration presented in [50], it is possible to define it, that is to define the *contour error*. Since the path is continuously differentiable and bounded, the tangential and normal vectors as function of the path parameter ϑ are

$$\mathbf{t} = \begin{bmatrix} \dfrac{\partial x^P(s)}{\partial s} \\ \\ \dfrac{\partial y^P(s)}{\partial s} \end{bmatrix}, \mathbf{n} = \begin{bmatrix} -\dfrac{\partial y^P(s)}{\partial s} \\ \\ \dfrac{\partial x^P(s)}{\partial s} \end{bmatrix} \tag{3.23}$$

The controlled vehicle should follow a given route with sufficiently low deviation from the given reference. It is then possible to assume that

$$\Delta\vartheta \approx \Delta s = v\Delta t \tag{3.24}$$

in which s is the vehicle actual path. The same assumptions made in the mentioned reference hold. It is therefore possible to define the next point on the reference path

$$\vartheta_{k+1} = \vartheta_k + v_k \Delta t_k \tag{3.25}$$

where $v_k \Delta t_k$ describes the approximated progress at time step k.

The contouring error can be expressed as:

$$e_{con}(\mathbf{z}_k, \vartheta_k) = \frac{\mathbf{n}_k^T}{||\mathbf{n}_k||} \begin{bmatrix} x_k - x^P(\vartheta_k) \\ y_k - y^P(\vartheta_k) \end{bmatrix} \tag{3.26}$$

This lateral error has to be lower than a half of the road width reduced by a half of the car width (as the whole car is expected to stay on the road):

$$-\frac{3}{2}l_w + \frac{w}{2} \leq e_c \leq \frac{l_w}{2} - \frac{w}{2} \tag{3.27}$$

in which w is the car width and l_w is the lane width. The error is positive when the car is shifted to the right respect to the middle of the lane, negative otherwise (see Figure 3.6). This constraint is valid for all ego circles.

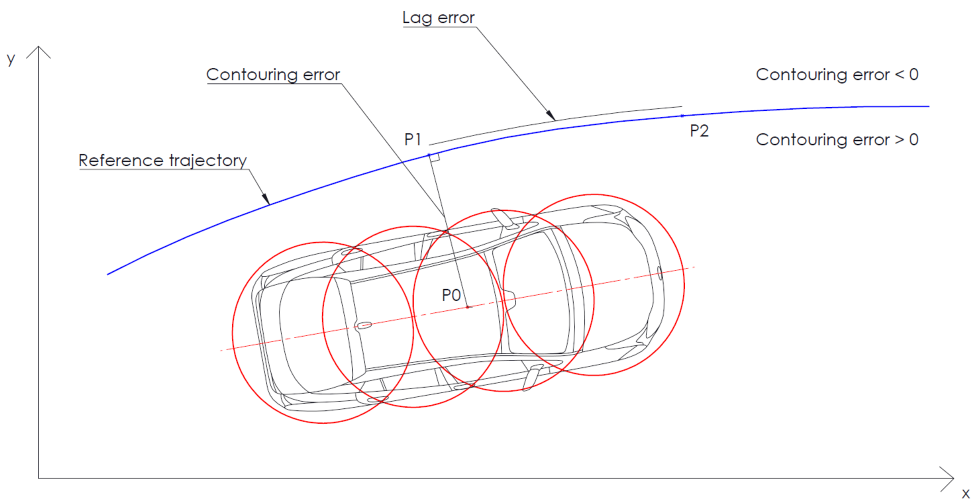


Figure 3.6: Representation of a vehicle position associated to positive contour error (the reference trajectory is in the middle of the lane). P0 (or $\mathbf{z}_k$)is the momentary car position, P1 (or $\mathbf{z}^P(\vartheta^P)$)is the closest point on the reference trajectory while P2 (or $\mathbf{z}^P(\vartheta_k)$) is where the vehicle should be (but it is not due to both contouring and lag error).

Collision Avoidance

This constraint, together with the cost taken into account in the objective function, is necessary to prevent agents collisions. The approach is the same adopted in [28]: for each moving obstacle j and prediction step k, we impose that each ego circle does not intersect with the elliptical area

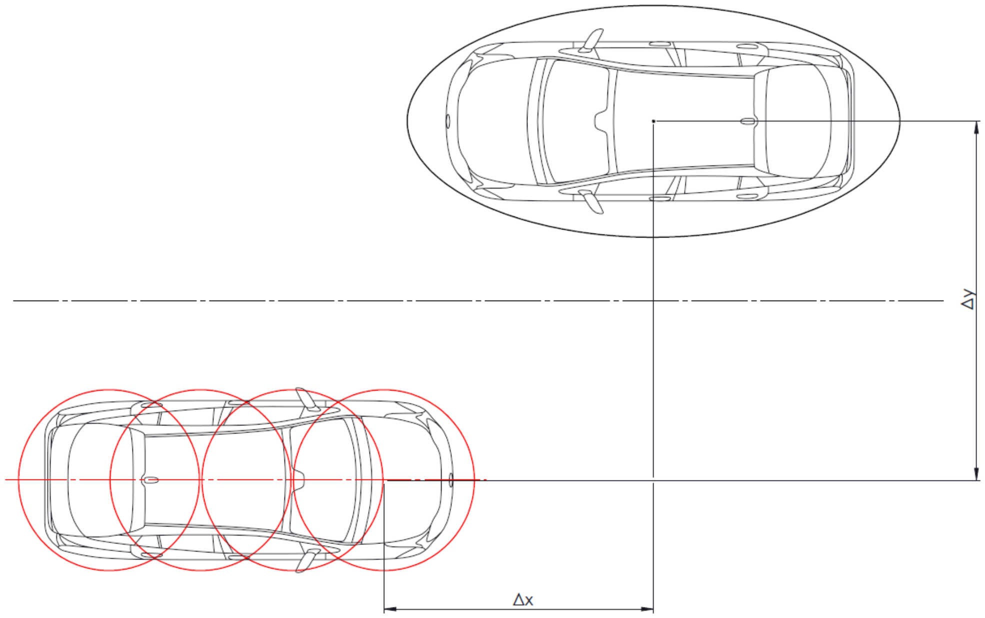


Figure 3.7: Considered distances for one of the ego circles in collision avoidance constraint.

representing the obstacle.

$$\begin{bmatrix} \Delta x_j^k \\ \Delta y_j^k \end{bmatrix}^T R(\varphi)^T \begin{bmatrix} \frac{1}{a^2} & 0 \\ 0 & \frac{1}{b^2} \end{bmatrix} R(\varphi) \begin{bmatrix} \Delta x_k^j \\ \Delta y_j^k \end{bmatrix} > 1 \tag{3.28}$$

in which r_{disc}, a and b are the same defined before, $\Delta x_k{}^j$ and $\Delta y_k{}^j$ are the separated components of the agents distances, while $R(\varphi)$ is the rotation matrix defined as follows:

$$R(\varphi) = \begin{bmatrix} cos(\varphi) & -sin(\varphi) \\ sin(\varphi) & cos(\varphi) \end{bmatrix} \tag{3.29}$$

3.4 Performance Index

The control action is found through optimization (minimization) of a performance index, also called *objective function* or *cost function*. This performance index decreases as the reference tracking error and the control action reach the lowest values possible. Specifically for this Path Planner, the NMPC controller returns the path that minimizes the implemented cost function while respecting the given constraints and system dynamics.

The general performance index expression is the sum of all available quantities that can somehow influence the result of every horizon time step:

$$J_{tot} = \sum_{k=0}^{N-1} J_{k,j} \;\; with \; j \in \; \mathbb{M} \tag{3.30}$$

in which N is the horizon length and M is the set of considered quantities, which are one by one analyzed below.

Contouring Error

The considered contouring error is the same explained before. A large cost coefficient multiplying this term force it too be small and the car proceeds as close as possible to the original trajectory and obstacles, if they cause deviations. Its effect is clearly visible in Figure 3.11.

$$J_{contour} = e_{con_{cc}} \, {e_c}^2 \tag{3.31}$$

Lag Error

The lag corresponds to the delay the vehicle has accumulated respect to the given initial trajectory (in which moving obstacles are not present). All the consideration introduced for the road boundaries are still valid, hence the lag error expression is:

$$e_l(\mathbf{z}_k, \vartheta_k) \; = \; \frac{\mathbf{t}_k^T}{||\mathbf{t}_k||} \begin{bmatrix} x_k \; - \; x^P(\vartheta_k) \\ y_k \; - \; y^P(\vartheta_k) \end{bmatrix} \tag{3.32}$$

and it is visible in Figure 3.6. Its value increases as the car has to decelerate because the obstacles met can not be safely avoided, for instance due to road regulations or lack of space. A higher cost coefficient push the car to keep up with the specified trajectory (that is being at given points at defined moments).

$$J_{lag} = e_{lag_{cc}} \, {e_l}^2 \tag{3.33}$$

Velocity Error

The velocity error is the difference between the actual car velocity and the specified for the tested scenario.

$$e_v \; = \; |v_{ref} - v_{actual}| \tag{3.34}$$

The greater the corresponding cost coefficient and the more the car tends to run at that speed. It can, for example, make the difference when deciding whether to overtake an obstacle or not.

$$J_v = v_{cc}\, e_v^2 \tag{3.35}$$

Longitudinal Acceleration

When safety allows it, the lower the longitudinal acceleration and the better the comfort. Moreover, this leads to lower fuel consumption.

$$J_{a_x} = a_{x_{cc}}\, {a_x}^2 \tag{3.36}$$

Lateral Acceleration

Lateral acceleration is positively weighted in order to facilitate road holding and improve comfort.

$$J_{a_y} = a_{y_{cc}}\, {a_y}^2 \tag{3.37}$$

Agents Distance

Mutual distances between the controlled vehicle and the other moving obstacles, which are the other agents in the problem, are assumed to be known when they are within the detection range, and for safety reasons this distances cannot become too small (see Section 3.3.2). Including a repulsive field dependent on agents distances prevents sudden changes of direction and increase clearances. In other words, the resultant path is smoother [28]. In [29] this technique is used to plan trajectories and perform emergency collision avoidance maneuvers. Different mathematical function has been tested (see Figure 3.9) but the implemented one (called "modified hyperbole")is reported below and showed in Figure 3.10 in 3D.

Defining as *agent i* the controlled agent and *agent j* another one close enough, the equations are:

$$\vartheta_{ij} = \tan^{-1}\left(\frac{y_j - y_i}{x_j - x_i}\right) - \psi_j \tag{3.38}$$

$$\text{min dist} = \frac{(a_j\, b_j)}{\sqrt{(b_j\, cos(\vartheta_{ij}))^2 + (a_j\, sin(\vartheta_{ij}))^2}} \tag{3.39}$$

$$\text{dist metric} = \frac{1}{2\,(\min(0, \text{min dist} - \text{ego min dist})) + \varepsilon} \tag{3.40}$$

$$J_{ad_{ij}} = \text{dist metric}_{cc}\, \text{dist metric} \tag{3.41}$$

in which Ψ_j is the agent j orientation angle, ϑ_{ij} is the agents relative angle, a_j and b_j are the agent j semi axis dimensions (see Figure 3.5), *ego min dist* is the shortest distance between agent j and the controlled vehicle ego circles (see Figure 3.4), dist metric$_{cc}$ is the cost coefficient, while ε is necessary for numerical stability (finite value). Its meaning is represented in Figure 3.8. As showed in Figure 3.9, the proposed potential field:

1. Is not symmetrical around the car COG (Center of Gravity) except in case of perfect perpendicularity between agents;
2. Is not properly smooth (modified exponential can be defined smoother than hyperbole) because it makes car control easier. If not, costs become unpredictable since what matters is their values when compared each other.

Its importance is exhaustively motivated in [28].

Total Cost

Minor weights have been assigned to some of the other states, such as longitudinal jerk, steering acceleration, steering change rate and lateral jerk. The total cost minimized is the sum of all the above mentioned ones:

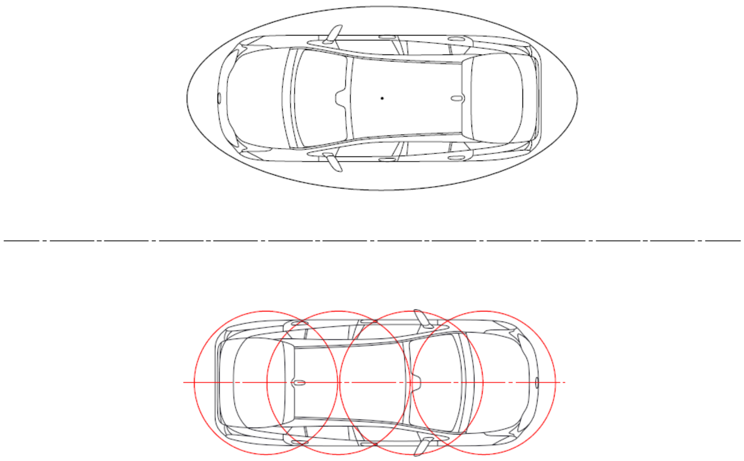
$$J_{tot} = J_{contour} + J_{lag} + J_v + J_{a_x} + J_{a_y} + \sum J_{ad_{ij}} + J_{other} \tag{3.42}$$

3.4.1 Dynamic Cost Coefficients

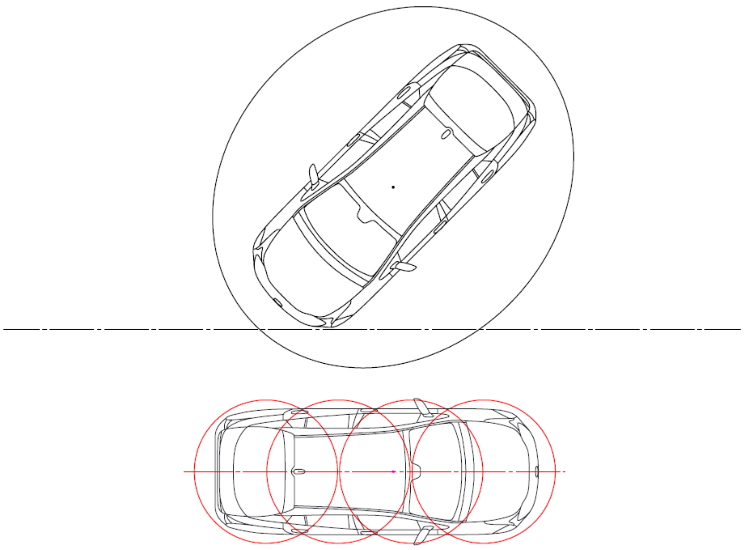
AVs are usually equipped with an additional module called *Behavioral Layer*, which is responsible for selecting an appropriate driving behavior, that is the maneuver to execute and how, at any moment. The outcome is based on the perceived behavior of other traffic agents, navigation goal, road conditions, and signals from infrastructure. An example of Behavioral Layer employing a Markov Decision Process (MDP) is described in [51].

In the realized simulation framework this module is not present because it goes beyond its main scope, but the same process is reproduced so that this Local Motion Planner can still benefit of enhanced performances. Basically, the previously described cost coefficients of the PP cost function can be changed in real time according to the situation the car faces.

In other words, here the cost coefficients and constraints influencing how maneuvers are executed are given for each situation. For example, in



(a)



(b)

Figure 3.8: Variation of obstacle modeled size with different perceived relative angles.

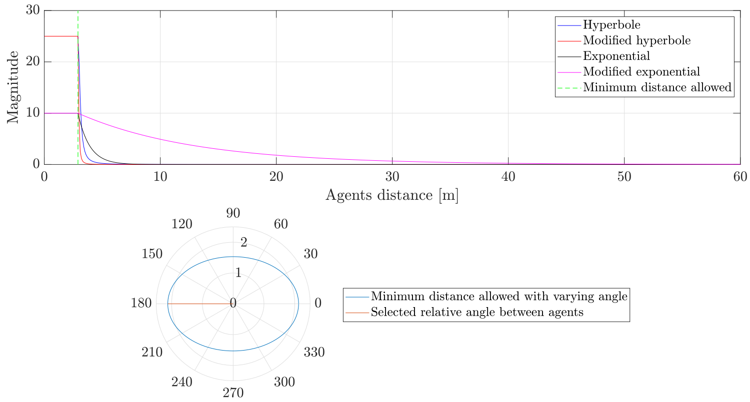


Figure 3.9: Comparison of different agents distance cost functions.

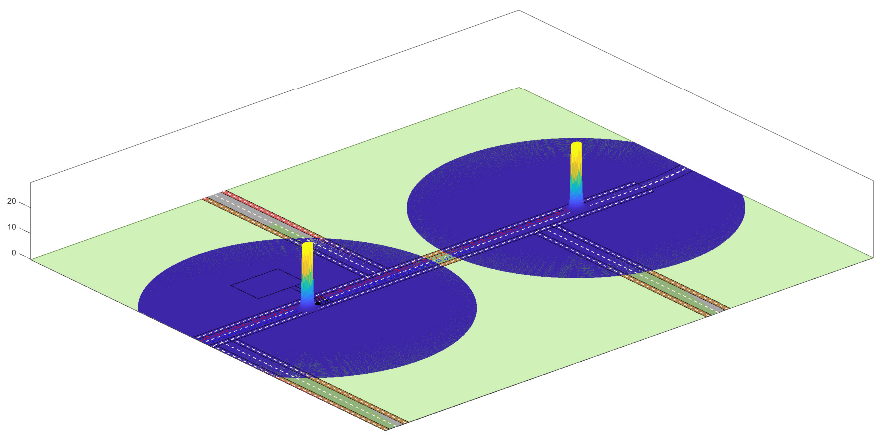


Figure 3.10: 3D representation of repulsive fields at 36 km/h: blue when far, yellow when nearby. The current positions correspond to initial agents positions and the diameter of the detection range.

Section 6.1 the controlled vehicle is expected to overtake, therefore proper cost coefficients and constraints are set.

In order to point out usefulness and effectiveness of this approach, different scenario with varying trajectories and agents behaviors are tested, while their optimal weights are stored and provided in form of parameters to the NLP.

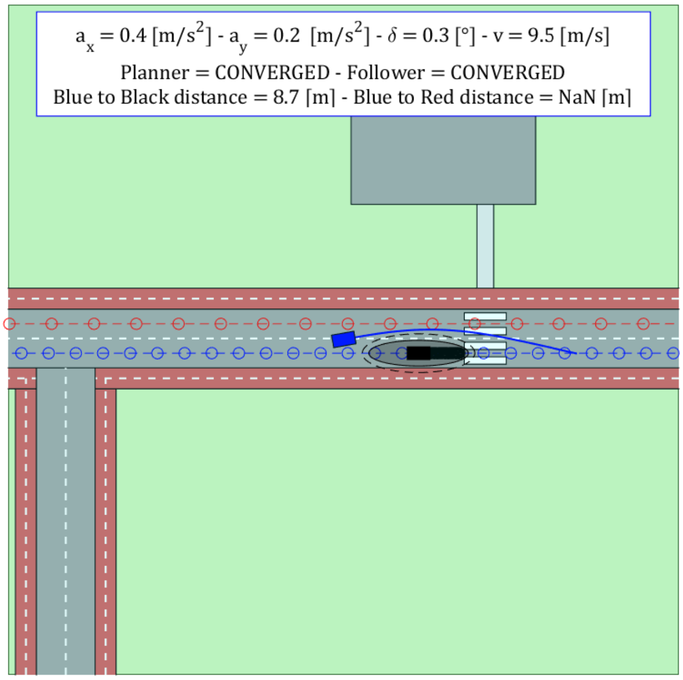
This additional feature is motivated by the fact that choosing proper cost coefficients makes the car move as desired, like a real driver would do, and speeds up the optimization process. In Figure 3.11 the same scenario is tested with different cost coefficients: it is visible how the shape changes according to that. Clearly other maneuvers features (trajectories, velocity, acceleration etc.) are also influenced.

3.5 Optimization

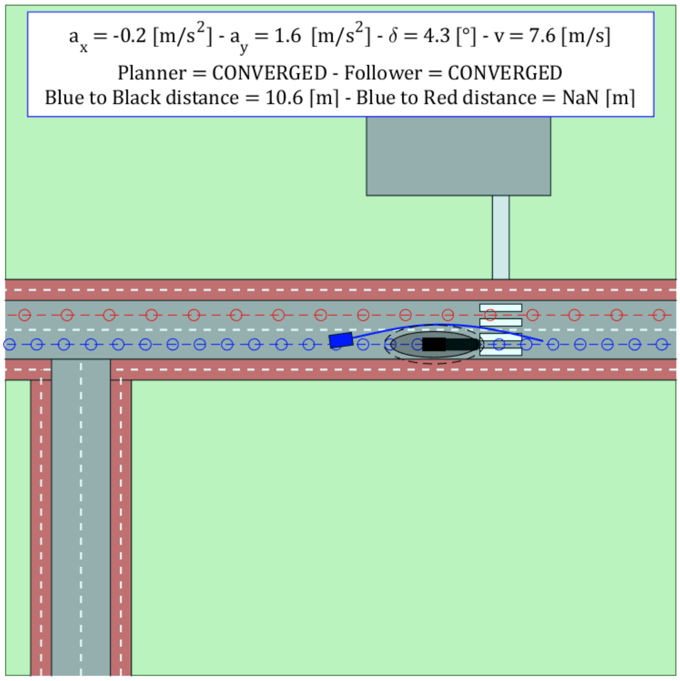
In this section the optimization process carried out to solve the continuous Optimal Control Problem (OCP) is described in its main features. A brief description of the employed commercial software is also provided.

The optimization problem components outlined in the previous sections are expressed in continuous form but computers work with discrete values. Therefore, the whole framework needs to be discretized and to obtain the so called Non-Linear Programming (NLP). Different methods are available as explained in [52]. The here adopted solver is based on a *Direct Multiple Shooting* method due to the special problem structure returned [53, 54].

When discretizing, a time step has to be defined. This arise a compromise between accuracy, which is possible with a small time step as the dynamics can be better explained, and computational effort. The chosen time step for the Path Planner is $dt = 0.1\,s$ since it seems able to capture the dynamics well enough (if a PF is present). Actually, the overall computational effort required depends on the horizon length as well, which is the other key setting: longer horizon means longer predictions but also more operations. A time horizon of $3\,s$ seems to be acceptable considering that corresponds to $30\,m$ at $10\,m/s$, which is comparable with the range of some sensors mounted on an Autonomous Vehicle and the emergency braking explained in Appendix B. Most important of all, a long horizon allows to start designing paths early enough to make them executable and smooth. The adopted horizon length is $N = 30\,steps$. Figure 3.12 contains PP performances with different settings (dt and N) and shows how the chosen settings optimize convergence rate or even allow to complete the whole maneuver.

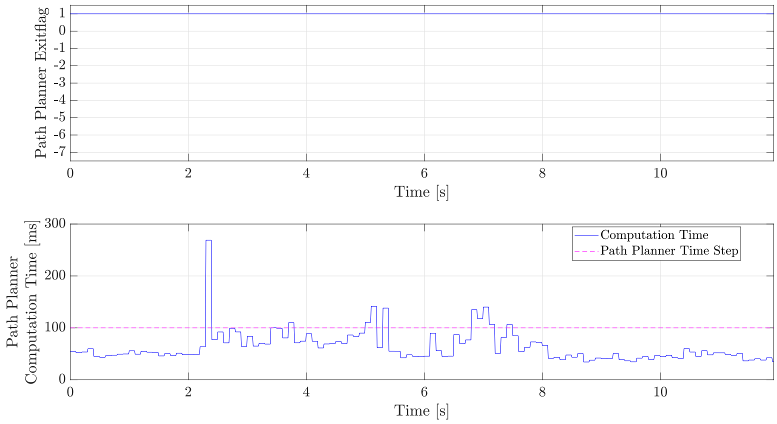


(a) Cost coefficients for a wide maneuver.

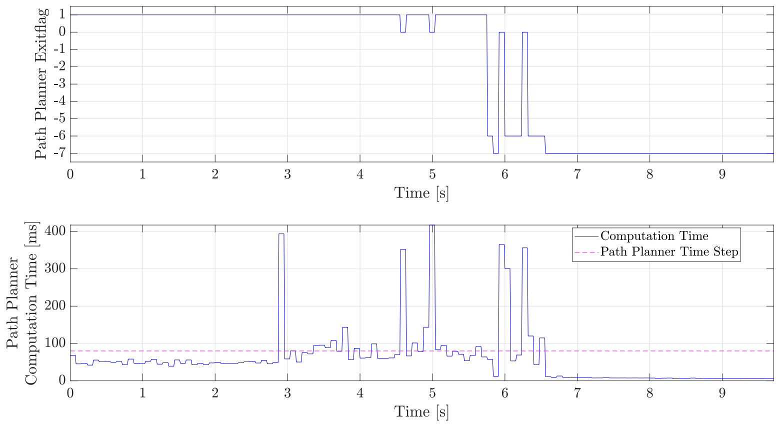


(b) Cost coefficients for a tight maneuver.

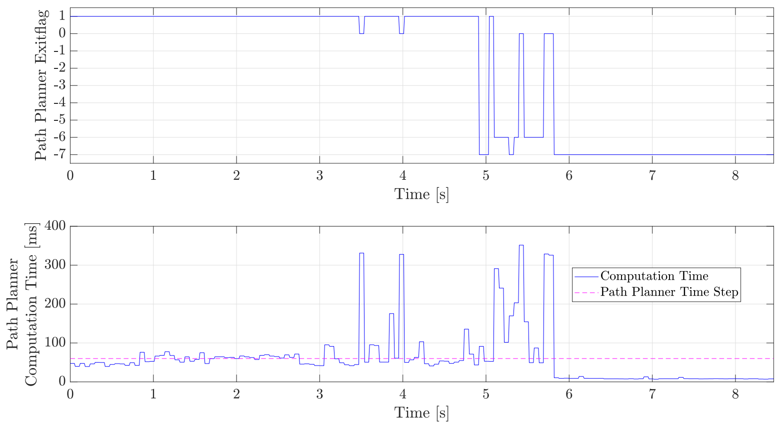
Figure 3.11: Overtaking path generated with different cost coefficients shows different shape and starting point.



(a) Path Planner performances with $dt = 0.1\ s$.



(b) Path Planner performances with $dt = 0.08\ s$.



(c) Path Planner performances with $dt = 0.06\ s$.

Figure 3.12: Scenario 1 optimization processes results with different time steps.

Once discretized, the NLP looks like this:

$$
\begin{aligned}
\min_{u} \quad & \sum_{k=0}^{N-1} J_{tot_k} \\
\text{s.t. :} \quad & x_{k+1} = f(x_k, u_k) \qquad k = 0, ..., N-1 \\
& u_{min} \leq u_k \leq u_{max} \\
& x_{min} \leq x_k \leq x_{max} \\
& x_0 = x_{current} \\
& \boldsymbol{u} = [u_0, ..., u_{N-1}] \\
& \boldsymbol{x} = [x_0, ..., x_{N-1}]
\end{aligned}
\tag{3.43}
$$

An optimal value which minimizes the given objective function for the whole horizon must be found within given constraints. If that optimum is found, the first set of optimal control input are applied to the system. However, when this does not happen (for reasons later explained) other actions are taken (see Section 2.2.2).

From a general perspective, in case of inequality/equality constrained optimization problems, the Karush-Kuhn-Tucker (KKT) condition gives necessary and sufficient conditions for an optimum. It states that a minimum or maximum for the treated problem exist if λ and μ are such that:

$$
\begin{aligned}
\nabla f(x) \,+\, \nabla g(x)\mu \,+\, \nabla h(x)\lambda \,&=\, 0 \\
\mu^T g(x) \,&=\, 0 \\
\mu \,&\geq\, 0 \\
h(x) \,&=\, 0 \\
g(x) \,&\leq\, 0
\end{aligned}
\tag{3.44}
$$

However, given the reality complexity, planning is more like a nonlinear, non-convex problem for which there is no general analytic solution. In this case, approximations of the optimum can only be found while an important feature of the chosen algorithm is whether it will converge to the optimum value or not, and different measures of convergence are available. When an algorithm converges, it is still necessary to decide at what point the estimate is "good enough". Just to give an idea, a revised expression of

KKT condition can be used as *stopping criteria*:

$$\begin{aligned} ||\nabla f(x_k) + \nabla g(x_k)\mu_k + \nabla h(x_k)\lambda_k||_2 &\leq \varepsilon_{KT,1} \\ |\mu_k^T g(x_k)| &\leq \varepsilon_{KT,2} \\ \mu_k &\geq -\varepsilon_{KT,3} \\ ||h(x_k)||_2 &\leq \varepsilon_{KT,4} \\ g(x_k) &\leq \varepsilon_{KT,5} \end{aligned} \tag{3.45}$$

where $\varepsilon_{KT,i} > 0$.

When neither the gradient nor the Jacobian are available, it is necessary to opt for a more heuristic stopping criteria, such as function variations between two consecutive steps or a maximum number of iterations. A detailed explanation of the implemented algorithm and possible termination strategies is omitted since this job is mostly executed with the default settings of the solver.

Since the stability is not ensured by the MPC law itself, the problem is usually equipped with a *terminal cost* and a *terminal constraint set*. By doing this, closed loop stability is reached. A detailed explanation of this topic goes beyond the scope of this work but can be found in [55].

The NLP is solved with ForcesPro ([53, 54]) that is a powerful commercial solver. By default, it uses a stable and robust algorithm for most problems called Primal-Dual Interior Point Method. It approximates Hessian and uses adaptive barrier rules in order to reduce computational time and make it comparable to the QP one while keeping non-linearity. Solver performances for Scenario 1 are showed in Figure 3.12. Roughly, the solver should be able to return a solution in less than the MPC time step ($0.1\,s$ in this case) which is represented by the magenta line in the same figure. As it is possible to note, in case of non-convergence (exitflag different from 1) that target is not reached. However, proper settings (such as the maximum number of iterations) and processor (currently i7-9750H) can reduce the time lost. Again, when an optimum is not found other measures are taken ensure safety.

3.6 Receding Horizon Principle

Model Predictive Control uses the receding horizon principle, which means that at the end of the optimization process only the first set of inputs is implemented. After that, the whole horizon is shifted of one sample and the loop starts again with new information coming from measurements (see Figure 3.0). This approach allows to effectively deal with one of the major LMP limitations (Section 1.2.5).

Chapter 4

Proposed Path Follower

In this chapter the employed Path Follower is described in its features and tasks. Since it is made of two different PFs working in parallel (MPC and PID-based), a detailed description is provided for each of them. This component basically *"connects the dots"*, that is the way-points coming from the PP, and its key role is demonstrated by showing the overall performances when left out. A comparison between the two is also reported. In Figure 2.1 the black box represents this component.

In other words, a Path Follower "adjusts" the inputs coming from the PP (see Figure 4.1) while keeping the actual vehicle state into account, like with a feedback loop. PP inputs are indeed discrete, sent at relatively low frequency and generated with a simplified model of the car. Although performances look good, the introduction of this component makes them much better.

Thanks to the presence of a PF, it has been possible to keep the PP time step relatively long, which translates into a long horizon (time and distance) with a given number of steps, and this is quite helpful when designing maneuvers. As a consequence of that, its removal or by-pass without structural modifications would probably cause the loss of vehicle control.

Precisely, the realized PF is made of two different PFs based on different approaches which can either work in parallel (to improve robustness) or singularly. Their pros and cons are evaluated in the coming sections, as well as their performance comparison and contribute to the whole planning task.

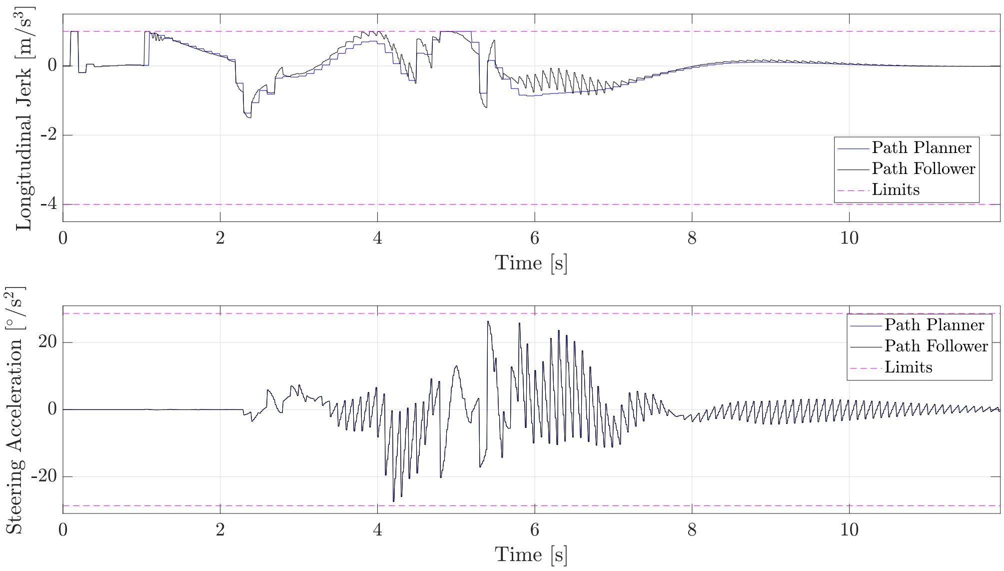


Figure 4.1: Inputs comparison between Path Planner and Path Follower (MPC-based).

4.1 MPC-based Path Follower

This PF is based on MPC, like the PP, and interacts with it but works with a different vehicle model and at a higher frequency to be more accurate. The process represented in Figure 2.1 simplifies and shows what this component does multiple times (brown loop and green loop) in a single PP time step (magenta loop): it recomputes inputs to make sure the vehicle follows the PP paths (which are like references). All the details are explained in the coming sections following the same arrangement used for the PP description. Thanks to this component:

1. The car dynamics is considered before sending inputs;
2. The Local Motion Planner results more reliable on the whole;
3. The overall processing time is not heavily affected or even accelerated.

Figure 4.1 shows the difference between the inputs computed by PP and this MPC PF for one of the tested scenarios (Section 6.1).

4.1.1 Prediction Model

In this case, the dynamic model proposed in [44] is implemented with two main differences:

- Tire forces are calculated with the Dugoff model and then taken into account;

- Inputs are *Longitudinal Jerk* and *Steering Acceleration*.

$$\dot{X} = v_x \cos(\Psi + \beta) \tag{4.1}$$
$$\dot{Y} = v_x \sin(\Psi + \beta) \tag{4.2}$$
$$\dot{\Psi} = v_z \tag{4.3}$$
$$\dot{v_x} = v_x\, v_y + a_x \tag{4.4}$$
$$\dot{v_y} = -v_z\, v_x + \frac{2}{m}\,(F_{l,f} \cos(\delta) + F_{l,r}) \tag{4.5}$$
$$\dot{v_z} = \frac{2}{I_z}\,(l_f\, F_{l,f} - l_r\, F_{l,r}) \tag{4.6}$$
$$\dot{a_x} = j_x \tag{4.7}$$
$$\dot{\delta} = \omega \tag{4.8}$$
$$\dot{\omega} = \Delta\omega \tag{4.9}$$
$$\beta = \tan^{-1}\left(\frac{l_r}{l_f + l_r}\, \tan(\delta)\right) \tag{4.10}$$

where X, Y and Ψ represent the vehicle position and orientation, v_x and v_y denote the longitudinal and lateral speeds in the body frame respectively, v_z is the yaw rate, δ is the steering while ω represents the steering change rate. As for the PP, the longitudinal jerk j_x and the steering acceleration $\Delta\omega$ are the inputs to this model. m and I_z denote the vehicle mass and yaw inertia respectively, while $F_{l,f}$ and $F_{l,r}$ represent the lateral tire forces in coordinate frames aligned with the wheels. Longitudinal tire forces have been omitted. The reference system is still the one shown in Figure 3.1.

Tire forces are explained with the Dugoff model [56], which is classified as a tire physical model: it assumes that the contact region between tire and road is realized through predefined mechanical elements (spring-mass-dampers). The main reason behind this choice is to keep the road friction coefficient as single parameter when taking modified road conditions into account. The computed tire forces are based on the last measured physical quantities therefore they are considered constant for the whole MPC horizon. This choice might lack of descriptiveness but makes implementation easier.

$$\mu = \mu_0 \left(1 - e_r V_x \sqrt{\kappa^2 + \tan^2 \alpha}\right) \tag{4.11}$$

$$\lambda = \frac{\mu F_z (1-\kappa)}{2\sqrt{(K_{xk}\kappa)^2 + (K_{y\alpha} \tan \alpha)^2}} \tag{4.12}$$

$$f(\lambda) = \begin{cases} \lambda\,(2-\lambda) & \text{if} \lambda < 1 \\ 1 & \text{if} \lambda \geq 1 \end{cases} \tag{4.13}$$

$$F_x = \frac{K_{x\kappa}\,\kappa}{1-\kappa} f(\lambda) \tag{4.14}$$

$$F_l = \frac{K_{y\alpha}\,\tan \alpha}{1-\kappa} f(\lambda) \tag{4.15}$$

in which μ_0 represents the peak friction coefficient, e_r is the friction reduction coefficient, V_x is the longitudinal velocity, $K_{x\kappa}$ is the longitudinal slip stiffness, $K_{y\alpha}$ is the cornering stiffness, κ is the wheel slip, a is the slip angle and F_z is the normal force. Both $F_{c,f}$ and $F_{c,r}$ are represented by F_y but with different values. The slip angles are computed as reported in Equation 4.16 (front) and Equation 4.17 (rear).

$$\alpha_f = -\tan^{-1} \left(\frac{(v_y + l_f v_z)\cos(\delta) - v_x \sin(\delta)}{(v_y + l_f v_z)\sin(\delta) + v_x \cos(\delta)} \right) \tag{4.16}$$

$$\alpha_r = -\tan^{-1} \left(\frac{v_y - l_r v_z}{v_x} \right) \tag{4.17}$$

A comparison between the proposed saturated model and the linear one (Equation 4.18) is shown in Figure 4.2. In practice, unrealistically large values could arise and so do numerical problems, other than the impossibility of considering the actual friction. However, both are available in the simulation environment and, as long as slip angles are small, computed forces are comparable. In Section 6.1.2 the key contribute of the Dugoff model is demonstrated.

$$F_{l,i} = -K_{y\alpha}\,\alpha_i \tag{4.18}$$

4.1.2 Constraints

Limits on Inputs and States

In order to reproduce the real limitations, improve comfort and maximize handling, both inputs and states are bounded. The same limits hold for the states in common between PP and PF:

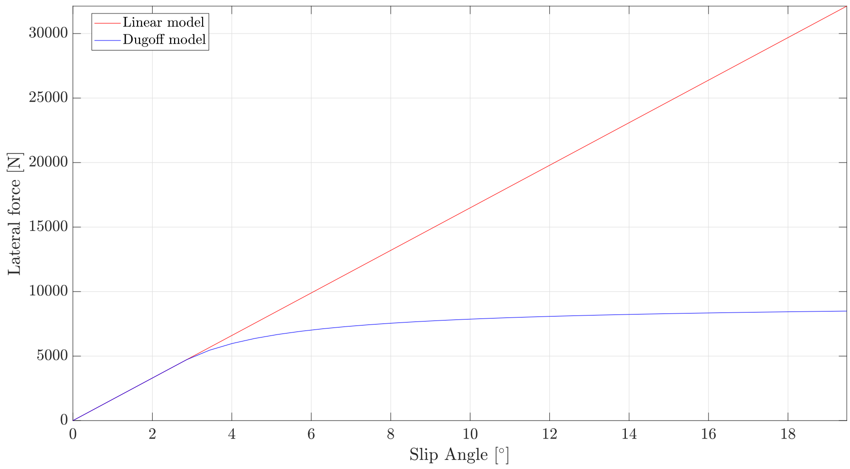


Figure 4.2: Comparison between the unsaturated linear model (red line) and the Dugoff model (blue line).

- **Inputs**: Longitudinal jerk, steering acceleration;
- **States**: Longitudinal velocity, lateral acceleration, longitudinal acceleration, steering and steering change rate.

No limits hold for lateral velocity and yaw rate.

Operation Constraints

$$-1.5 \leq X_{error} \leq 1.5 \ \ m \tag{4.19}$$
$$-0.5 \leq Y_{error} \leq 0.5 \ \ m \tag{4.20}$$
$$-0.087 \leq \Psi_{error} \leq 0.087 \ \ rad \tag{4.21}$$

The main aim of these constraints is to ensure that the variance between the PP and the PF path, which are based on different models, is not too large. Theoretically, they should not be necessary since the minimum cost is associated to perfectly matching paths, but their presence enhance safety at the price of a no more unconstrained problem. Figure 4.3 shows the meaning behind the position error constraint (x and y). It similarly holds for the orientation error.

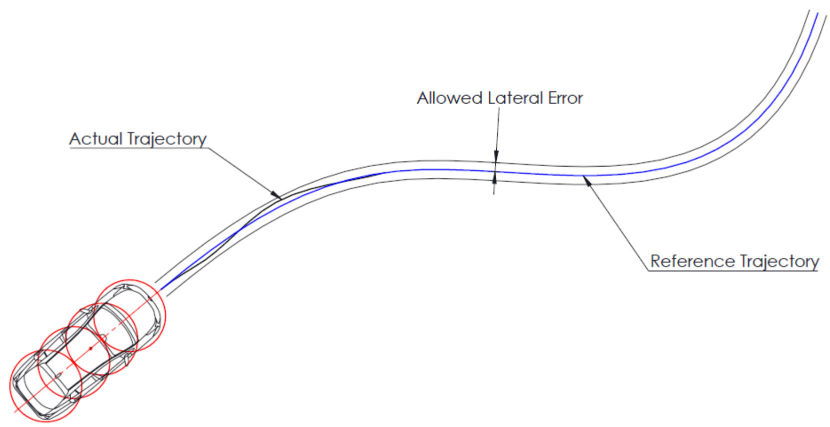


Figure 4.3: Graphical representation of the Path Follower operation constraint.

4.1.3 Performance Index

Here a cost function is created to compute inputs by pushing the PF paths onto the PP one, ensure collision avoidance while considering the dynamics too (see Figure 4.4). With this goal, large cost coefficients are applied to position and orientation errors that become the main costs, while the others are much smaller therefore not reported.

$$J_{tot} = J_{X_{error}} + J_{Y_{error}} + J_{\Psi_{error}} \tag{4.22}$$

4.1.4 Optimization

The PF optimization process is not different from the PP one and, as described in the previous section, the lowest cost respecting constraints is obtained when the dynamic path perfectly matches the PP kinematic path.

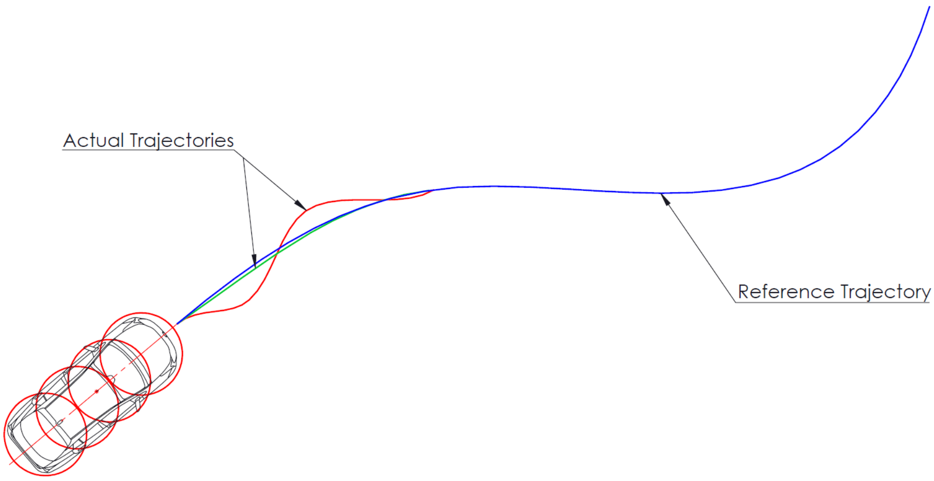


Figure 4.4: The minimum value of the PF performance index corresponds to perfect match with the PP trajectory: the green trajectory is associated to a smaller cost than the red trajectory one.

Since the NLP is not too complicated, the computational time is generally short. This means that the procedure can be carried out multiple times in a single PP time step. Here the horizon is $N = 25$ *steps* and the time step is $dt = 0.02\ s$.

It has to be noted that, since PP and PF work at two different frequencies, interpolation and re-sampling are necessary to make communication possible.

4.2 Path Follower PID-based

This Path Follower consists of a more "classical" approach as substantially made of two PIDs controller (one for each input).

The discrete inputs coming from the PP every $0.1\ s$ are integrated (to obtain acceleration and steering) and them fed into those PIDs as references, while the difference between them and the measured values constitute the errors. As always, the PID controller goal is to minimize that error through feedback and this lead to a smoother control, more promptness and precision. Their tuning has been executed manually, that is following heuristic methods (such as the Ziegler–Nichols method), until when the performances where good enough. Further tuning could slightly improve the result.

Overall, this PF shows good performances but does not provide any

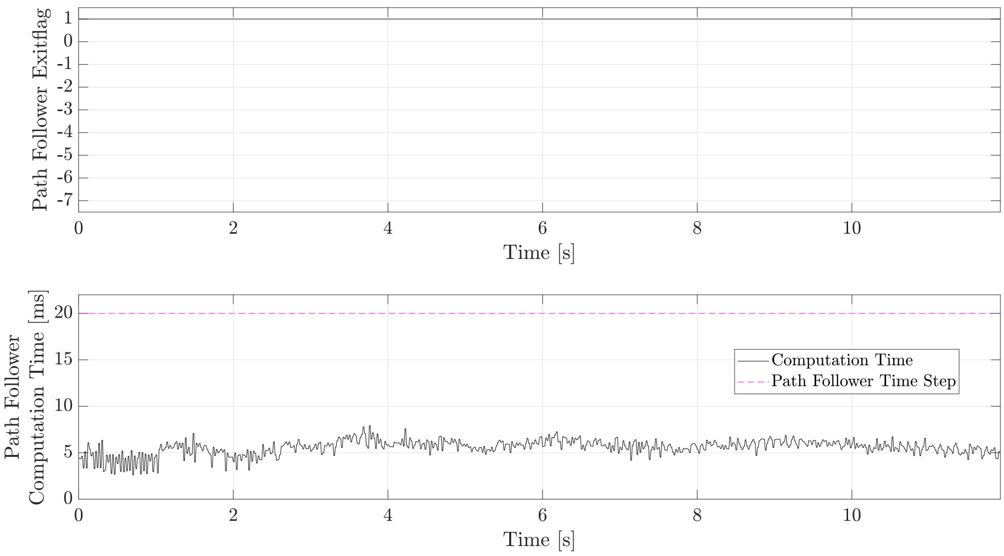


Figure 4.5: Scenario 1 optimization processes results: since the PF problem is relatively simple the computational time is short.

feedback to the PP and can not correct inputs with altered road conditions, which are key features of the MPC-based PF.

4.3 Performance Comparison

In order to illustrate the PF importance, a comparison between LMP performances with and without PF is reported together with a brief pros and cons analysis of the two PFs. Both of them are based on the simple overtaking maneuver studied in Scenario 1 (Section 6.1) .

4.3.1 Path Planner with and without Path Follower

As previously mentioned, the presence of a PF makes inputs smoother. This is possible thanks to the fact that it measures the vehicle actual state at higher frequency compared to the PP one and is based on a more descriptive model. On top of that, the PF is expected to facilitate PP convergence. Actually, due to the fact that the PP time step is long and its model is not really representative (for the reasons explained in Section 3.5), a PF is even necessary to ensure convergence in certain situations (like the one shown below). This because the just mentioned characteristics make the PP predictions slightly inaccurate: the actual next state is not exactly what it was supposed to be. To prove it, the same overtaking reported in

Section 6.1 has been simulated without any PF and the main results are reported below next to the successful ones. In Figure 4.6 the trajectory and decreasing velocity are shown, while in Figure 4.8 it is possible to see how the solver struggles to converge. Basically, the maneuver can not be completed: the safety features are triggered one after the other and the vehicle is forced to stop for safety reasons. Frames of the whole simulation have not been reported for shortness.

4.3.2 Path Followers Comparison

The designed PF incorporates two different PFs in order to improve robustness by combining advantages coming from the different techniques they are based on. Pros and cons are here compared.

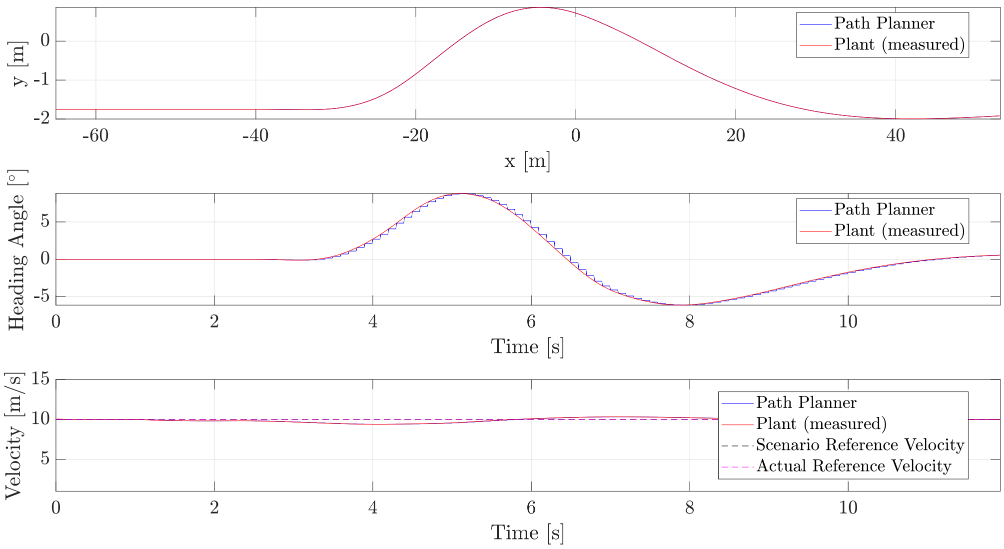
MPC-based Path Follower Pros and Cons

As previously showed, the MPC-based PF is based on a car dynamic model, which includes tire forces (found through a detailed tire model) and road friction coefficient, and predicts car positions and velocities for half of a second. This features bring the following pros with them:

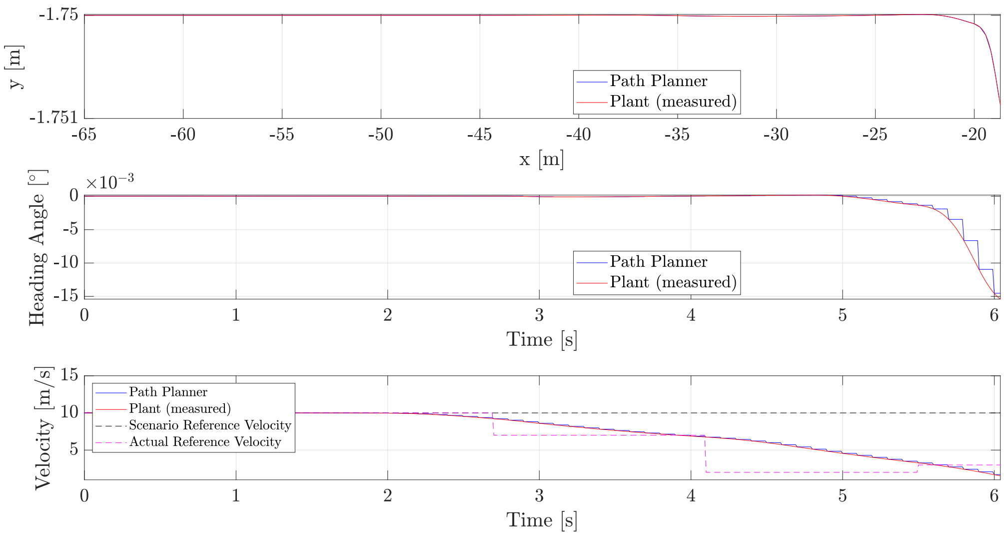
1. Much smoother and accurate transition between PP way-points (the "dots") that can then be placed further away;
2. Inputs correction according to the dynamics missed by the PP;
3. Optimal inputs generation by definition;
4. Possibility to send future positions error respect to obstacles to the PP;
5. Live change of the road friction coefficient to ensure the best handling (see Section 6.1.2);

Especially the last two can enhance safety since paths deviations, hence obstacle enlargement, are expected to increase in case of road friction coefficient inaccurate estimation. However,this PF also comes with some cons:

1. Additional computational power required;
2. The implemented model has to be simple to ensure real time operation;

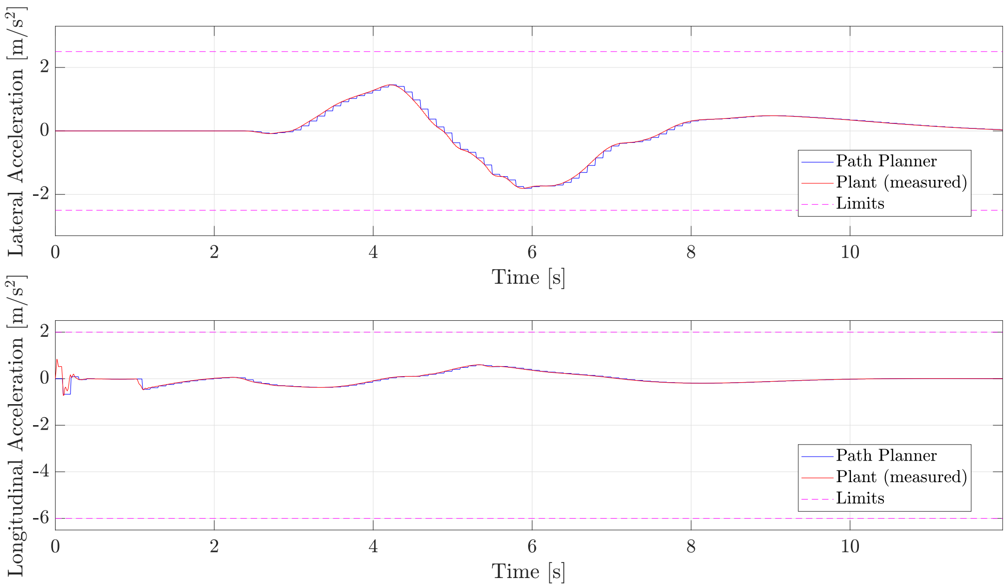


(a) Position, yaw and velocity of the controlled vehicle from both Path Planner (predicited) and Plant (measured), with PF.

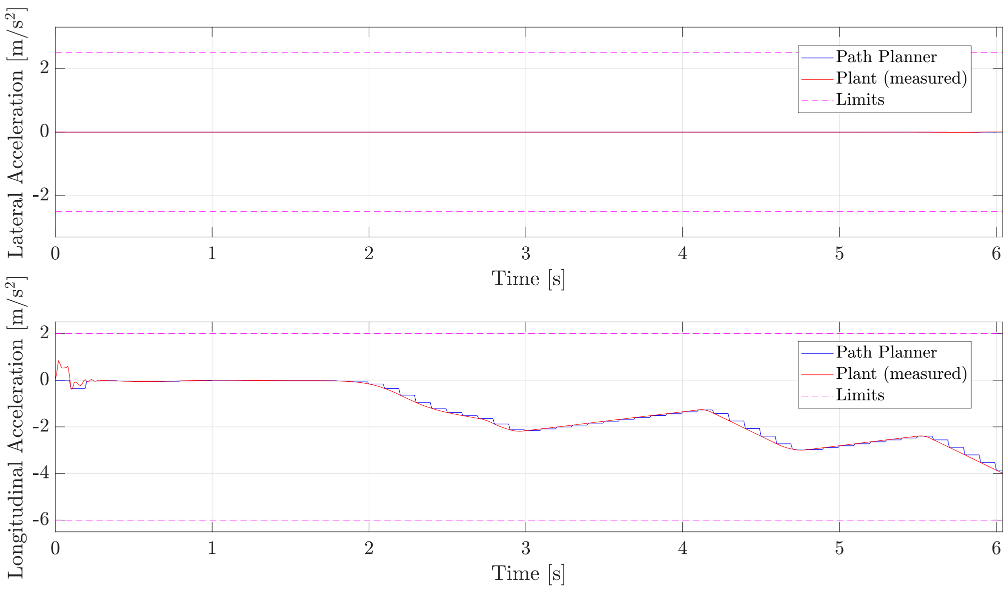


(b) Position, yaw and velocity of the controlled vehicle from both Path Planner (predicited) and Plant (measured), without PF.

Figure 4.6: Trajectory, heading angle and velocity with and without Path Follower.

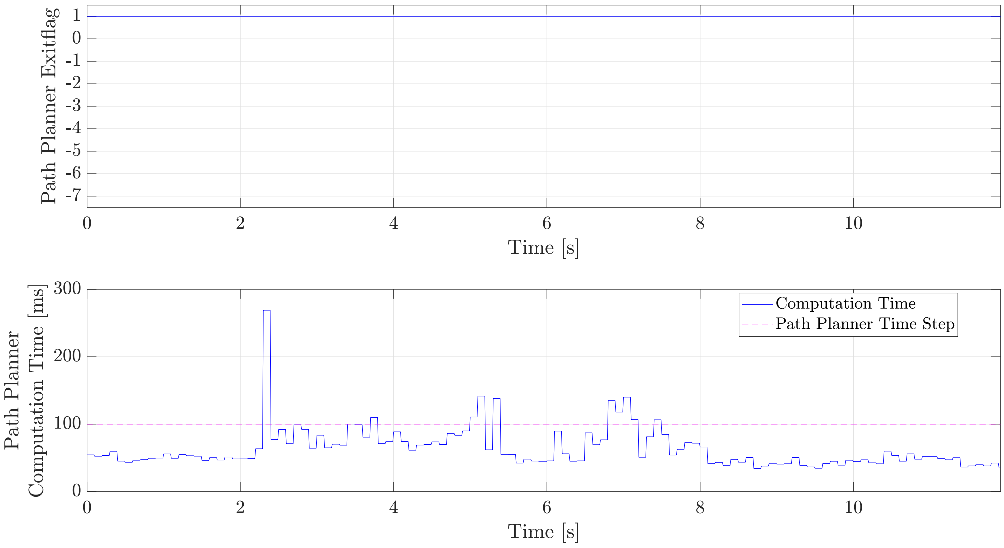


(a) Longitudinal and lateral acceleration of the controlled vehicle with PF.

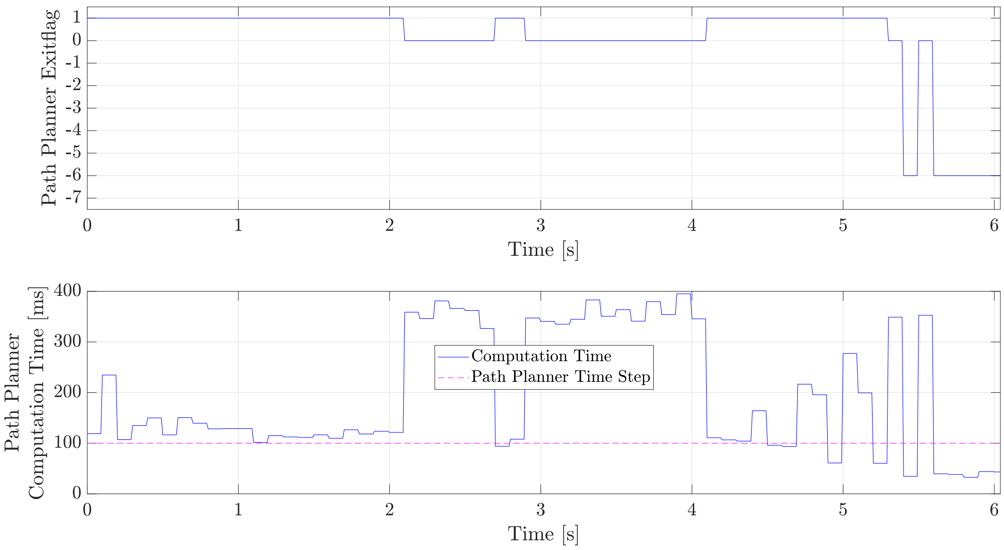


(b) Longitudinal and lateral acceleration of the controlled vehicle without PF.

Figure 4.7: Lateral and longitudinal acceleration with and without Path Follower.



(a) Path Planner optimization result with PF.



(b) Path Planner optimization result without PF.

Figure 4.8: Path Planner optimization performances with and without Path Follower.

3. Possible failure in generating inputs. Low velocity in particular can lead to numerical issues due to the model structure.

A proper combination of powerful solvers and adequate hardware can easily tackle the first two issues, while some expedients in the code limit all of them. The presence of a secondary PF also gives a positive contribute.

PID-based Path Follower Pros and Cons

The main advantage of using a PID PF is that it never fails, and can therefore supply usable inputs just following the PP ones, while ensuring good performances.

The downside of using it alone is the fact that all the MPC-based PF advantages are missed, although they are quite useful. Again, the key lies in employing them together.

Chapter 5

Tailored Simulation Environment

In this chapter the key features of the tailored simulation environment that has been created are described. It allows to test different maneuvers and road conditions while having multiple agents following predefined trajectories and modeled in different ways. The whole environment is developed in MATLAB & Simulink [57].

5.1 Plant

The created simulation environment includes three different vehicle simulators (or plants) associated to different levels of fidelity and usability.

High Fidelity Plant

This simulator has been realized within the Simscape Multibody environment [58]. It consists of 9 degrees of freedom which come from translation motion (longitudinal, lateral and vertical), rotational motion (yaw, roll and pitch) and the suspension. To model the tire behavior the Delft-tire 6.2 [59] with Magic Formula steady-state slip model is implemented. It describes the nonlinear slip forces and moments. The relaxation behavior in the tires is implemented by empirical relations for the relaxation lengths. The steering of the plant has simplified dynamics. The steering is implemented with a transfer-function and time delay including the Ackerman geometry. The vehicle model has been validated [60] and represents a Toyota Prius of the third generation. It is the same plant employed in [61]. This plant can reproduce the real behavior quite well

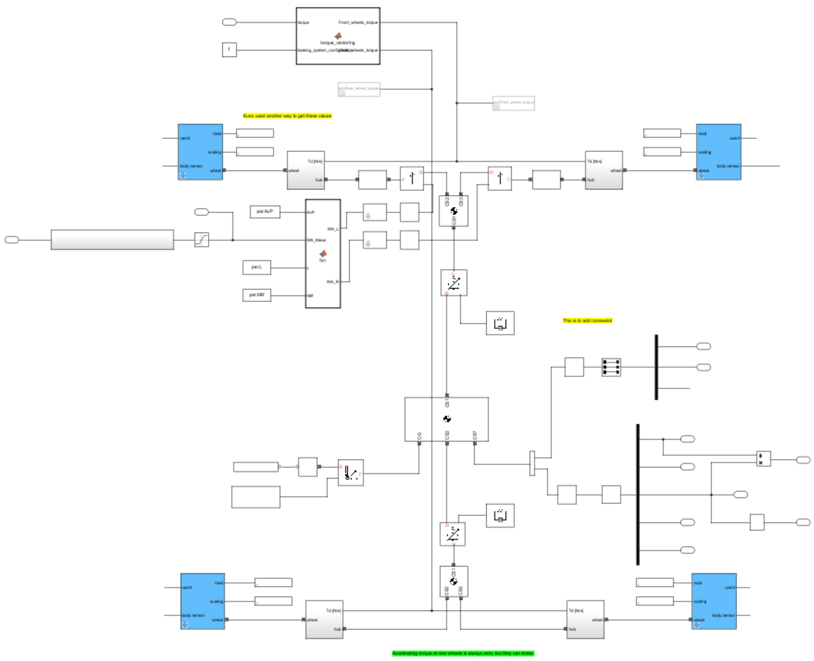


Figure 5.1: High fidelity plant realized with Simscape Multibody [58].

although numerical problems may arise at velocity lower than 6 m/s.

Dynamic Plant

It basically reproduces the same car dynamic model the Path Follower is based on, including the Dugoff model to estimate tire forces. This plant allows to test low speed maneuvers but is not as accurate as the previous one.

Kinematic Plant

It reproduces the Path Planner car kinematic model. It is useful for simple checks when major changes are introduced.

5.2 Map

The realized map includes some aspects of a real environment in which future AVs will operate, such as a straight road, an intersection, a tight curve, a crossing and cycling lanes. They are all visible in Figure 5.2

Roads have two lanes 3.5m wide each and cycling lanes on the sides. The main straight road allows to test overtaking maneuvers. The intersection, which is also a tight curve, is included to simulate turnings and sudden insertion of other vehicles into the occupied lane. The crossing aims to simulate the necessity of an emergency braking or avoidance, that are usually hard-to-handle situations.

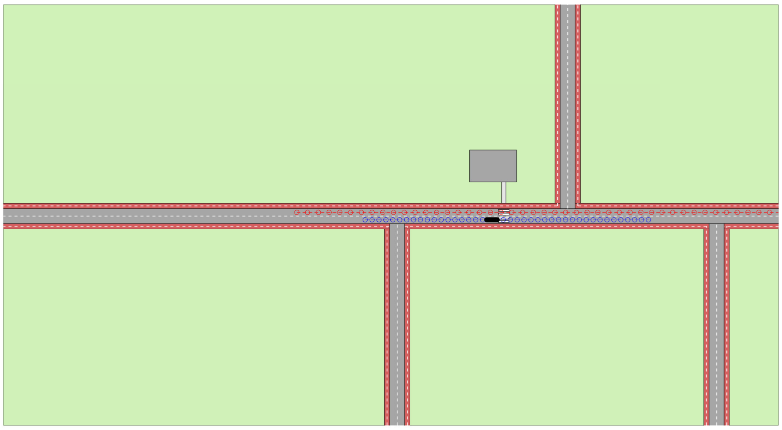


Figure 5.2: Top view of the created simulation environment with Scenario 1 trajectories of all agents (blue, red and black sequence of circles).

5.3 Agents

In this environment an agent is basically a moving obstacle whose position and state predictions are considered when planning the next path. Currently, two agents more than the controlled one have been considered, and this comes with a certain computational load. It increases as soon as more obstacles are taken into account. However, the structure of the created environment allows to do that quite easily and see how the LMP performances change.

Each agent follows a trajectory that needs to be specified in the form of way-points before the simulation starts, while the velocity remains the same once set. No dynamics is taken into account for them.

5.4 Pseudo-Detection

In real AVs the detection module is responsible for detecting other obstacles and predicting their intentions in terms of future movements

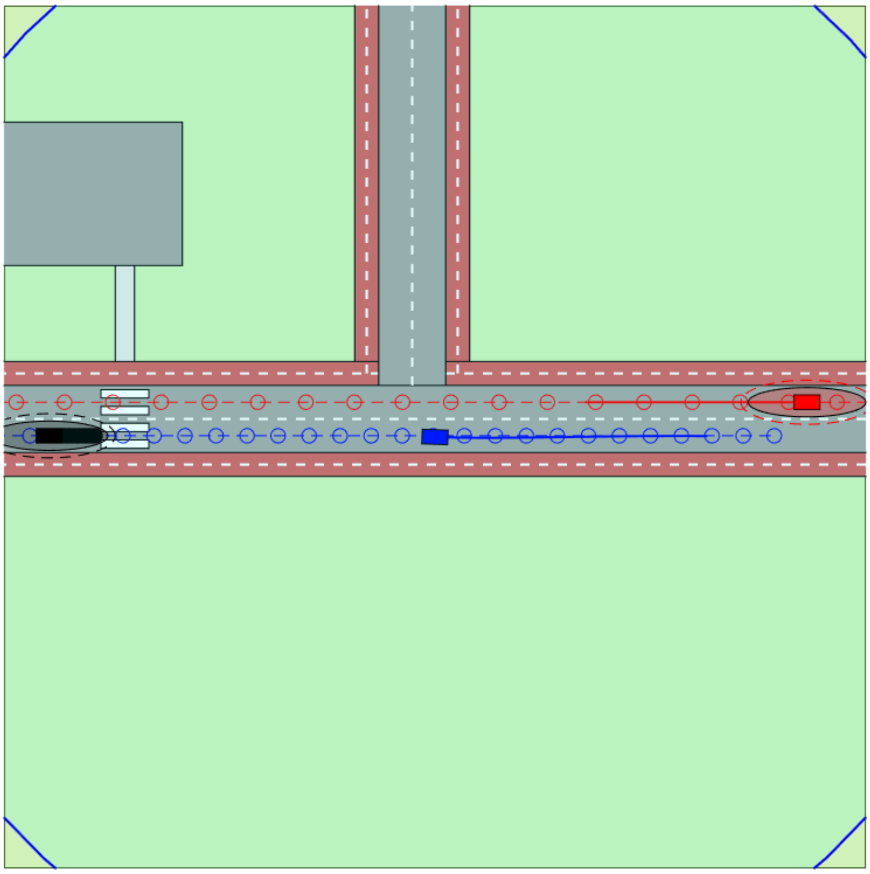


Figure 5.3: The black and red rectangles with respective egos (ellipses) represent the two other moving obstacles (i.e. agents) present in the tested scenarios. The blue rectangle represents the controlled agent.

Section A. This is definitely a hard task to carry out and to implement, due to the extreme real-world variability, weather conditions and possible obstructions. Nevertheless, it can influence performances quite a lot. For this reason a simplified version has been implemented to at least take it into account. As mentioned in 6.5, more research should be done to find ways on how to handle variability in uncertain dynamic environment (like in [62]).

The two main settings when simulating regard *obstacle localization*, that can be

1. Infinite around the vehicle;

2. Finite around the vehicle according to sensors (here 60 m).

and *obstacle state prediction*, which can be in form of

1. Perfect knowledge (like V2V communication);

2. Straight movement aligned to the momentary orientation while assuming constant speed.

The last setting in particular has demonstrated of being able to make quite a lot of difference in terms of effectiveness: *Vehicles to Everything Communication (V2X)* (with other road users, traffic management and other vehicles), or simply its reduced version V2V makes the planning task remarkably lighter in terms of computational workload, as it is later shown in different simulations.

5.5 Simulation Settings

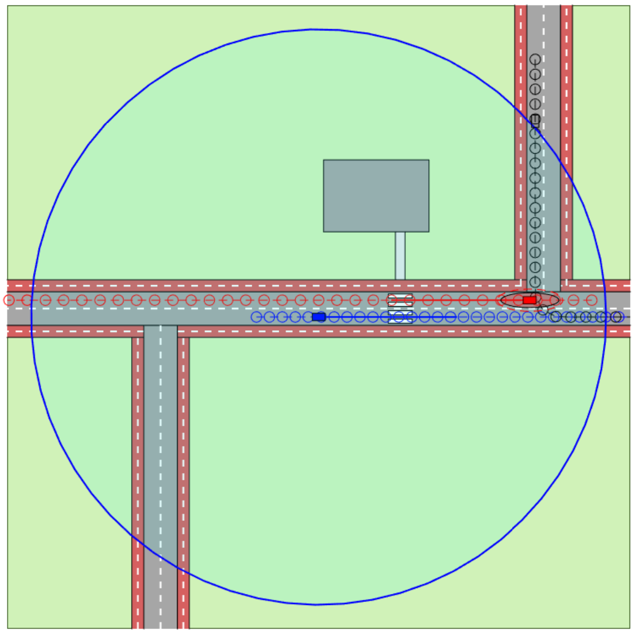
Overall, this simulation environment is equipped with multiple settings. Some of them were mentioned in previous sections. This allows to point out performance variation according to the situation faced. They are:

1. **Plant**: Simscape Multibody, dynamic or kinematic

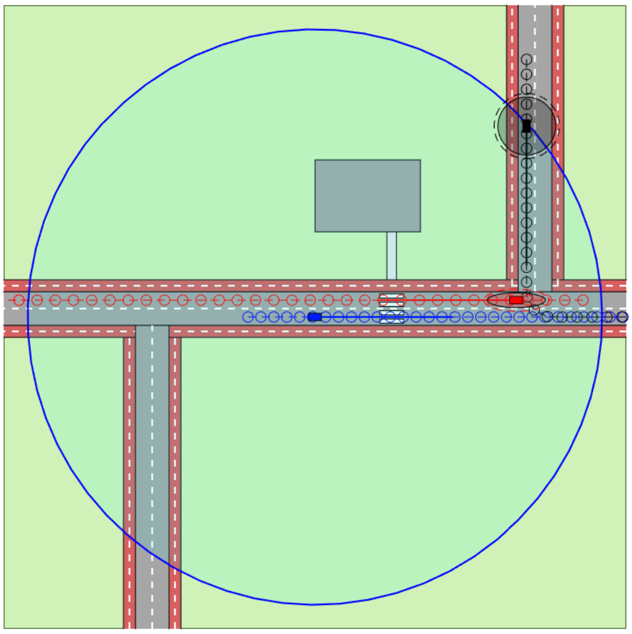
2. **Path Follower**: PID only, MPC only, combination of both PID and MPC or no PF (see Figure 2.2);

3. **Additional Safety Features**: active or not (Section 2.2.2);

4. **Road friction coefficient**. it is assumed to be 1 if not differently specified.



(a) The black agent is right outside the detection range.



(b) The black agent is just inside the detection range: future postions are predicted assuming it will keep the same orientation and speed.

Figure 5.4: Pseudo-detection operating mode: the light blue circle represents the detection range (here 60 m).

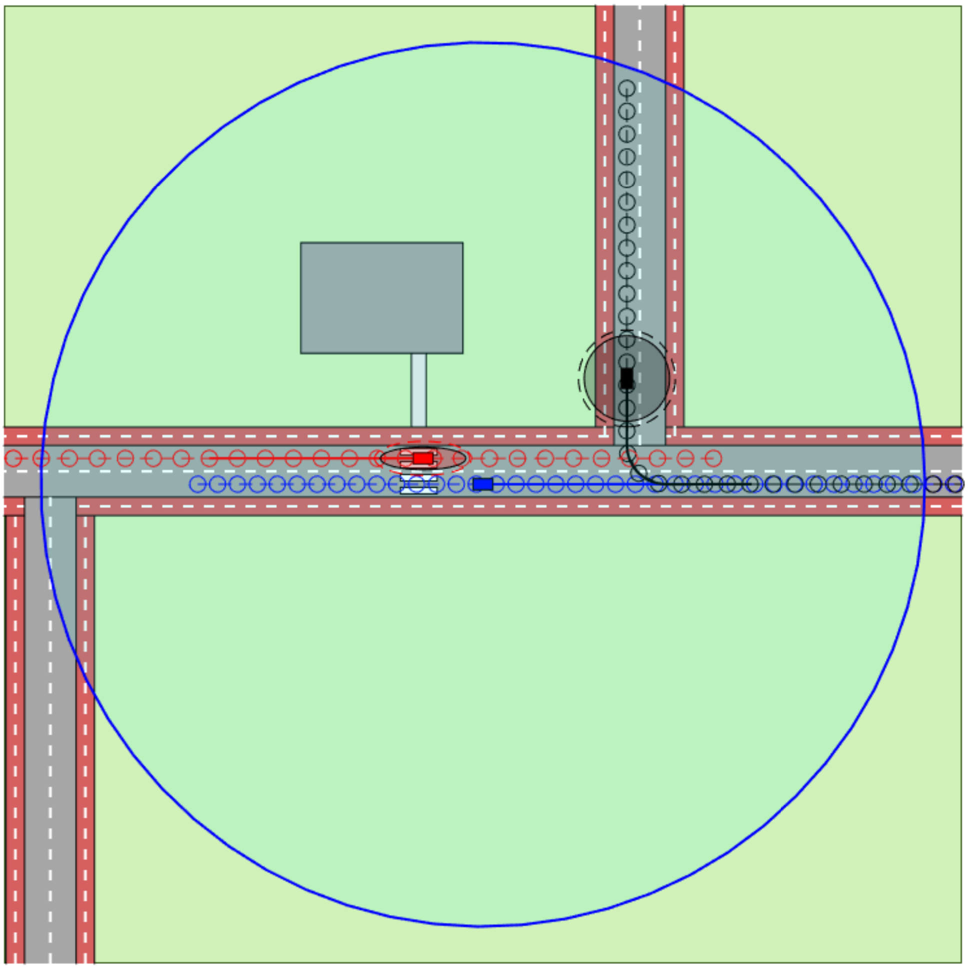


Figure 5.5: Agents state prediction strategy reproducing V2V (Vehicle to Vehicle) communication, in this case applied to the black agent.

Chapter 6

Simulations

In this chapter the LMP functioning is analyzed when operating in different scenarios, namely the most common maneuvers that an AV faces when operating in a real urban scenario. Each simulation provides diagrams and a video whose only most important frames are reported in this document for shortness reasons. In each frame there are captions reporting longitudinal acceleration a_x, lateral acceleration a_y, steering δ, velocity v, solvers outputs and agents distances (if available).

As reported in Section 5, different settings are possible to make each simulation case specific. If not differently specified, these ones are common to all simulations:

1. **Plant**: High Fidelity Plant (Simscape Multibody);
2. **Path Follower**: Combined Path Follower (MPC & PID);
3. **Safety Features**: Active.

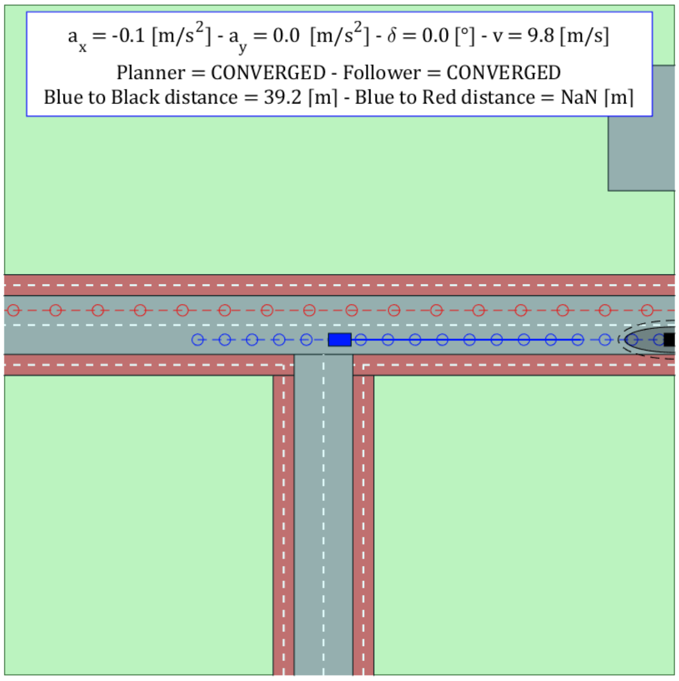
Overall, they show how this LPM can proceed safely and smoothly through different context, and how versatile is the created simulation environment.

6.1 Scenario 1 - Overtaking

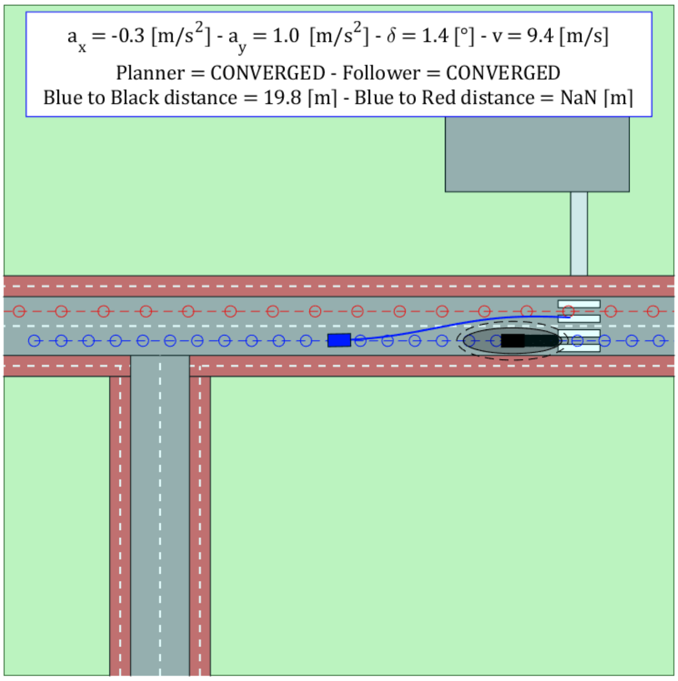
In this scenario the car is required to overtake a stationary obstacle if the situation allows it, which means it needs to check that no pedestrians or cyclist are approaching the crossing or vehicles coming along the other lane.

Throughout the whole simulation, the plant follows the PP paths almost perfectly and keeps the velocity as close as possible to the reference one ($10\,m/s$) Figure 6.2. The registered lateral and longitudinal accelerations changes smoothly and largely remain within the specified limits, which can be considered an indicator of good comfort. In terms of computational performances, the PP almost always converges to an optimum value within its time step, while the PF (MPC-based) largely operates in the given time span. In Figure 6.4 the tire forces computed and used by the PF are compared with the ones registered during the simulation after sum. They are not exactly the same but it has to be kept in mind as the planning is carried out with a car bicycle model, that has one front wheel, while the plant can provide wheel-specific forces (both front-right and front-left). In Figure 6.5 the estimated differences between the agents distance (respect to the controlled car) computed by the PP and the PF are shown together with the detected distances. The largest values is around $1\,cm$ in the actual overtaking phase, when the plant is indeed expected to deviate from the kinematic paths.

It is worth to mention one more time as the results can widely change according to the chosen PP and PF cost coefficients, which again points out the importance of a proper tuning.

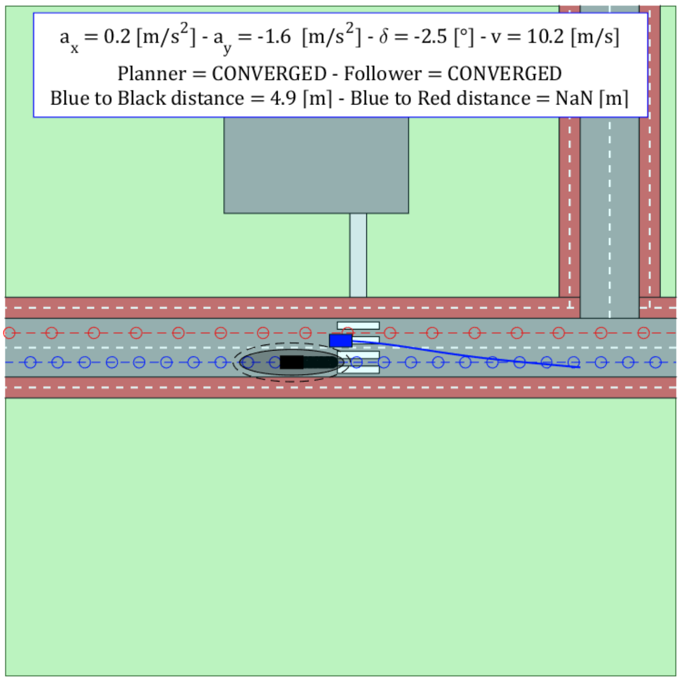


(a)

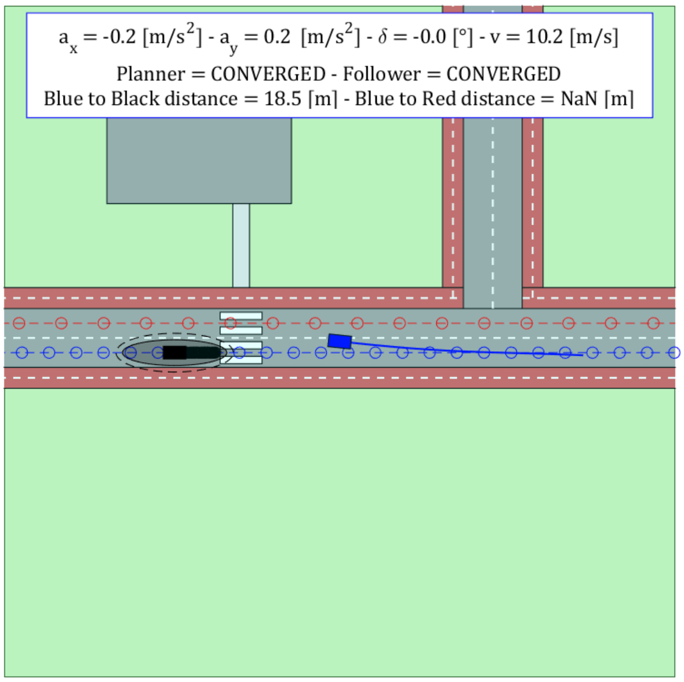


(b)

Figure 6.1: Scenario 1 frames.

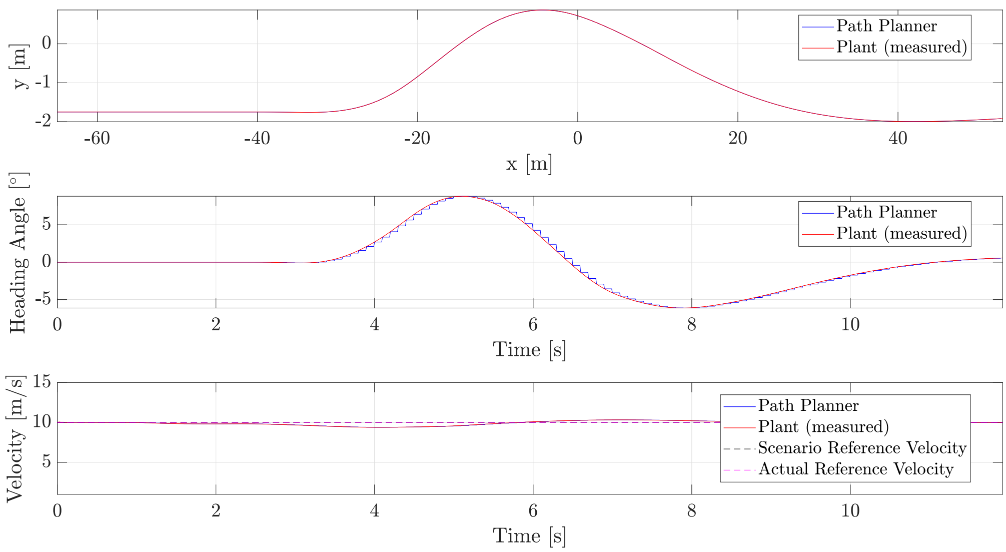


(c)

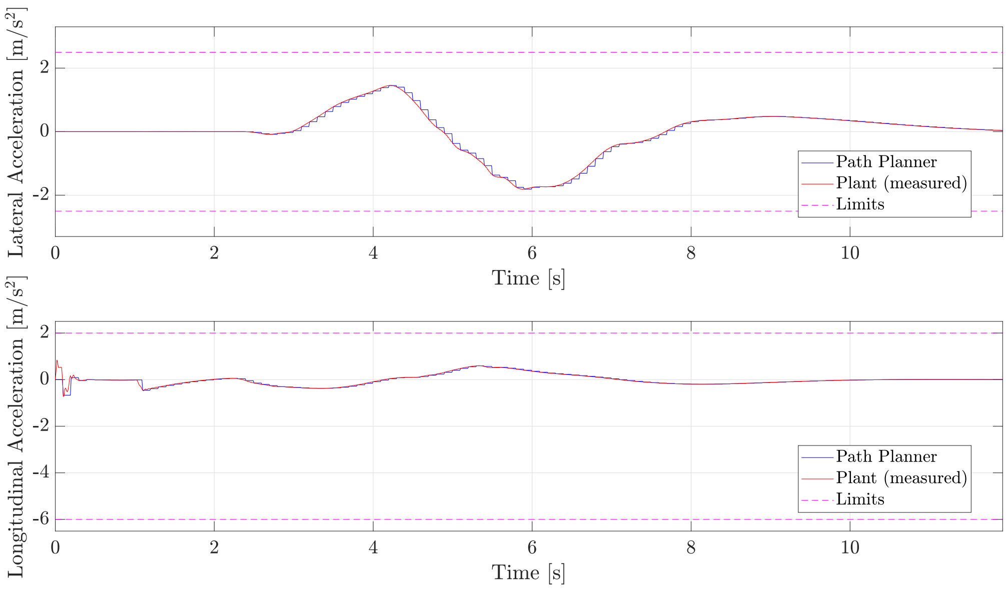


(d)

Figure 6.1: Scenario 1 frames.

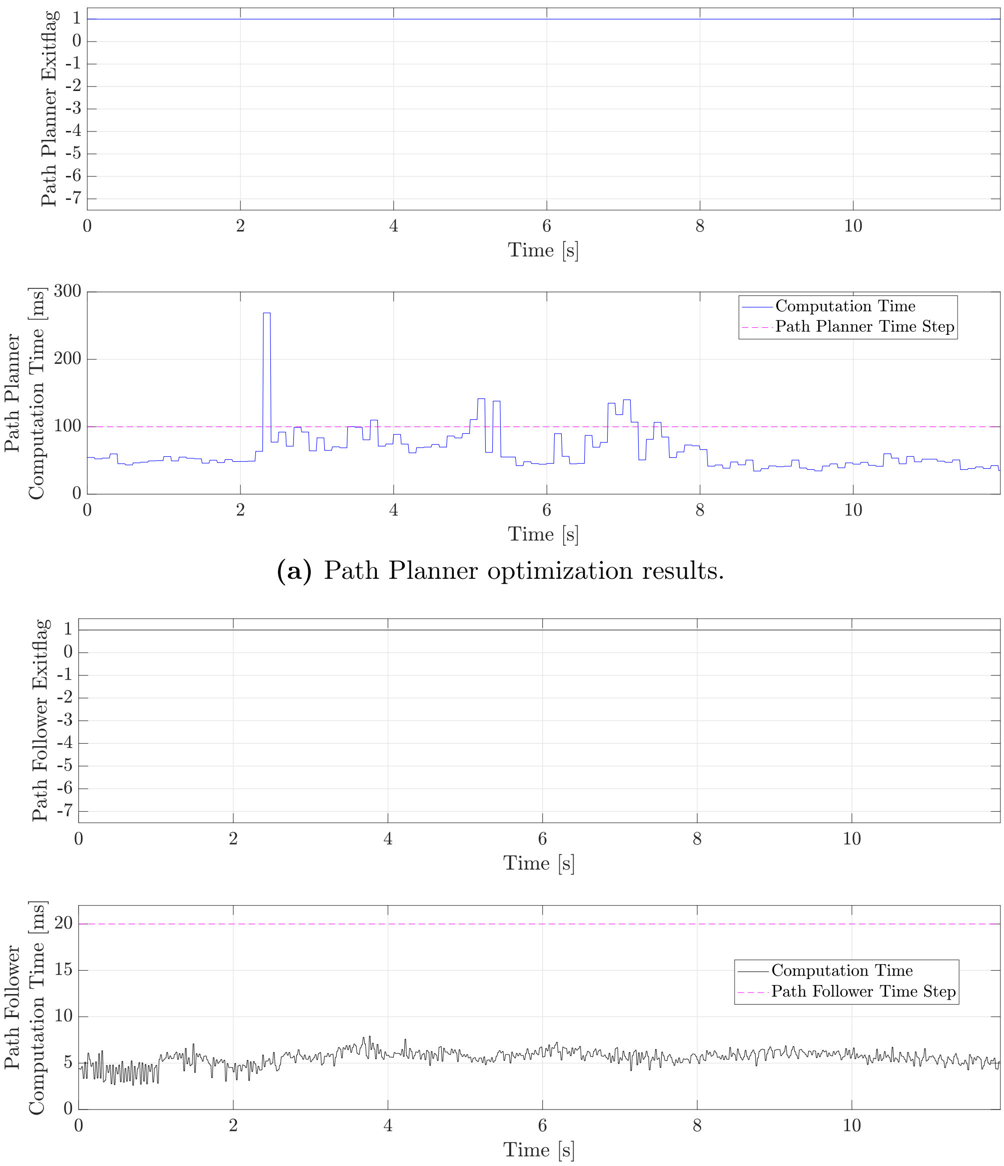


(a) Position, yaw and velocity of the controlled vehicle from both Path Planner (predicted) and Plant (measured).



(b) Lateral and longitudinal acceleration of the controlled vehicle.

Figure 6.2: Scenario 1 simulation results.



(a) Path Planner optimization results.

(b) Path Follower optimization results.

Figure 6.3: Scenario 1 optimization processes results.

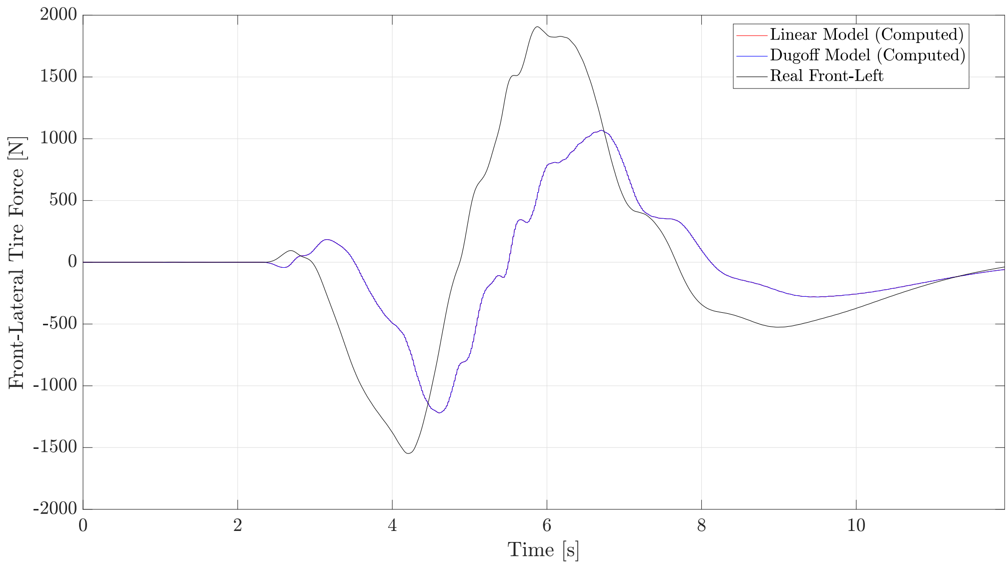


Figure 6.4: Scenario 1 tire forces.

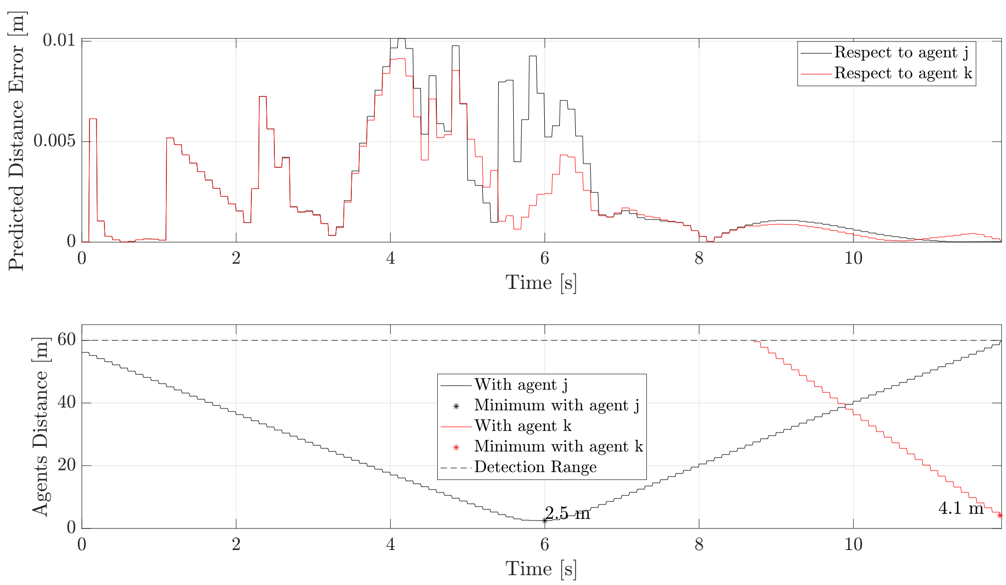
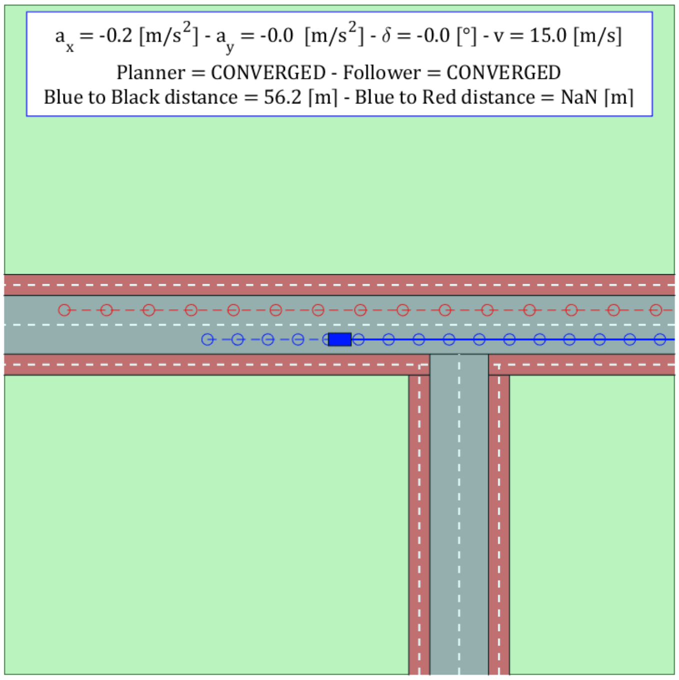


Figure 6.5: The MPC-based PF allows to estimate future distances between the controlled agent and the moving obstacles more accurately. They are then compared to the PP ones in order to find out the maximum deviation, which is in this case smaller than 1 cm. This same information is used to enlarge obstacles ego and enhance safety (Section 4.3.2). Right below the detected distances.

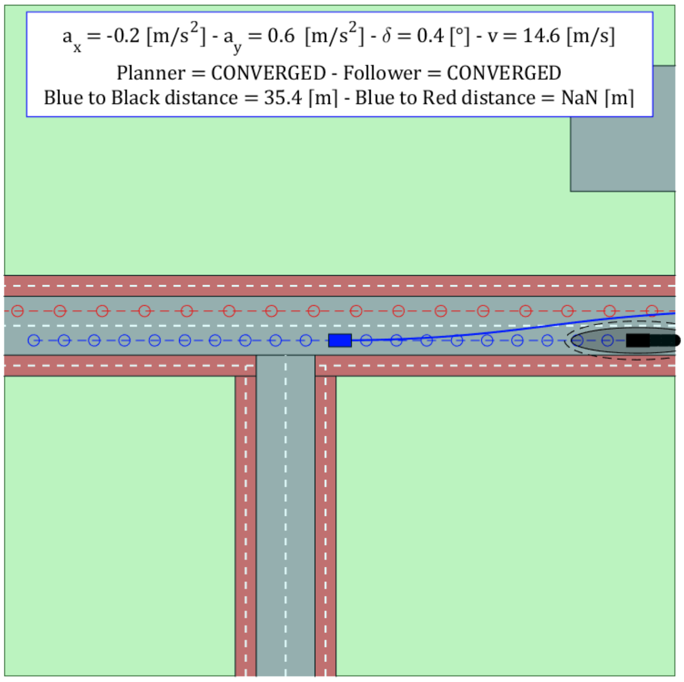
6.1.1 At High Velocity

In this scenario the controlled agent is required to carry out exactly the same maneuver but at 15 m/s instead of 10 m/s.

As before, the plant follows the PP paths almost perfectly and keeps the velocity as close as possible to the reference one Figure 6.7, while lateral and longitudinal accelerations are a bit larger in magnitude but still smooth. The optimization process are executed almost as good as before except for a couple of instants when the PP could not return an optimum set of values within the allowed number of iterations. Regarding the estimated maximum deviation between the distance from obstacles predicted by PP and PF, as reported in Figure 6.9 the largest values is around 2 cm in this case.

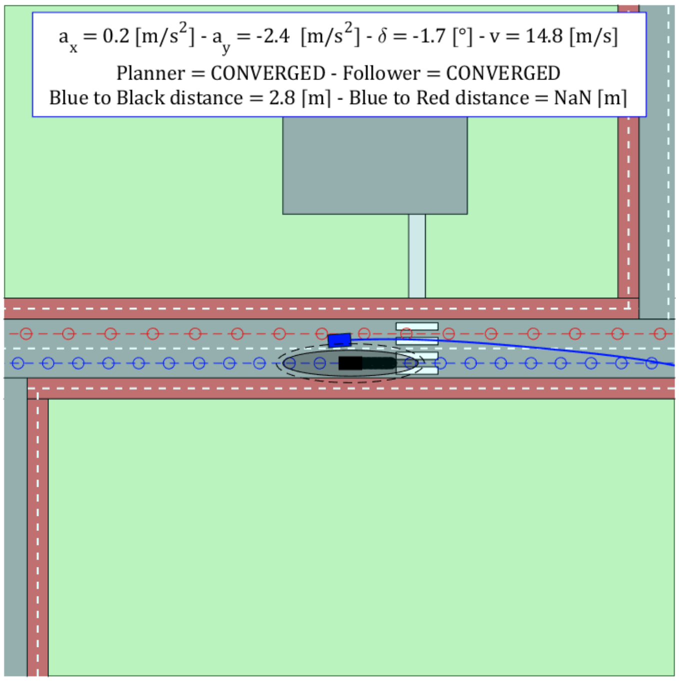


(a)

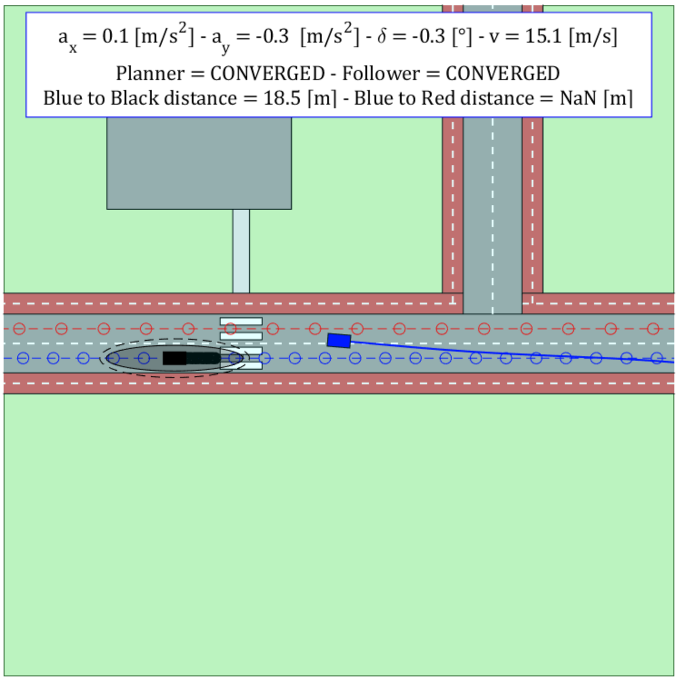


(b)

Figure 6.6: Scenario 1 at high velocity frames.

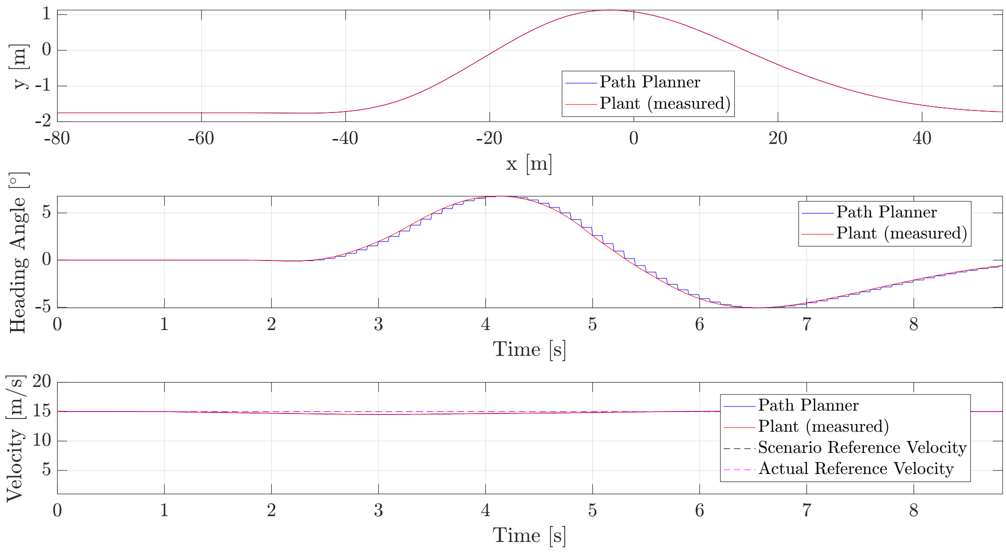


(c)

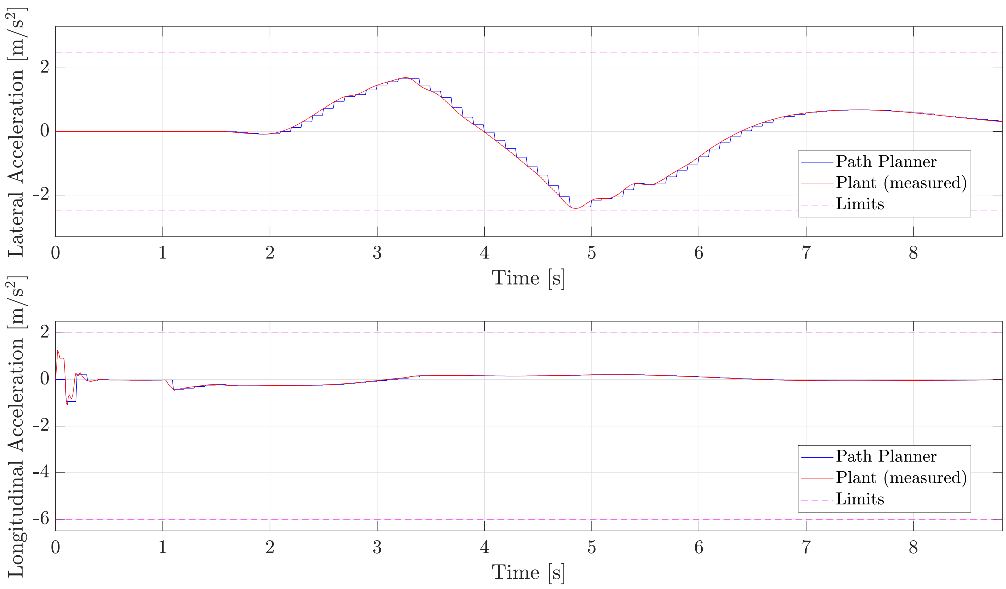


(d)

Figure 6.6: Scenario 1 at high velocity frames.

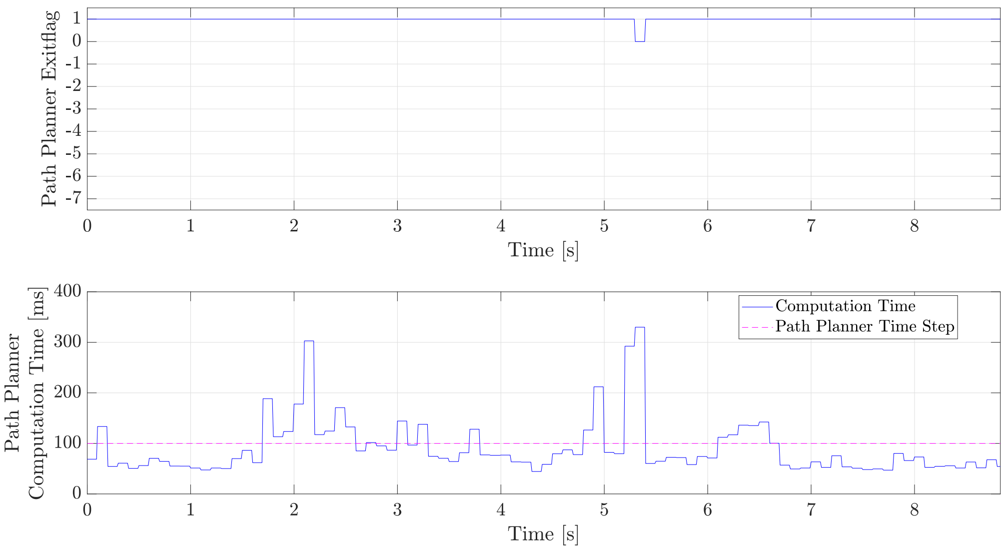


(a) Position, yaw and velocity of the controlled vehicle from both Path Planner (predicted) and Plant (measured).

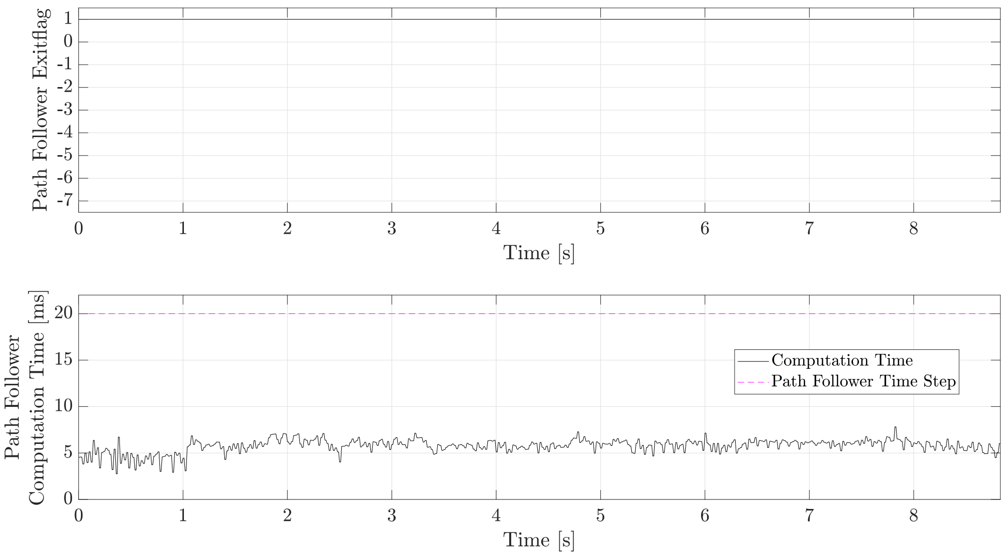


(b) Lateral and longitudinal acceleration of the controlled vehicle.

Figure 6.7: Scenario 1 at high velocity simulation results.



(a) Path Planner optimization results.



(b) Path Follower optimization results.

Figure 6.8: Scenario 1 at high velocity optimization processes results.

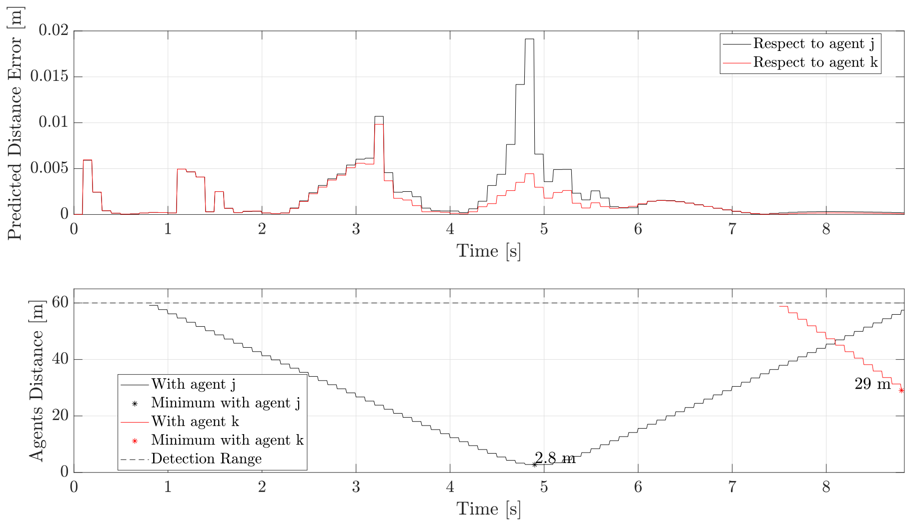


Figure 6.9: Estimated maximum deviation between PP and PF prediction of distances from obstacles and actual distances.

6.1.2 With Friction Drop

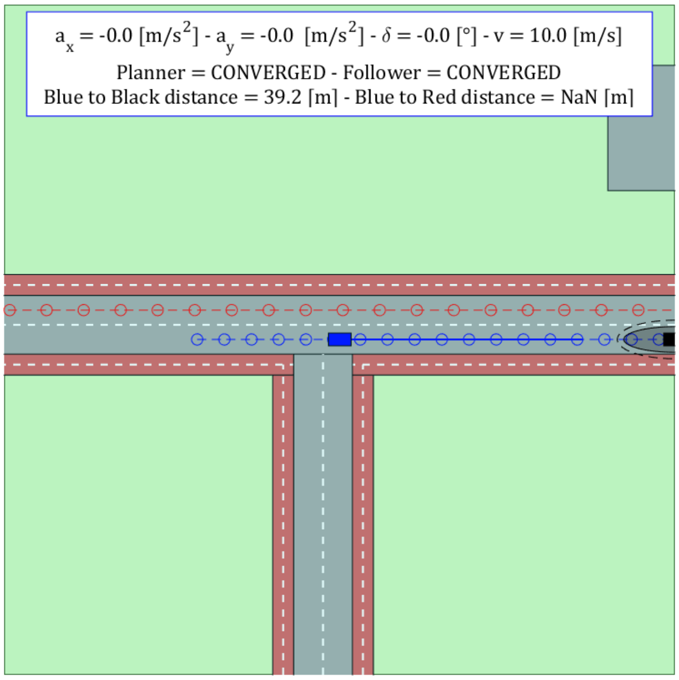
In this scenario the controlled agent is required to carry out the same maneuver as before at 10 m/s but on a slippery road whose friction coefficient is estimated to be 0.3.

Firstly, the simulation is executed with the PID based PF to show its limitations: the vehicle goes off road. Secondly, the combined PF is employed and, thanks to the MPC-based PF contribute, the maneuver is successfully completed.

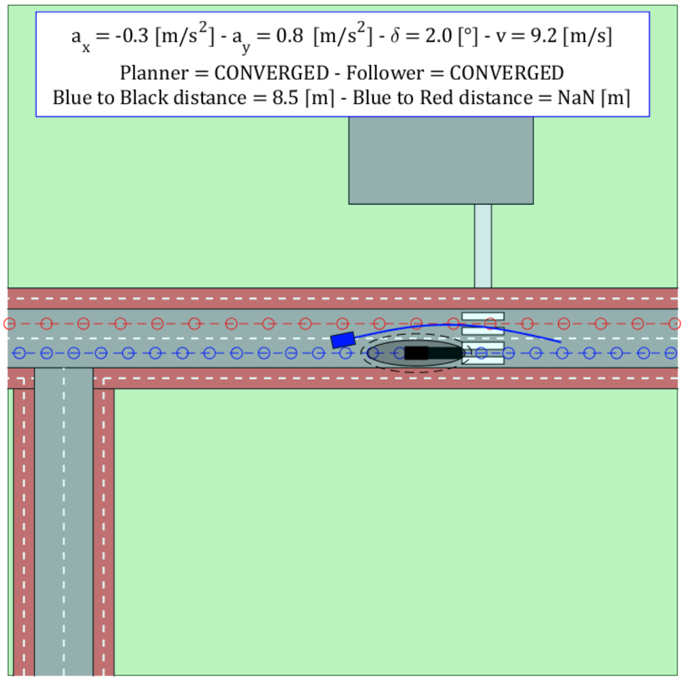
No friction coefficient correction within the PF

Since the PID based PF is not able to take any dynamics or situations specificity into account, the inputs are not properly corrected for a situation like this and the car continuously put itself in a configuration which is different from the planned one.

At the end of the simulations, when the car control is basically lost, the reference velocity is reduced due to the activation of the safety features (see Figure 6.11). Eventually, the simulation stops when the vehicle ends off road. The optimization processes were completed successfully until a reasonable deviation from the expected trajectory was kept. Although the MPC PF should not be considered as it was not active in this simulation, Figure 6.12 shows how it started failing first, while the PP followed right after. Regarding the tire forces, they are reported for a comparison with the next section even though the graphs do not provide much useful information due to loss of control.

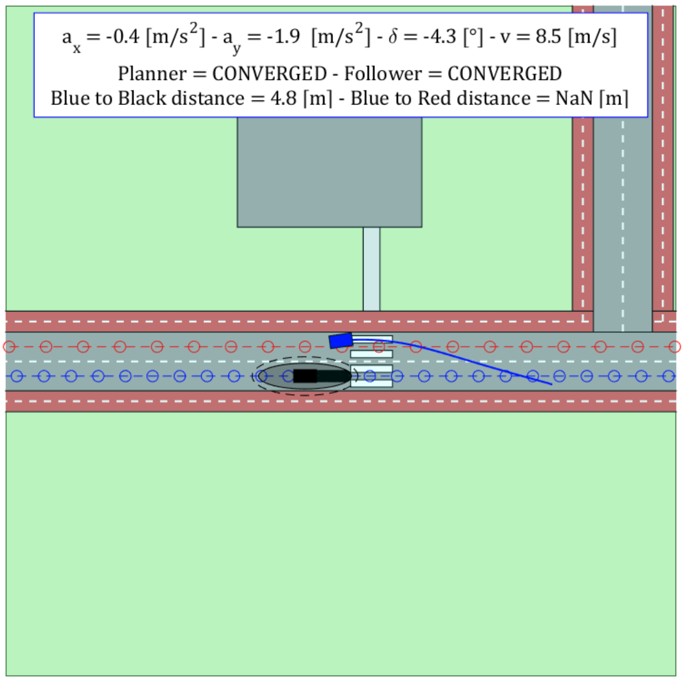


(a)

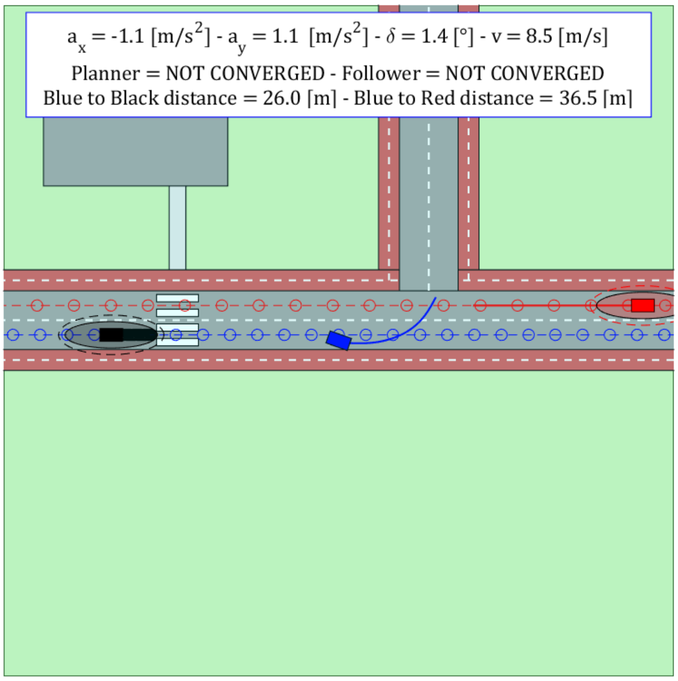


(b)

Figure 6.10: Scenario 1 with friction drop but no correction frames.

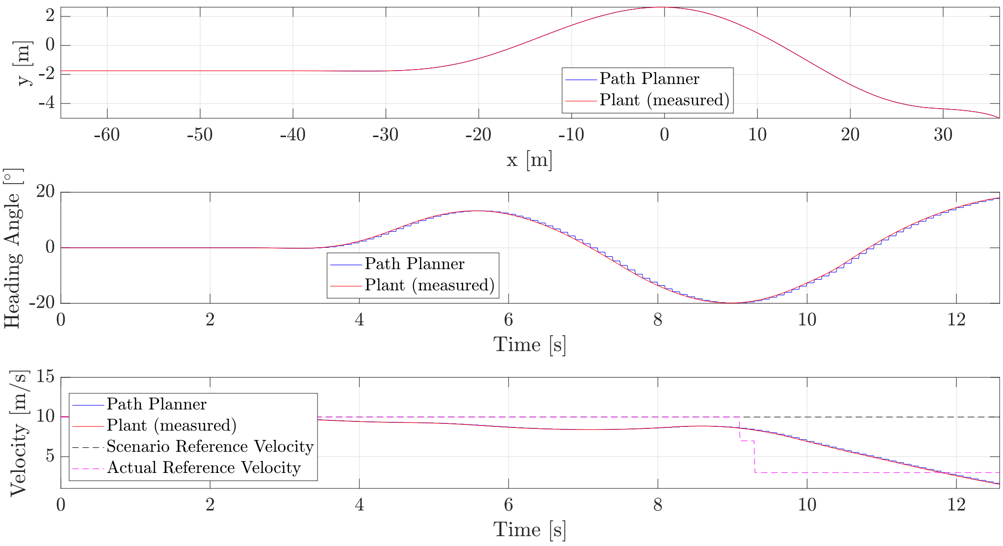


(c)

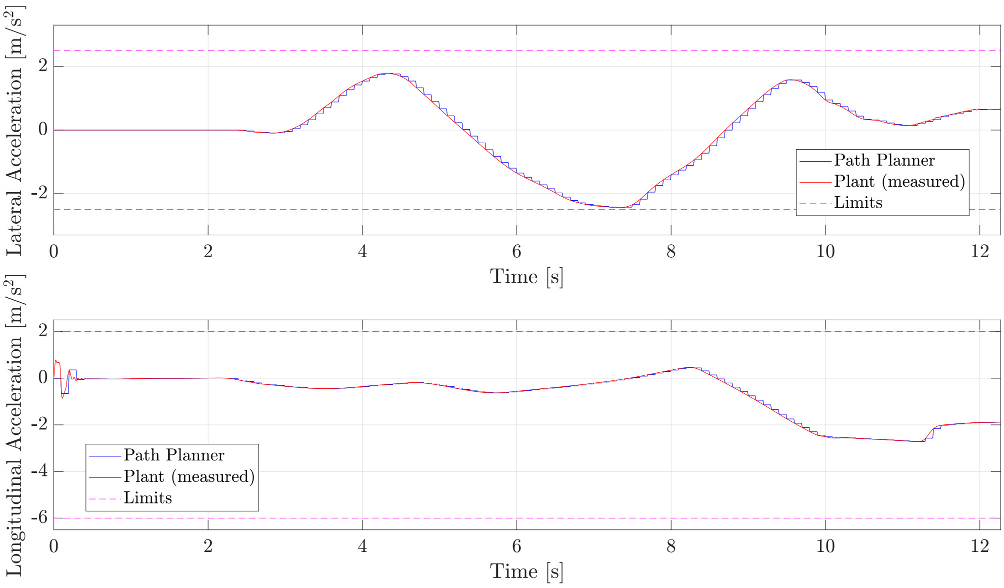


(d)

Figure 6.10: Scenario 1 with friction drop but no correction frames.

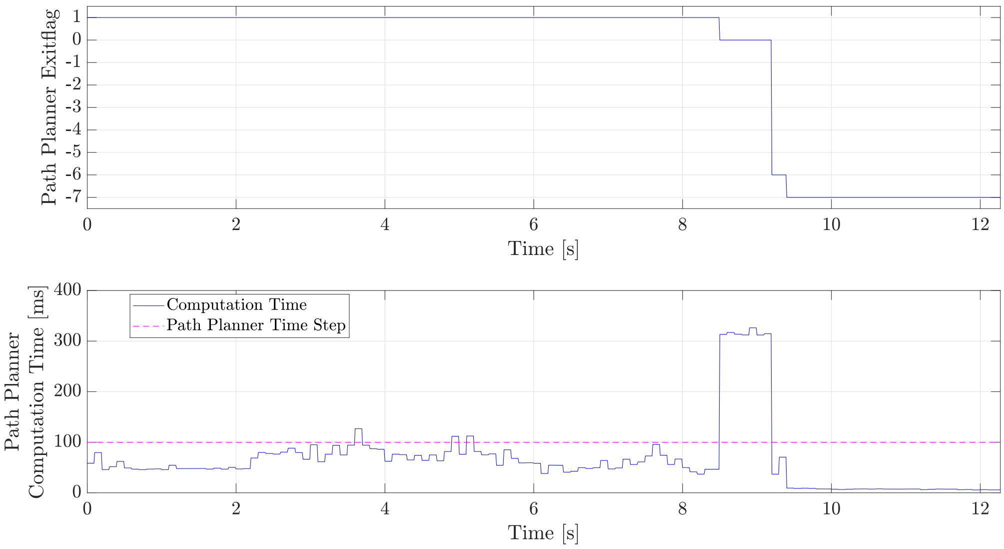


(a) Position, yaw and velocity of the controlled vehicle from both Path Planner (predicited) and Plant (measured).

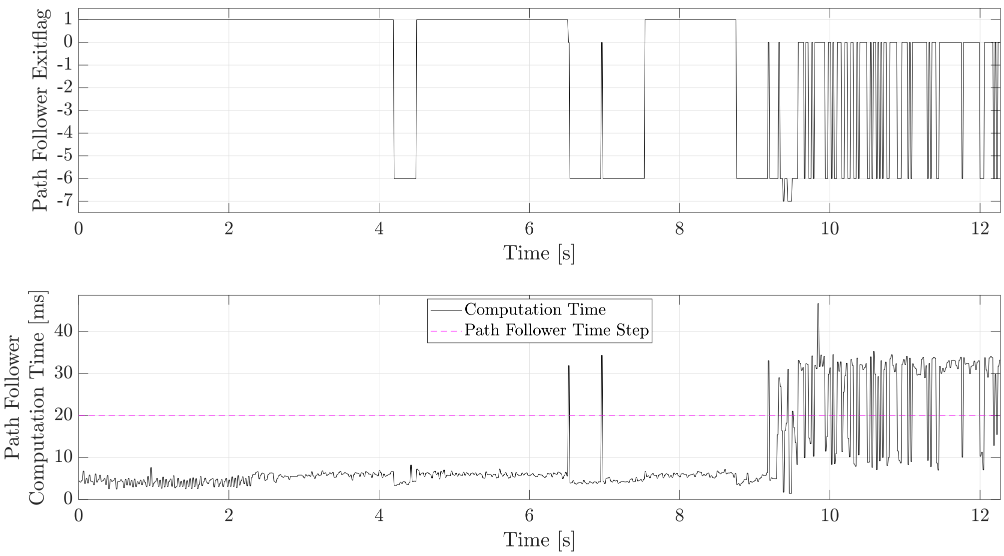


(b) Lateral and longitudinal acceleration of the controlled vehicle.

Figure 6.11: Scenario 1 with friction drop but no correction simulation results.



(a) Path Planner optimization results.



(b) Path Follower optimization results.

Figure 6.12: Scenario 1 with friction drop but no correction optimization processes results.

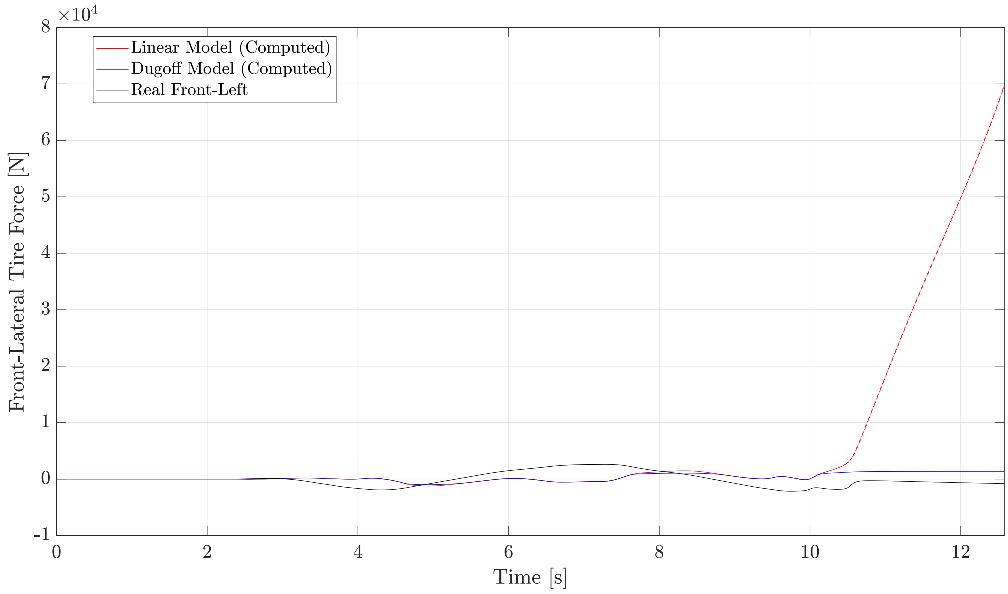
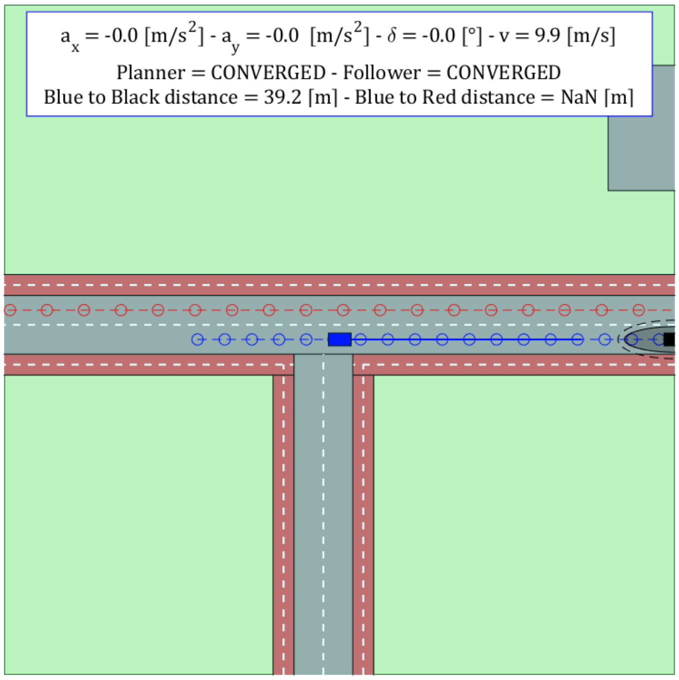


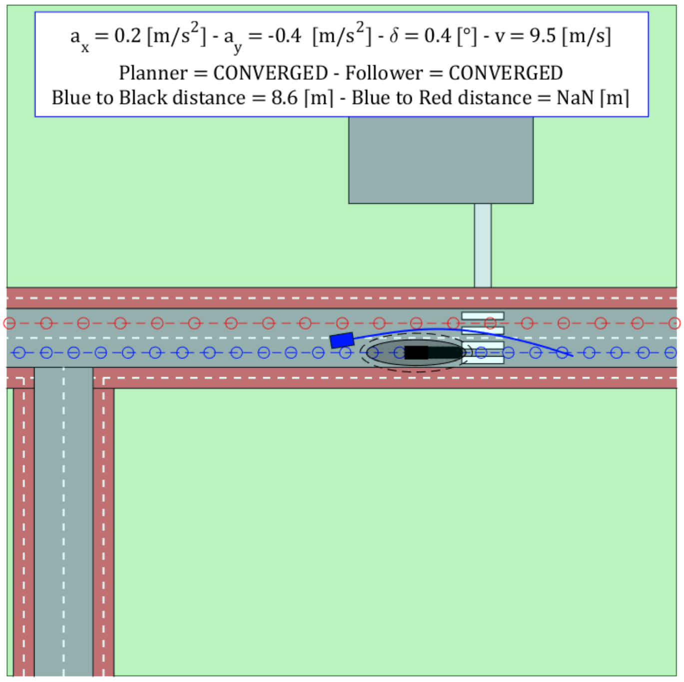
Figure 6.13: Scenario 1 with friction drop but no correction tire forces.

With friction coefficient correction within the PF

In this case the MPC-based PF is active and makes the difference: it effectively generates inputs that can keep the car on the paths designed by the PP. This is possible thanks to the PF road friction coefficient match with the actual one (0.3), which in turn influences the dynamic states computed at each time step. The lateral and longitudinal acceleration are well below the limits and change smoothly throughout the simulation. Solvers performances are also quite good since both MPCs converge almost always and the corresponding computation times are within the desired range. The corrected tire forces are visible in Figure 6.17. In this specific case there is a difference between values obtained through the linear model and the Dugoff model, that is enough to motivate its adoption.

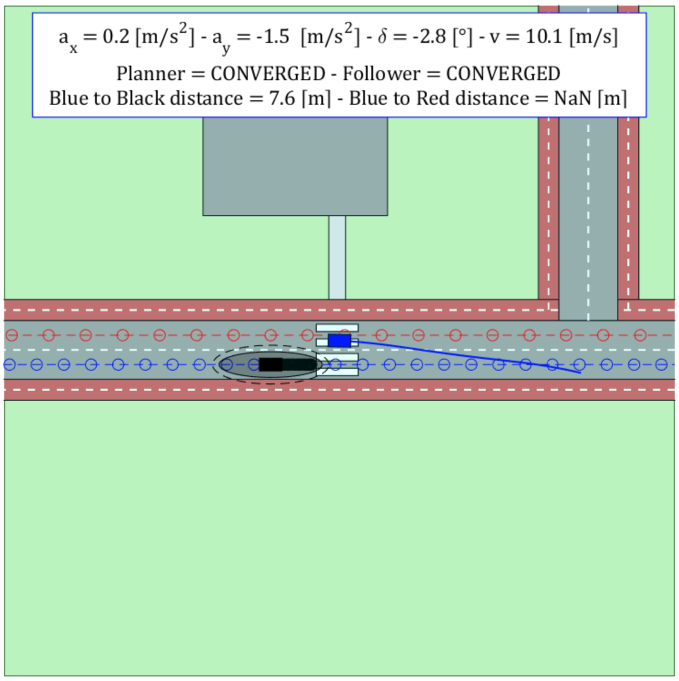


(a)

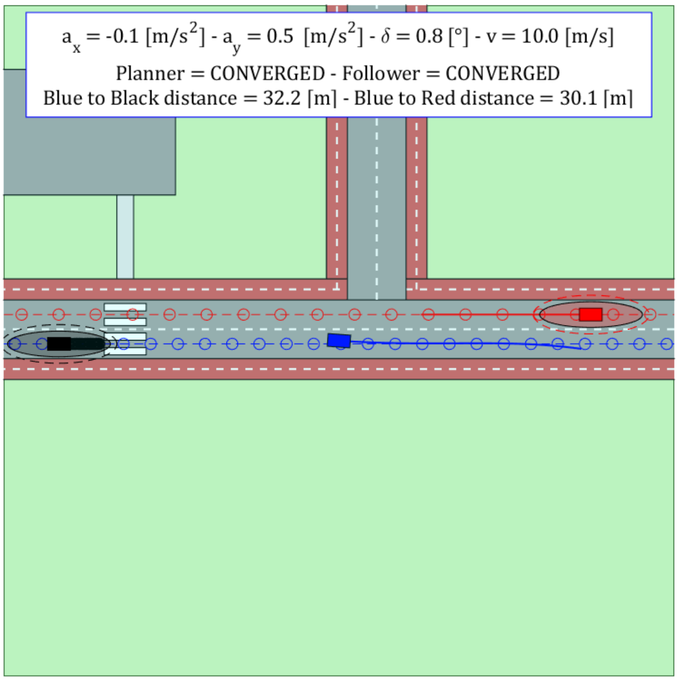


(b)

Figure 6.14: Scenario 1 with friction drop and correction frames.

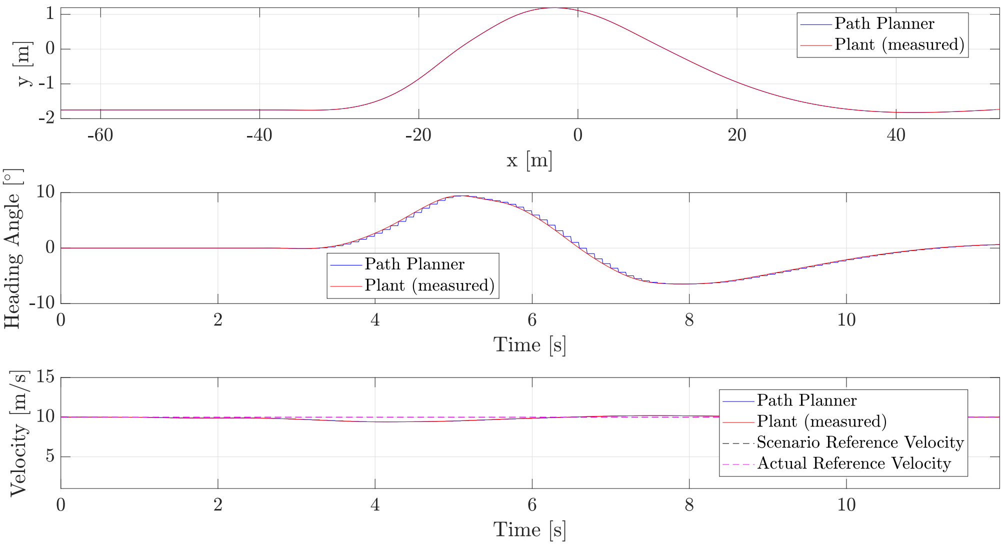


(c)

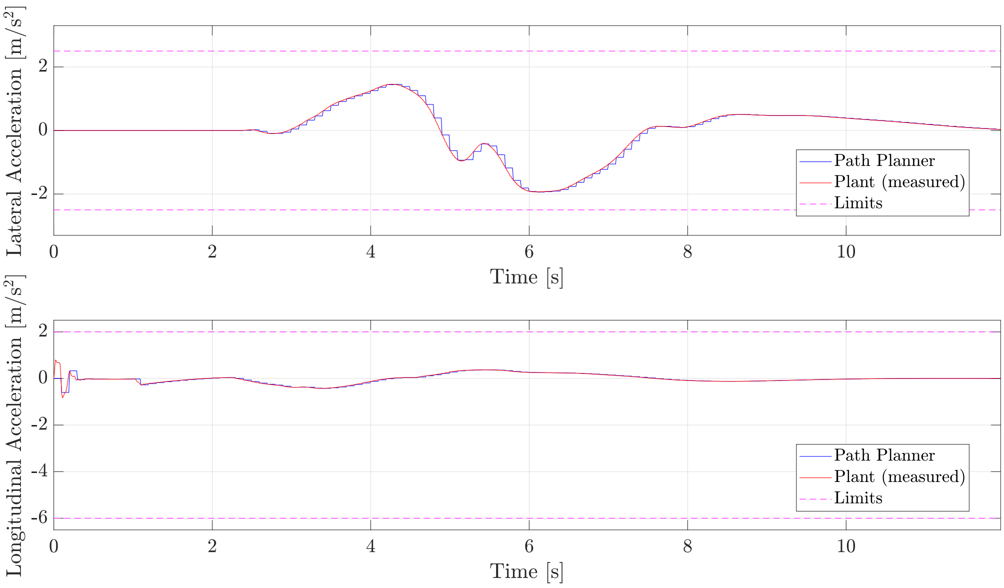


(d)

Figure 6.14: Scenario 1 with friction drop and correction frames.

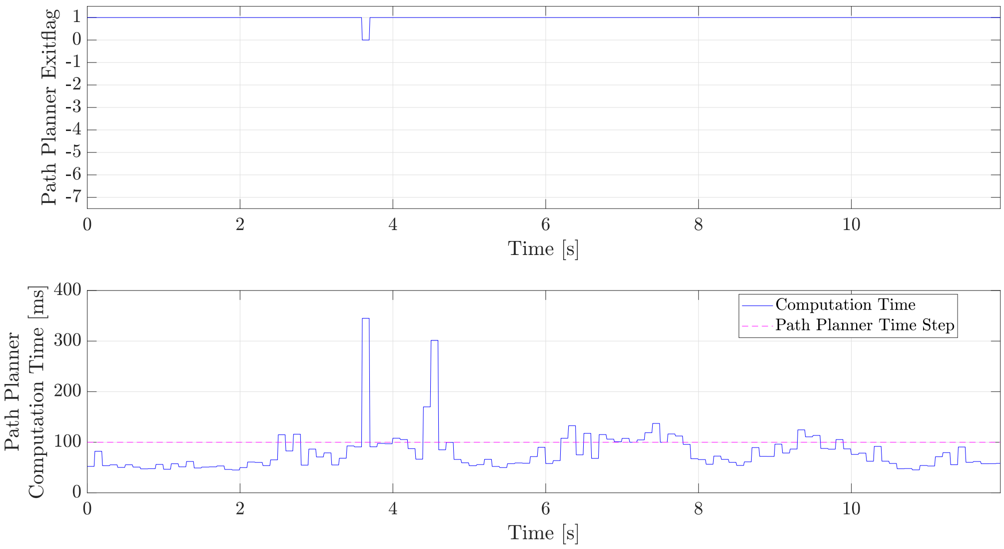


(a) Position, yaw and velocity of the controlled vehicle from both Path Planner (predicited) and Plant (measured).

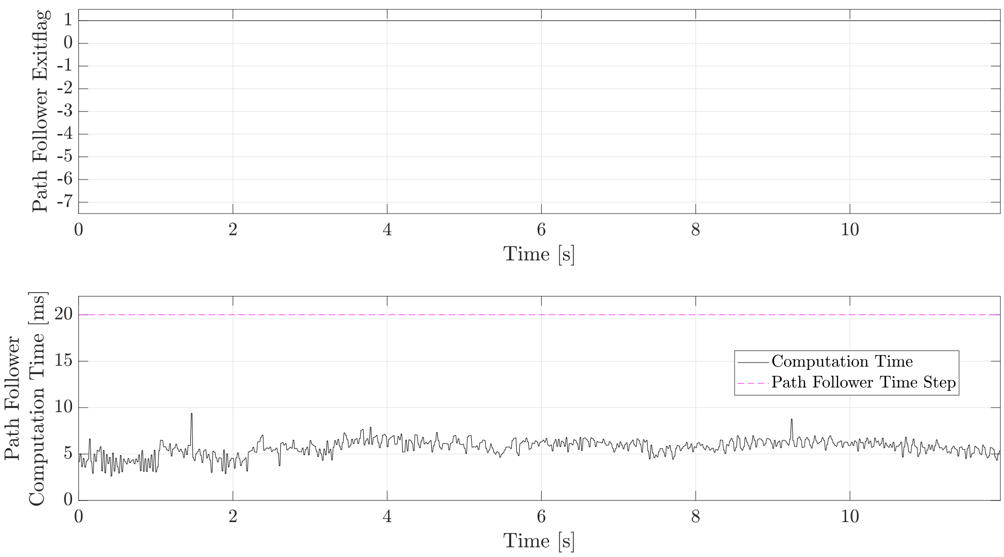


(b) Lateral and longitudinal acceleration of the controlled vehicle.

Figure 6.15: Scenario 1 with friction drop and correction simulation results.



(a) Path Planner optimization results.



(b) Path Follower optimization results.

Figure 6.16: Scenario 1 with friction drop and correction optimization processes results.

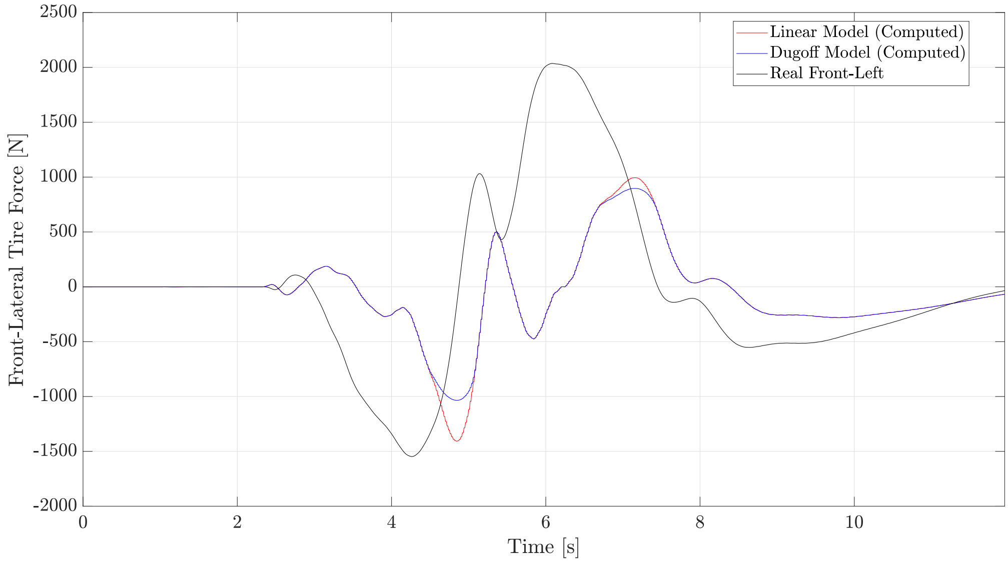


Figure 6.17: Scenario 1 with friction drop and correction tire forces.

6.2 Scenario 2 - Road Intersection

In this scenario the controlled vehicle faces another agent coming out from an intersection without reducing its velocity (something that should normally be done). When it is within the detection range and visible, its future position are estimated to be right in front of it due to the prediction strategy adopted (see Section 5.4). This way of proceeding is just easier to implement but lacks of physical sense, since the detected car is expected to turn either left or right, and drastically enlarge the computational burden the PP has to deal with.

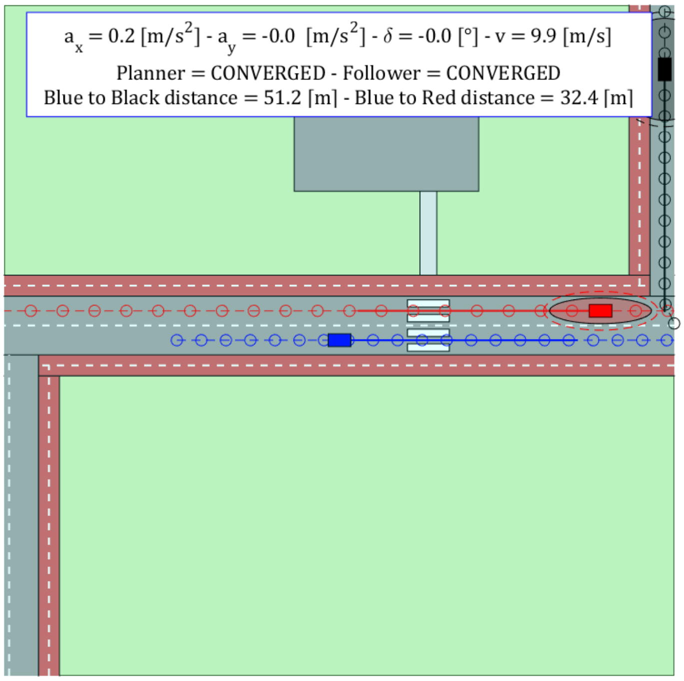
Firstly, the black agent is assumed to be visible throughout the whole simulation. Secondly, it is hidden for the first part to basically reproduce a blind corner. Thirdly, perfect visibility and V2V communication (theoretically the base for the best prediction strategy) are assumed to be present. As demonstrated, this last change can remarkably enhance performances and safety.

6.2.1 Complete Visibility

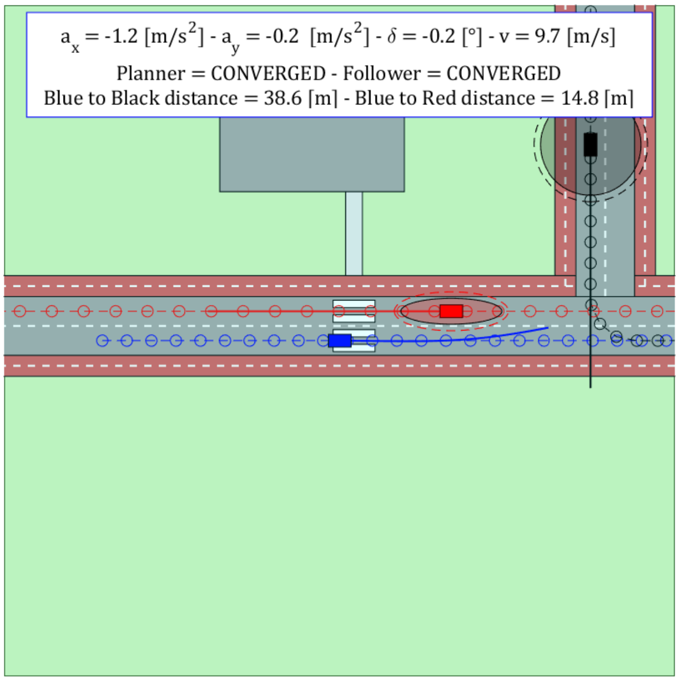
In this case the controlled agent can completely detect what is around itself, especially the black agent. In reality, this would mean in absence of things like bad weather conditions or obstacles right before the road intersection.

The designed LMP effectively handled the situation by slowing down and steering a little to make sure the moving obstacle can be avoided. In doing so it keeps the lateral acceleration within the desired range, while the longitudinal one matches the first need of reducing velocity and the second one of reaching the reference value.

Computationally, both solvers always converge despite a bit longer time span in two occasions. Overall, performances definitively outdo the desired target.

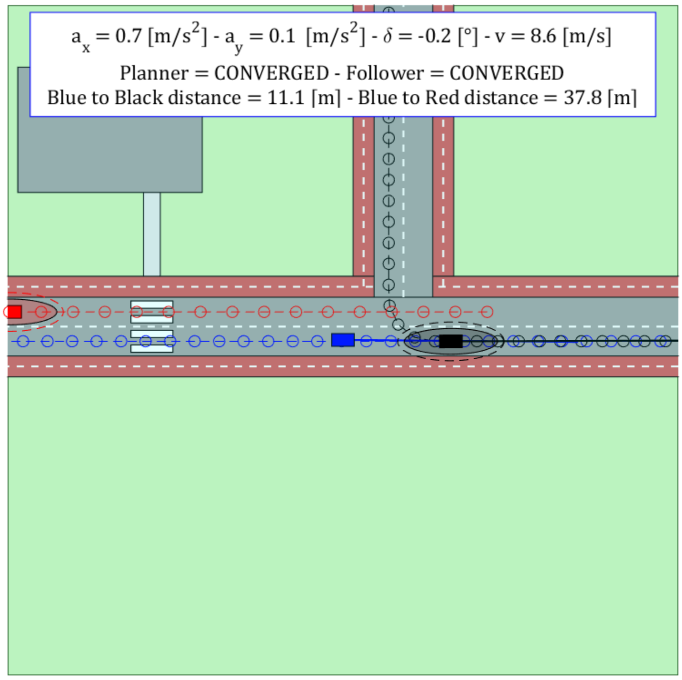


(a)

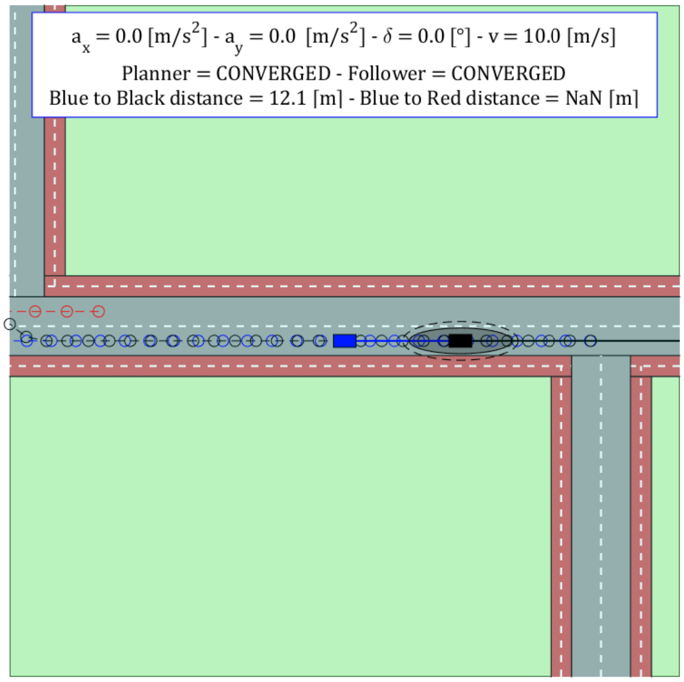


(b)

Figure 6.18: Scenario 2 frames.

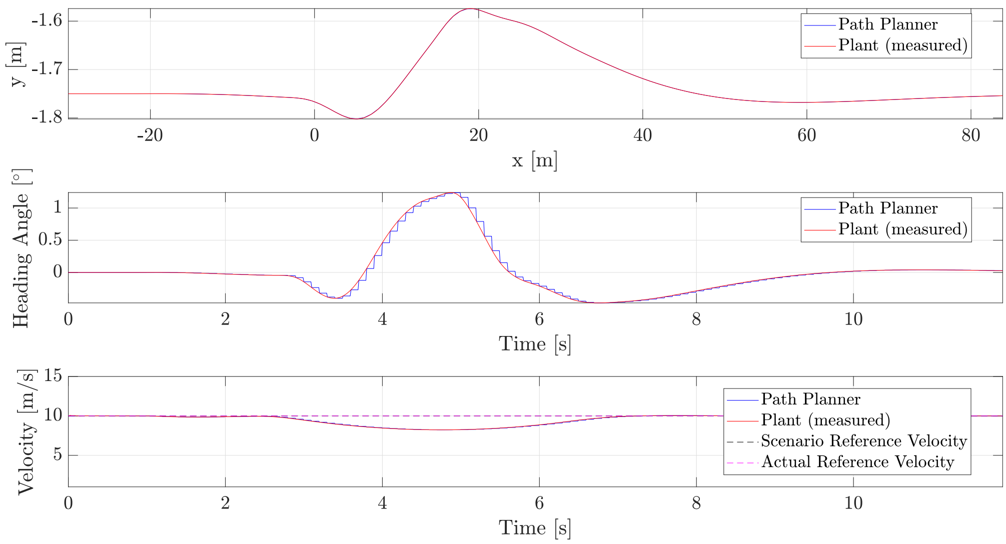


(c)

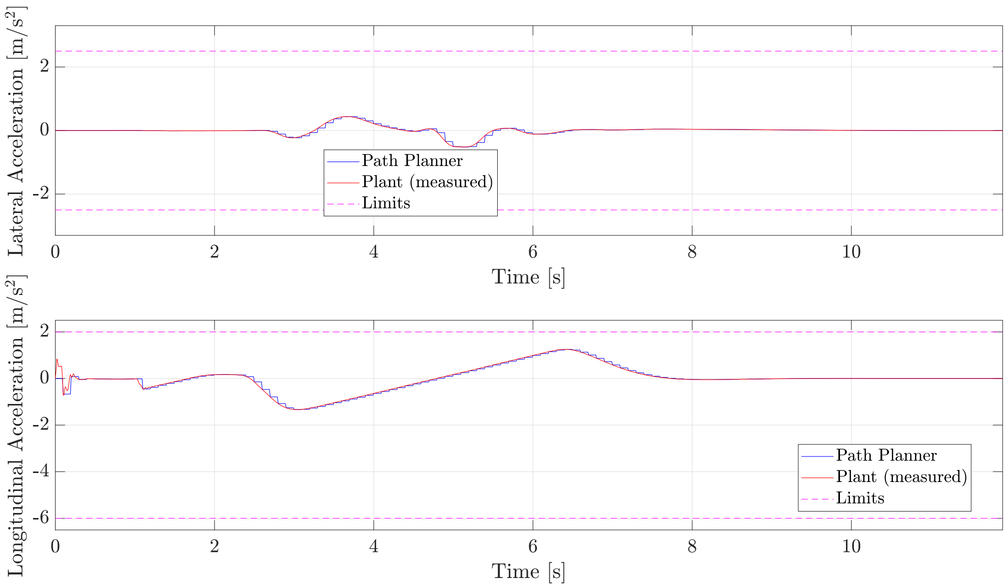


(d)

Figure 6.18: Scenario 2 frames.

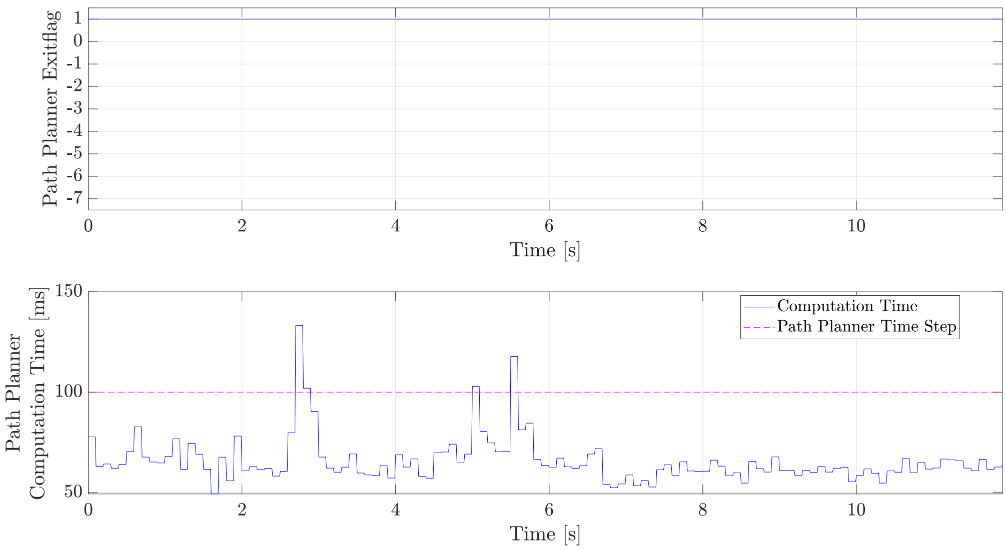


(a) Position, yaw and velocity of the controlled vehicle from both Path Planner (predicited) and Plant (measured).

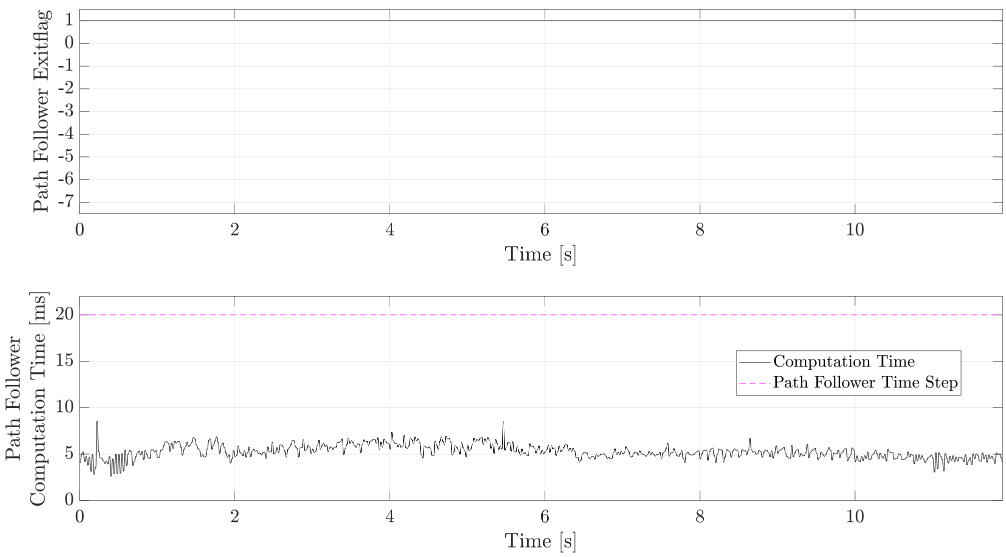


(b) Lateral and longitudinal acceleration of the controlled vehicle.

Figure 6.19: Scenario 2 simulation results.



(a) Path Planner optimization results.



(b) Path Follower optimization results.

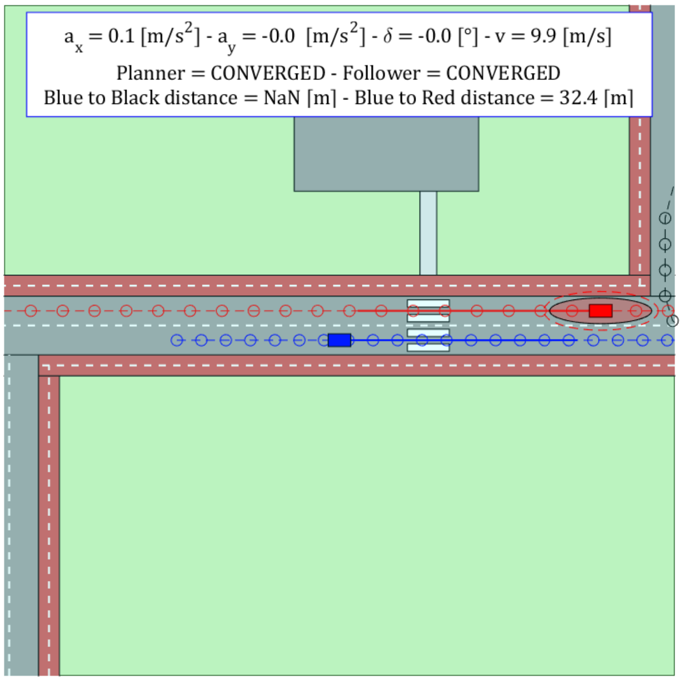
Figure 6.20: Scenario 2 optimization processes results.

6.2.2 Blind Spot

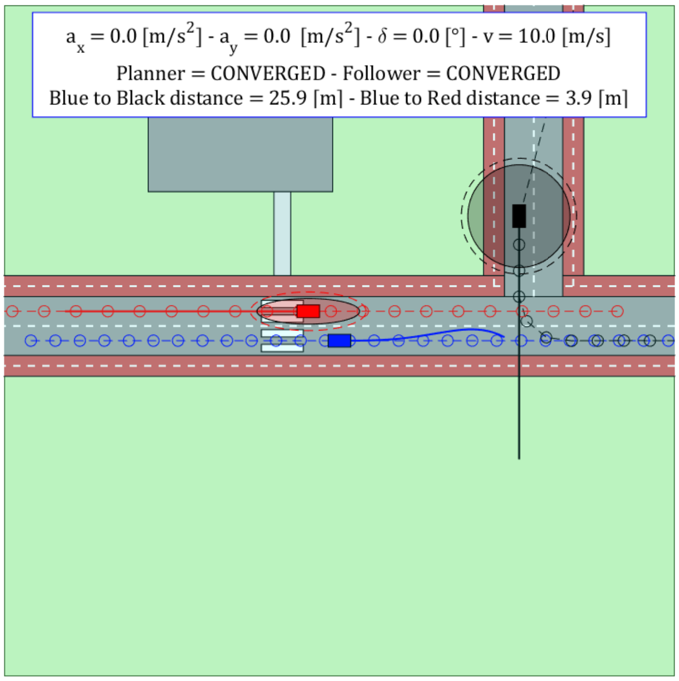
In this case, the black agent is made invisible until it gets really close to the intersection to reproduce a blind spot and push the LMP to a situation hard to handle.

In order to effectively complete this simulation the high-fidelity plant can not be used because the nonlinearities generated would lead to the PP failure. This proves how the designed LPM is effective in urban scenarios but struggles in emergency maneuver. However, better prediction strategies, perhaps involving agents communication (as shown in the next simulation), can definitively reduce the gap. The safety features described in Section 2.2.2 are active for the whole simulations despite not being triggered.

Although the almost last minute detection, the controlled agent manages to avoid a crash by steering and decelerating more than what it did in the previous test. Consequently, lateral and longitudinal accelerations reach higher values but still within the given ranges. Both solvers perform quite well since they mostly converged in short time spans.

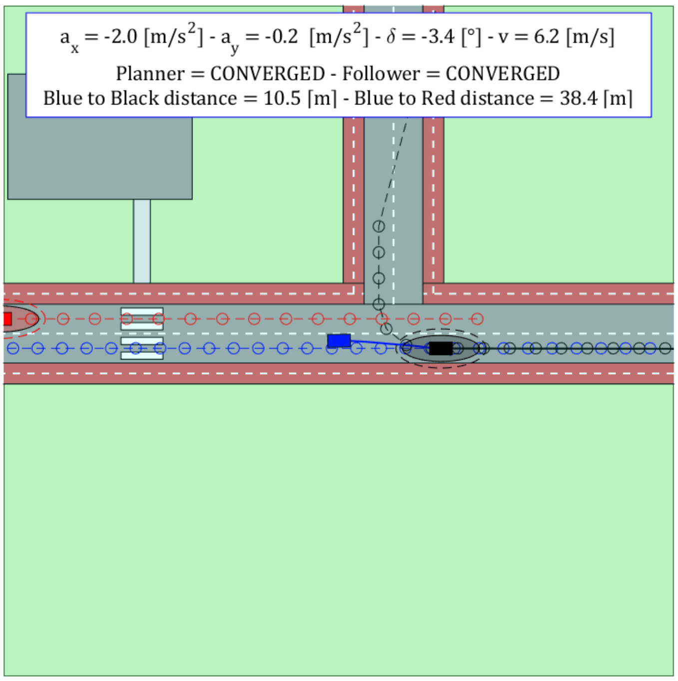


(a)

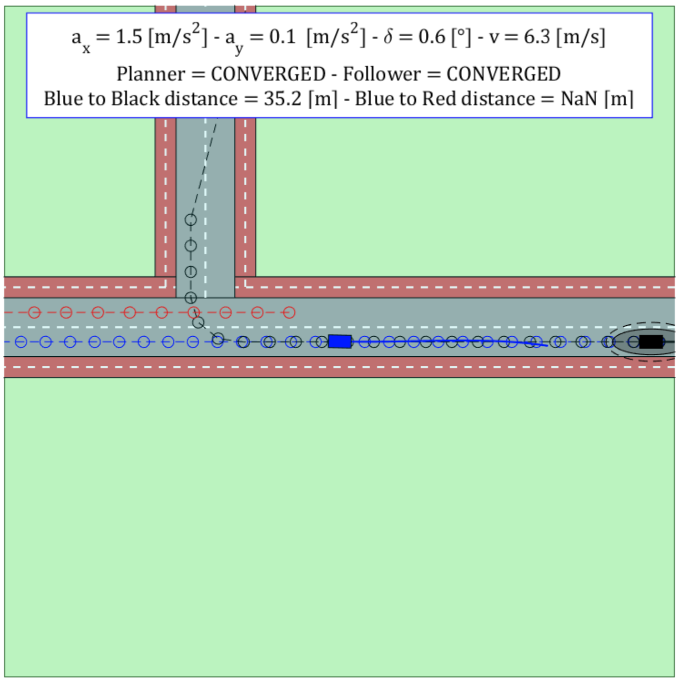


(b)

Figure 6.21: Scenario 2 with hidden obstacle frames.

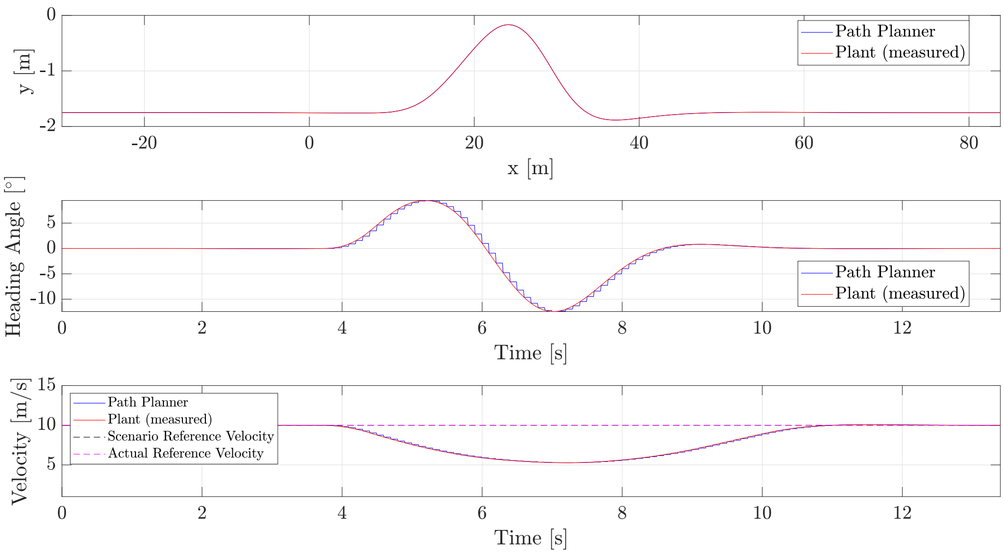


(c)

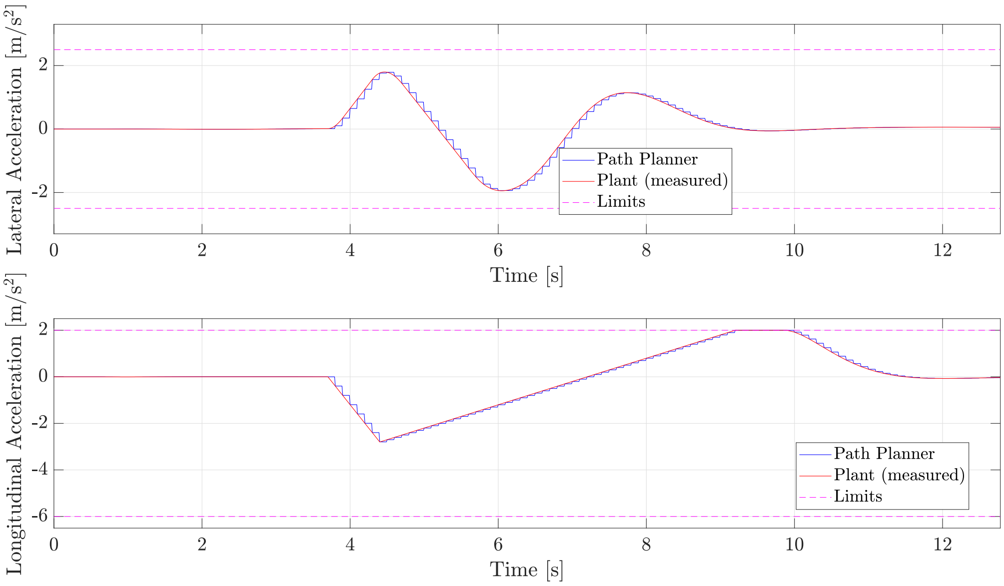


(d)

Figure 6.21: Scenario 2 with hidden obstacle frames.

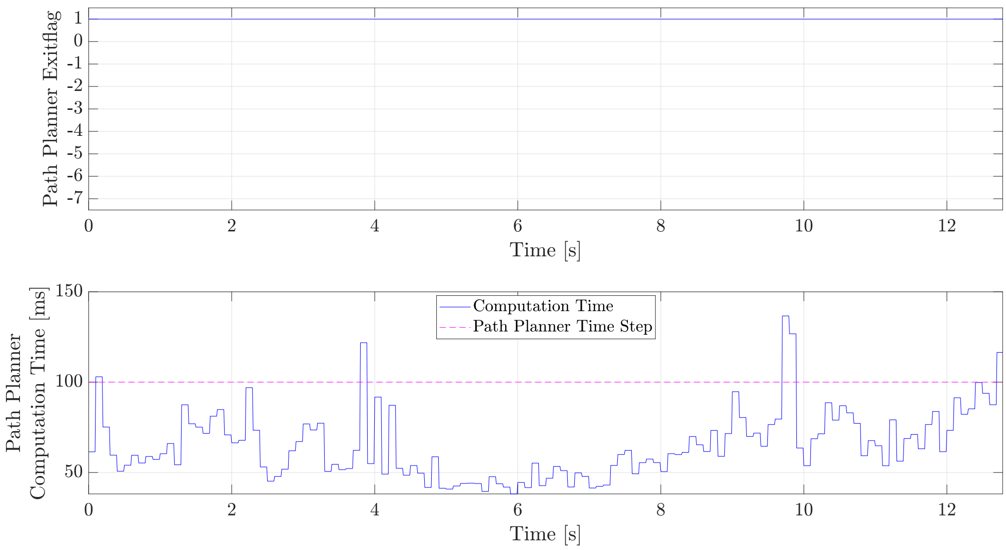


(a) Position, yaw and velocity of the controlled vehicle from both Path Planner (predicited) and Plant (measured).

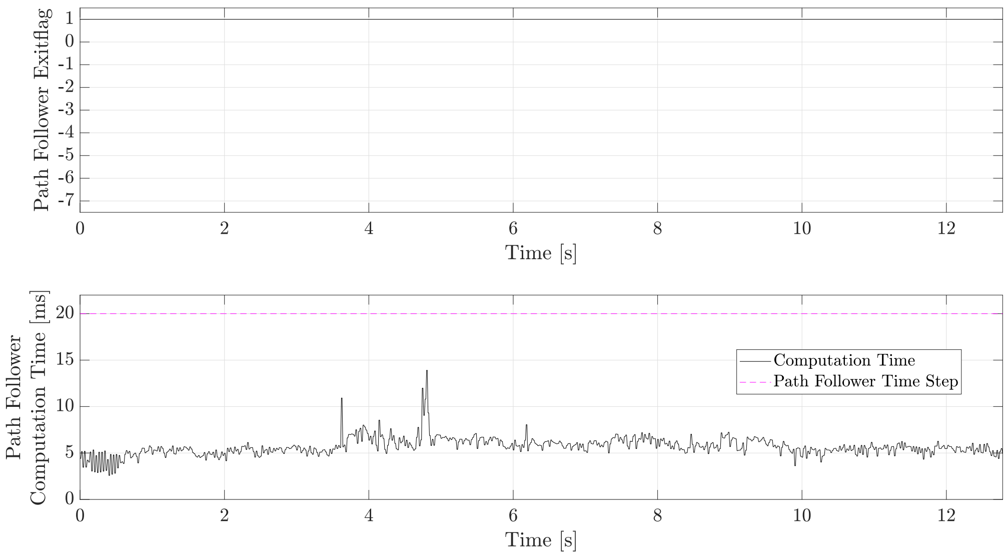


(b) Lateral and longitudinal acceleration of the controlled vehicle.

Figure 6.22: Scenario 2 with hidden obstacle simulation results.



(a) Path Planner optimization results.

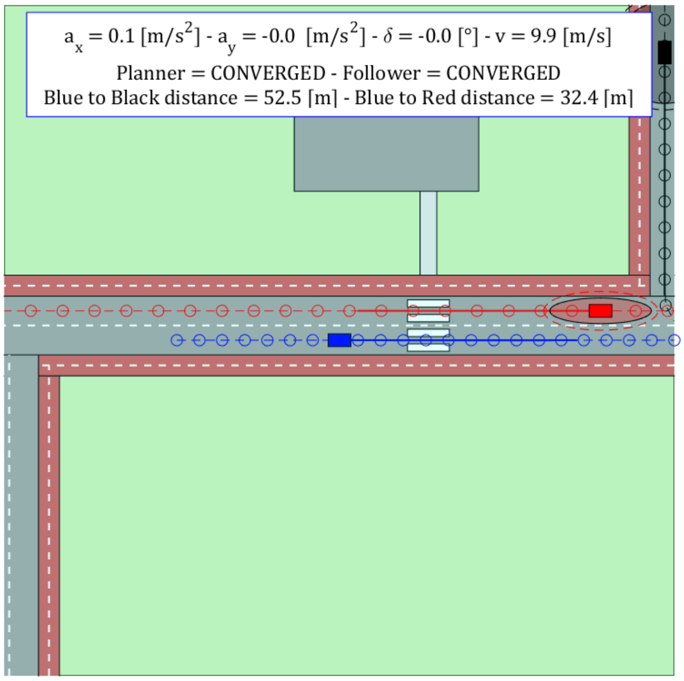


(b) Path Follower optimization results.

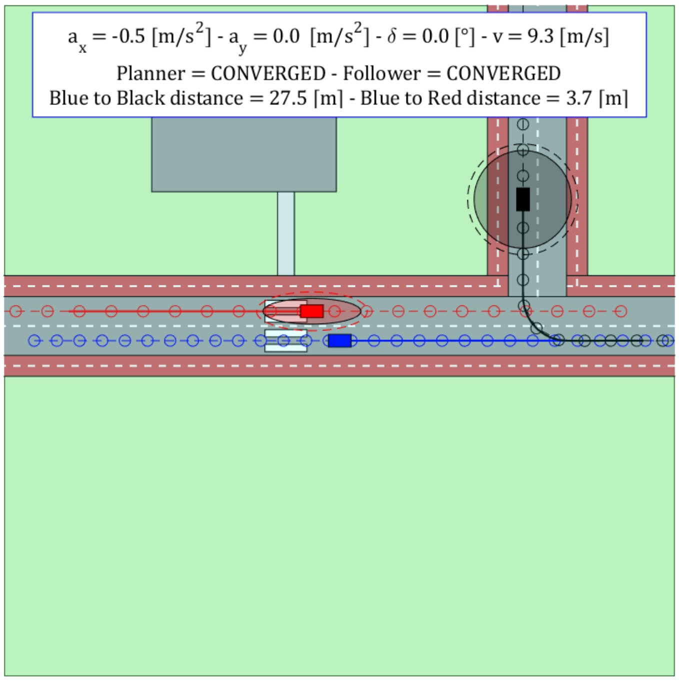
Figure 6.23: Scenario 2 with hidden obstacle optimization processes results.

6.2.3 With V2V Communication

In this case V2V communication is assumed to be present. This means that all agents of the problem know where the others are and will be in future. This change remarkably improves performances: according to the chosen weighting factors in the objective function, the controlled vehicle slows done enough to find itself at a safety distance once close to the black one, while steering the minimum possible. Accelerations graphs show that while computational performances are very good except for an instant that required a few more iterations.

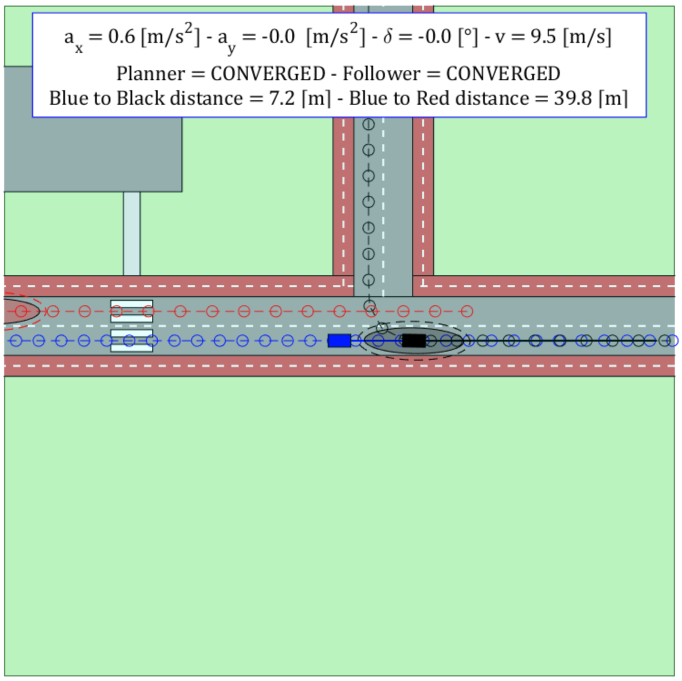


(a)

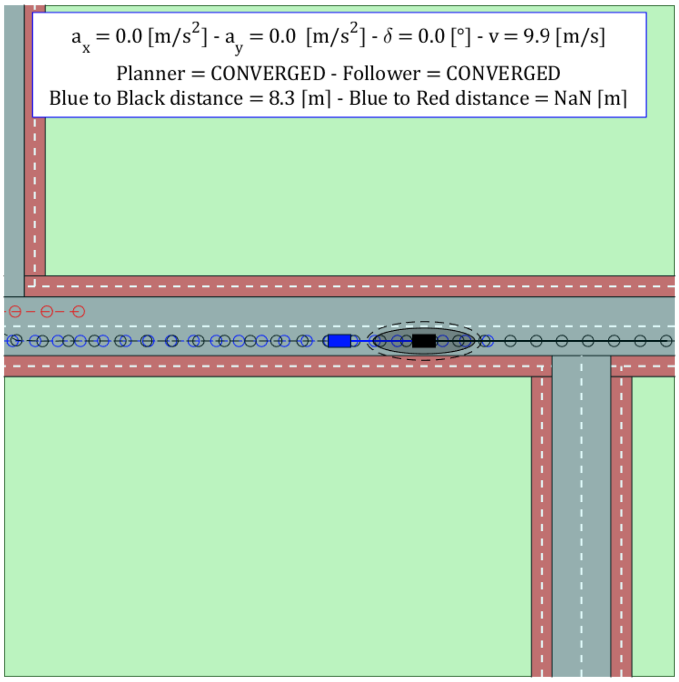


(b)

Figure 6.24: Scenario 2 with hidden obstacle and V2V communication frames.

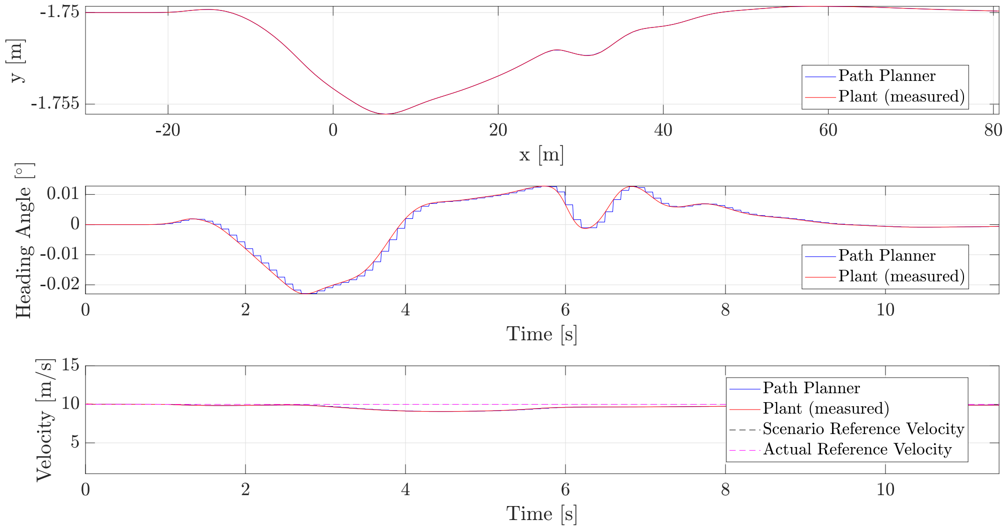


(c)

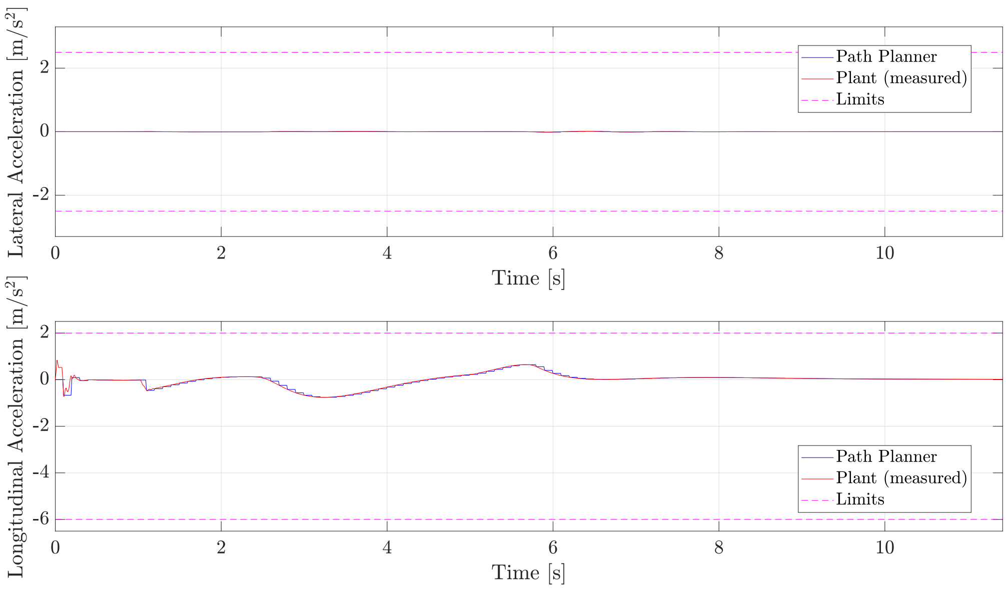


(d)

Figure 6.24: Scenario 2 with hidden obstacle and V2V communication frames.

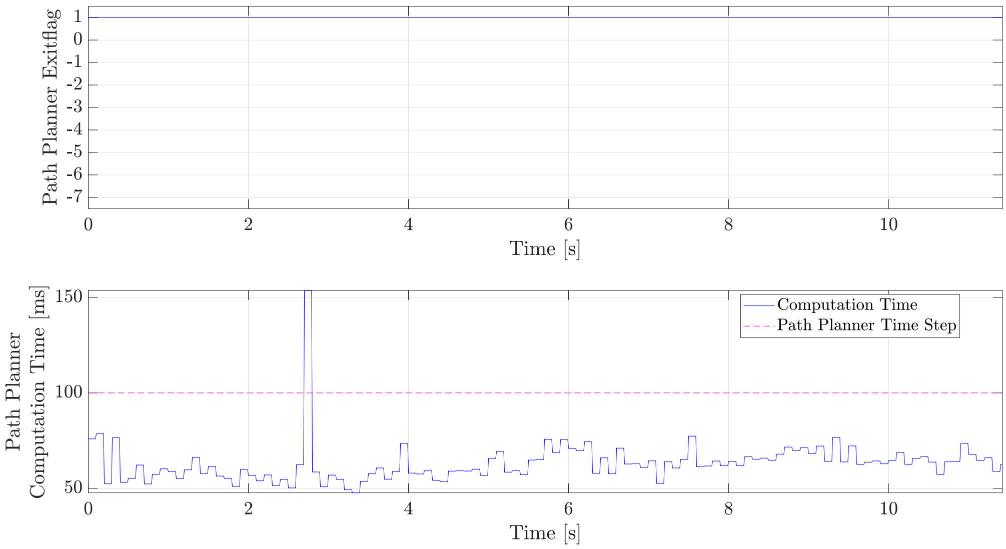


(a) Position, yaw and velocity of the controlled vehicle from both Path Planner (predicited) and Plant (measured).

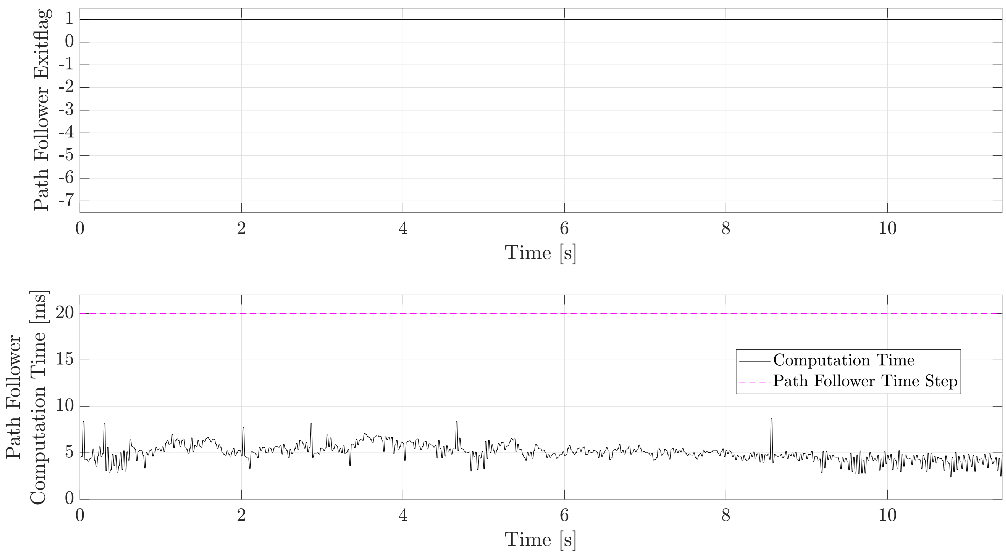


(b) Lateral and longitudinal acceleration of the controlled vehicle.

Figure 6.25: Scenario 2 with hidden obstacle and V2V communication simulation results.



(a) Path Planner optimization results.



(b) Path Follower optimization results.

Figure 6.26: Scenario 2 with hidden obstacle and V2V communication optimization processes results.

6.3 Scenario 3 - Lane Invasion

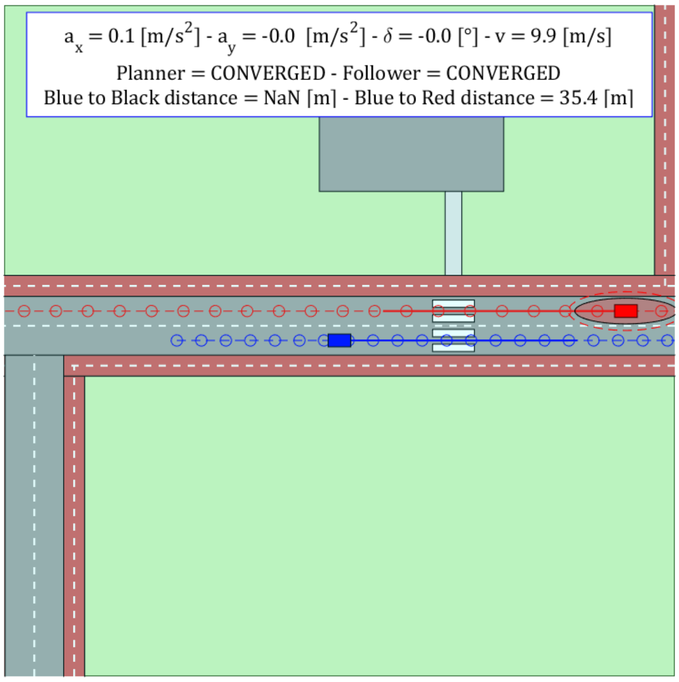
In this scenario the controlled vehicle faces another car coming in opposite direction that invades the lane to then turn to the right.

Firstly, the simulation is carried with the easiest strategy to predict other agents future positions and active safety features. Secondly, V2V communication is assumed to be available and safety features are active but not triggered.

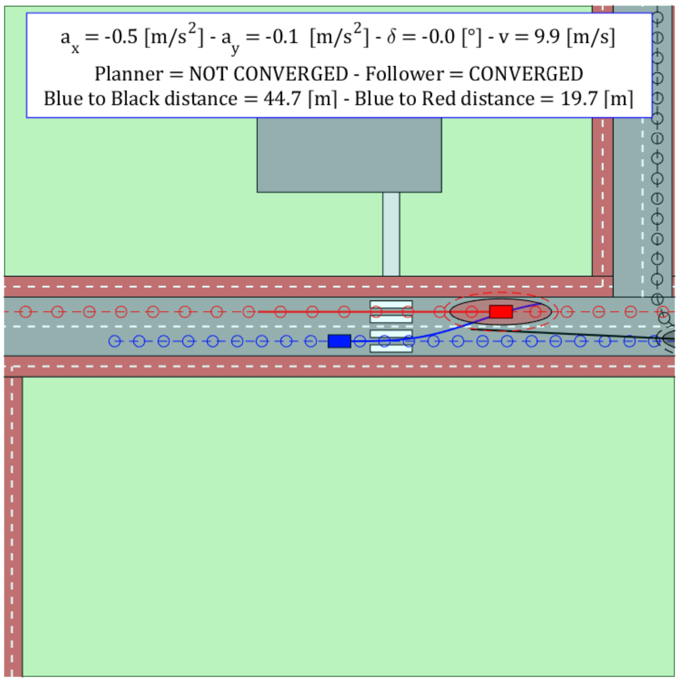
6.3.1 Without V2V Communication

In this case, the second movement prediction strategy (see Section 5.4) is adopted. Due to this, the mathematical problem that the PP has to solve when the blue and black vehicles are directed exactly one towards the other has no feasible solution, and this destabilizes the LMP (as proved by the exitflags). However, thanks to the safety actions that triggered straight away, it can retrieve control and the simulation is completed.

The safety features activation strongly slows down the vehicle while the orientation remains almost constant. This causes a sudden decreases of longitudinal acceleration while the lateral one is not heavily impacted. In terms of computational performances, as shown in Figure 6.29 the PP and the PF fail around 3 s but work really well for the rest of the simulation.

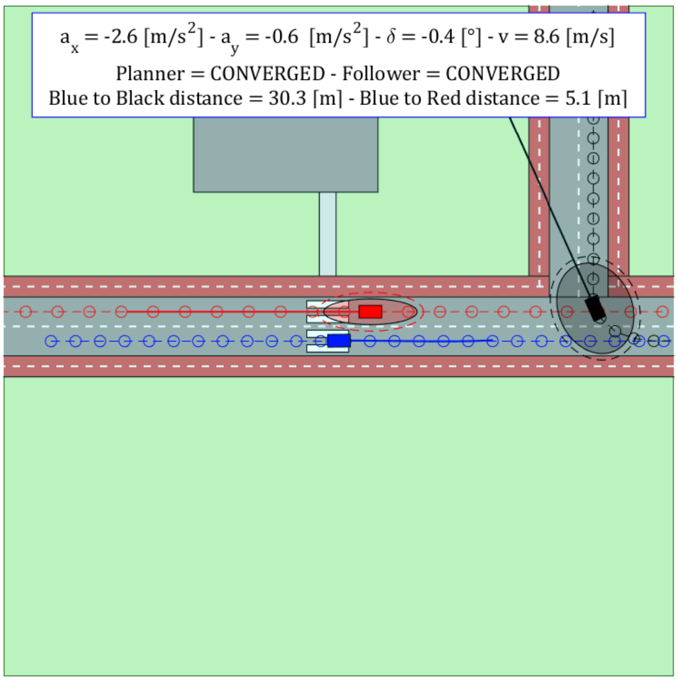


(a)

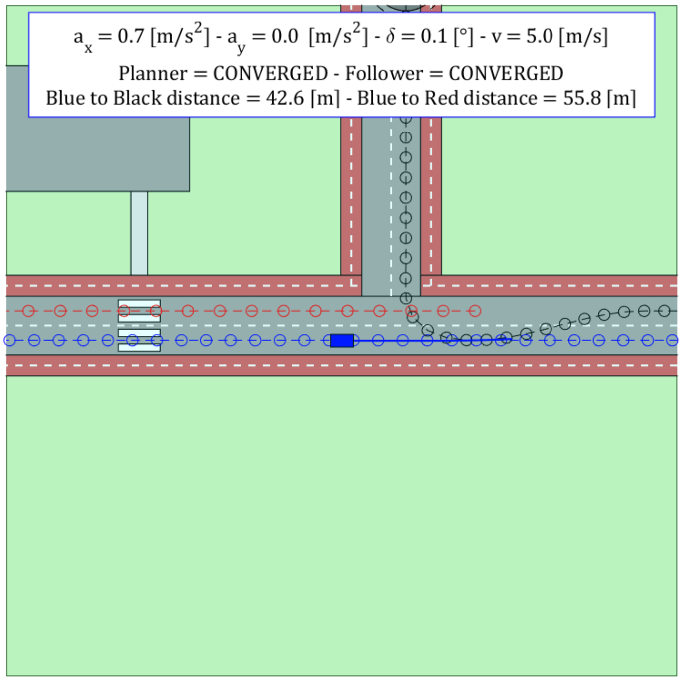


(b)

Figure 6.27: Scenario 3 frames.

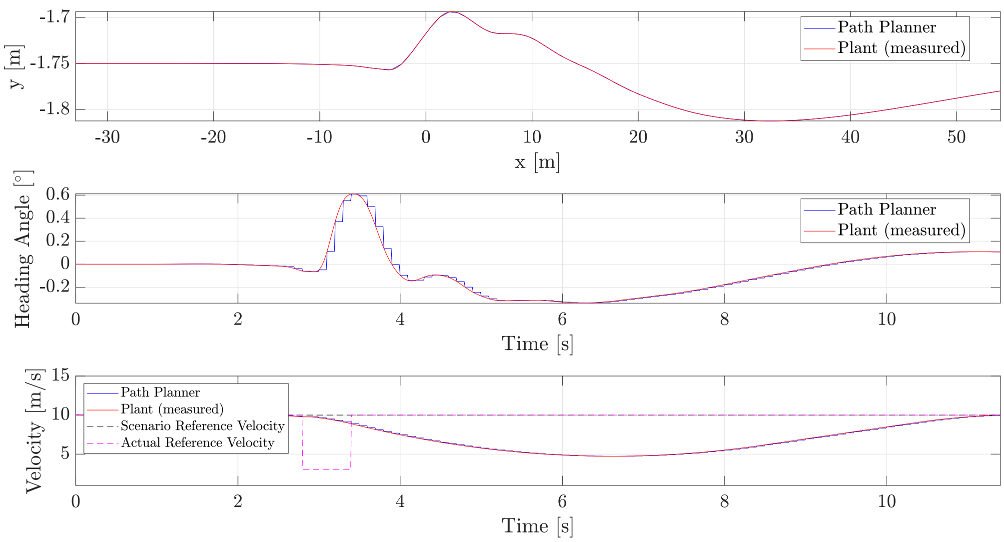


(c)

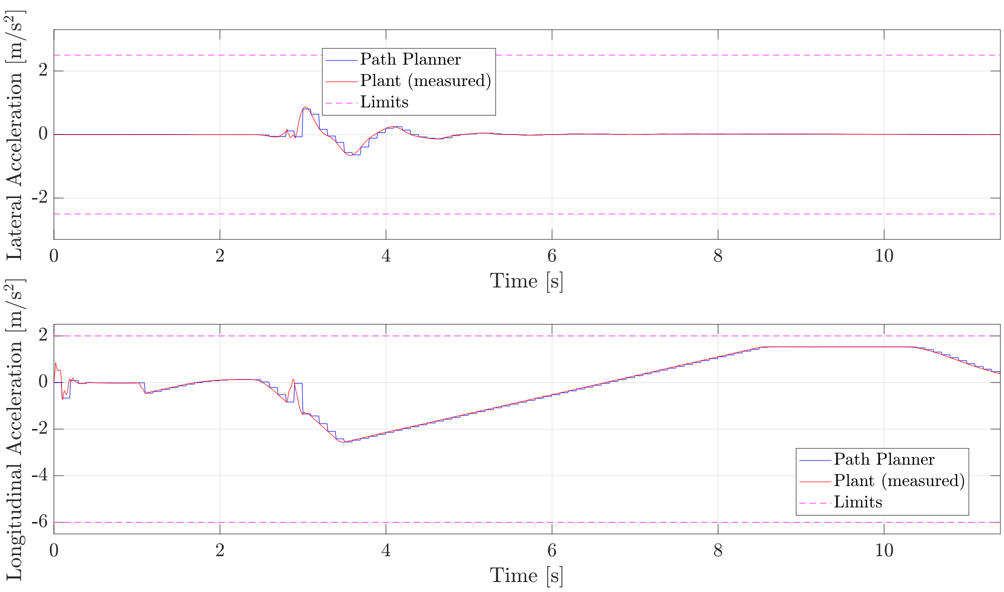


(d)

Figure 6.27: Scenario 3 frames.

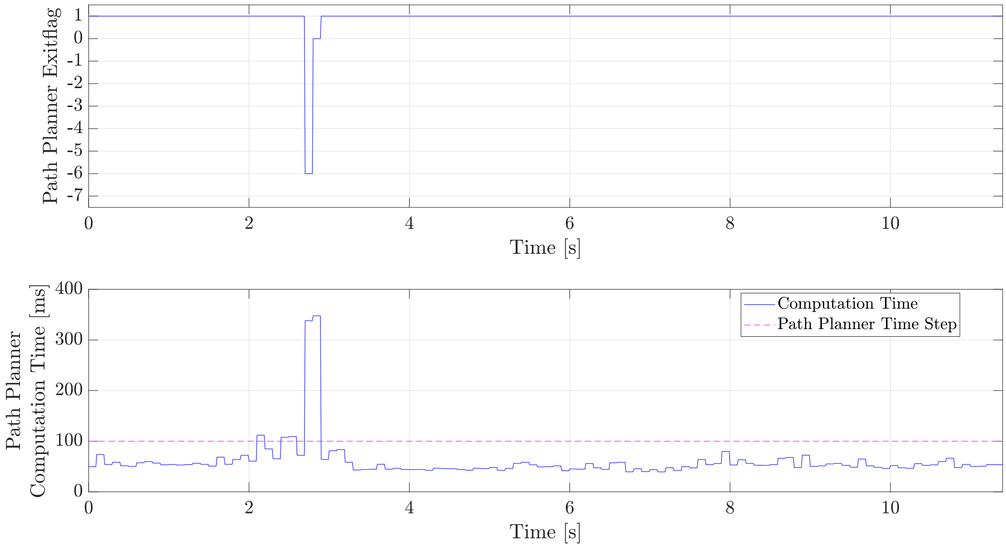


(a) Position, yaw and velocity of the controlled vehicle from both Path Planner (predicited) and Plant (measured).

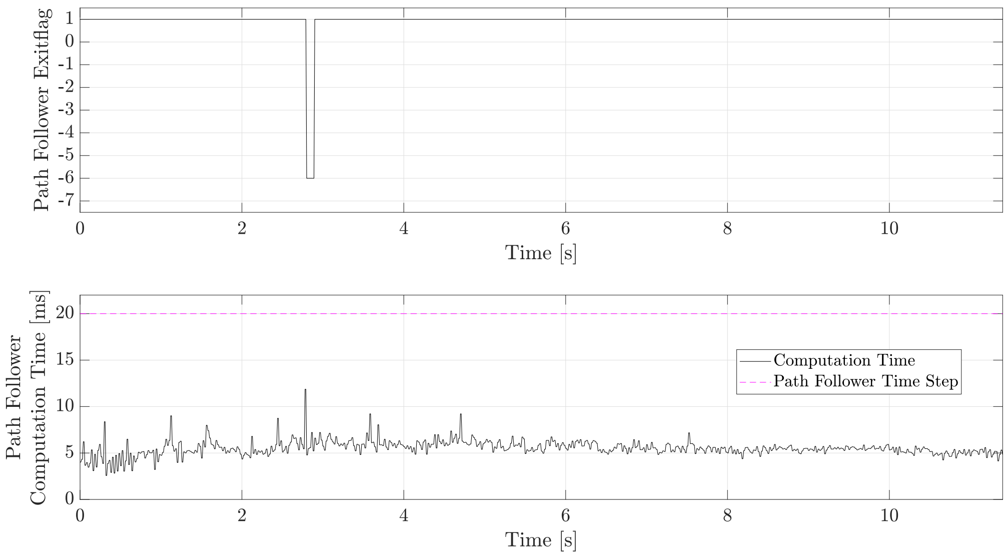


(b) Lateral and longitudinal acceleration of the controlled vehicle.

Figure 6.28: Scenario 3 simulation results.



(a) Path Planner optimization results.



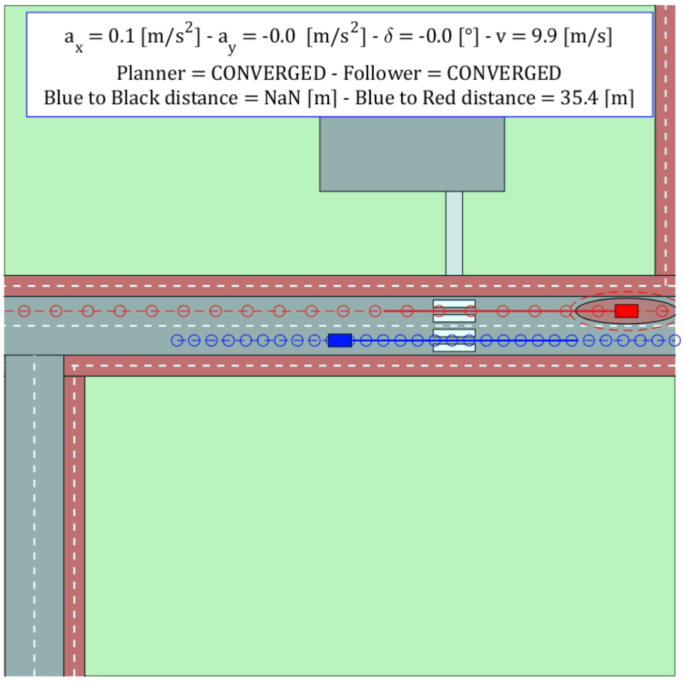
(b) Path Follower optimization results.

Figure 6.29: Scenario 3 optimization processes results.

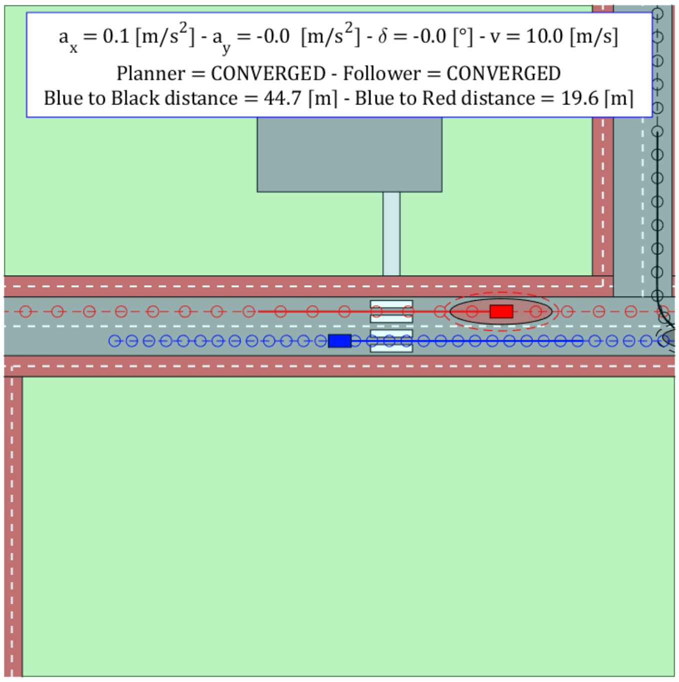
6.3.2 With V2V Communication

In this case V2V communication is available therefore all agents know each other's future positions.

The controlled car runs smoothly almost without steering but just reducing the velocity as necessary. This of course has a positive impact on passengers comfort as proved by the lateral acceleration profile. Computational performances are well within the defined range for both PP and PF.

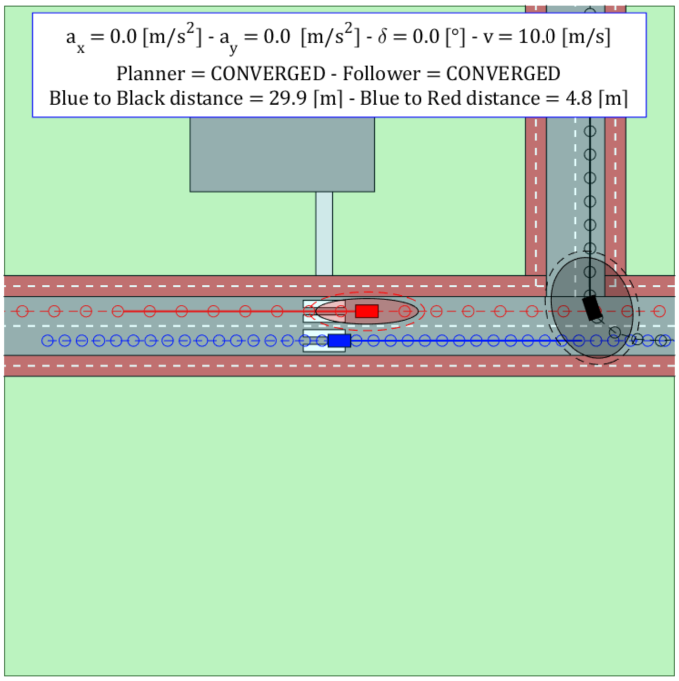


(a)

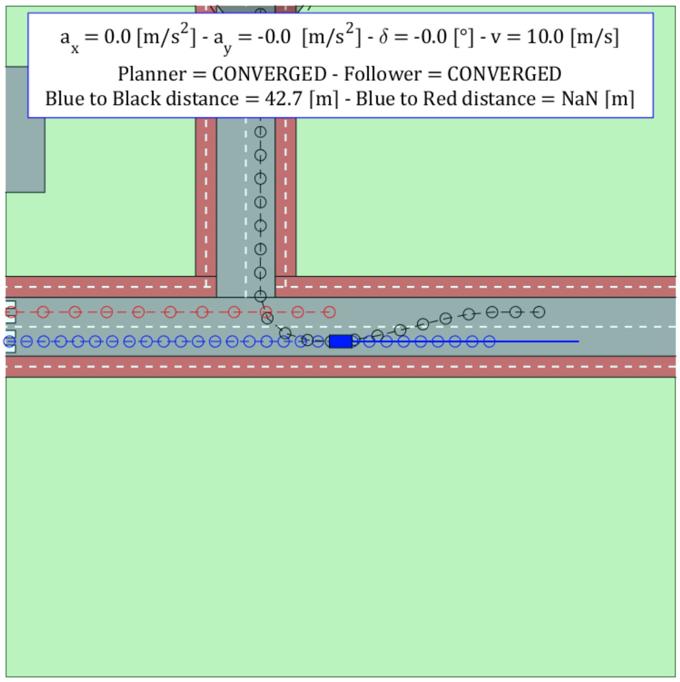


(b)

Figure 6.30: Scenario 3 with V2V communication frames.

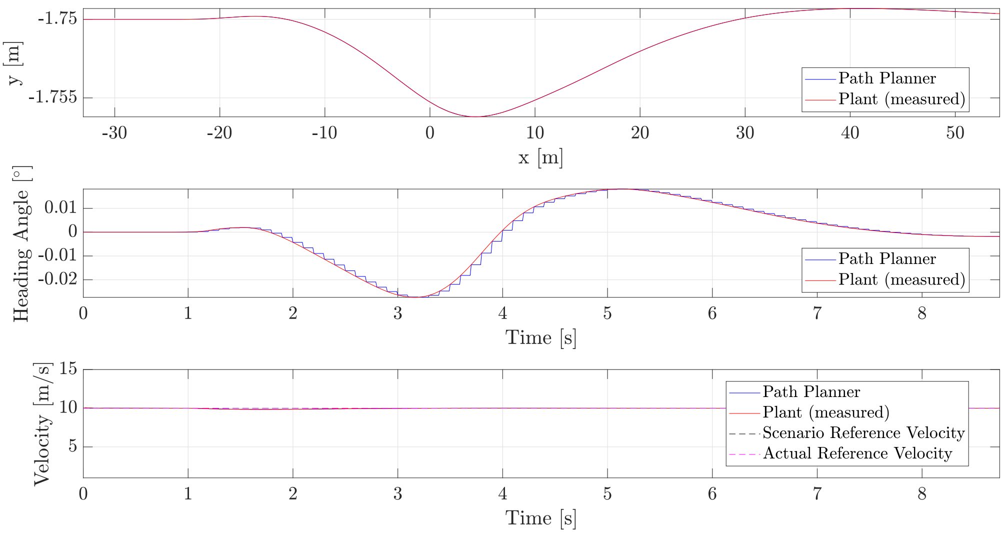


(c)

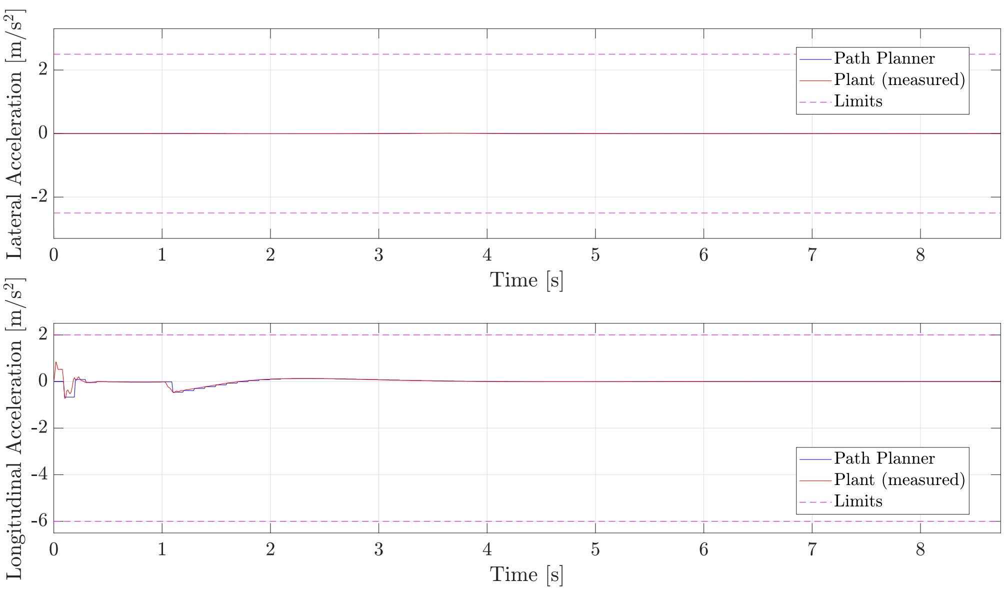


(d)

Figure 6.30: Scenario 3 with V2V communication frames.

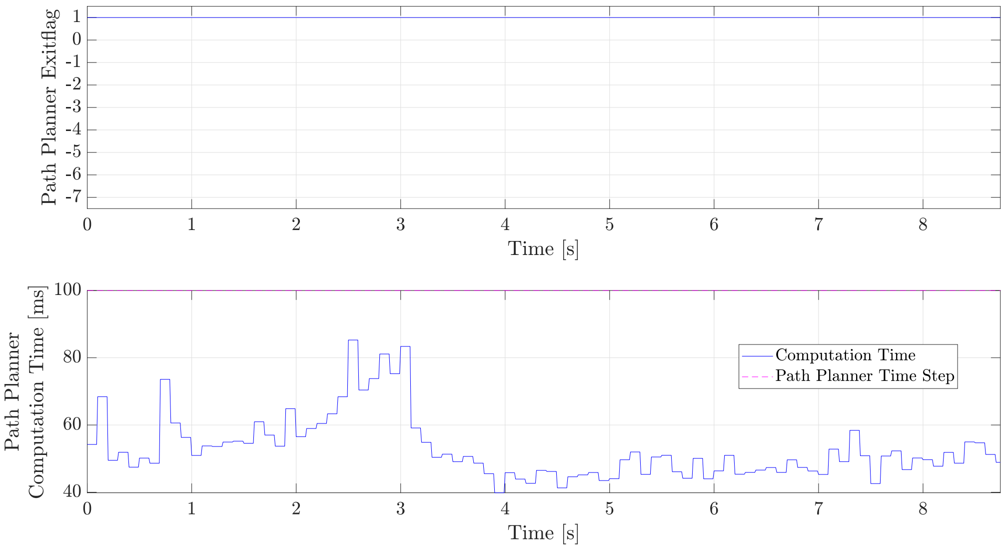


(a) Position, yaw and velocity of the controlled vehicle from both Path Planner (predicited) and Plant (measured).

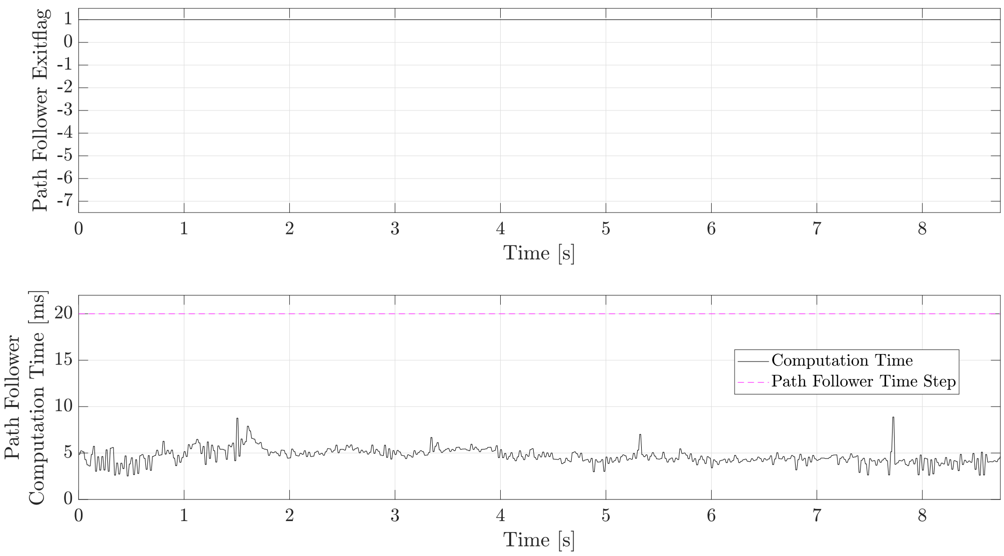


(b) Lateral and longitudinal acceleration of the controlled vehicle.

Figure 6.31: Scenario 3 with V2V communication simulation results.



(a) Path Planner optimization results.



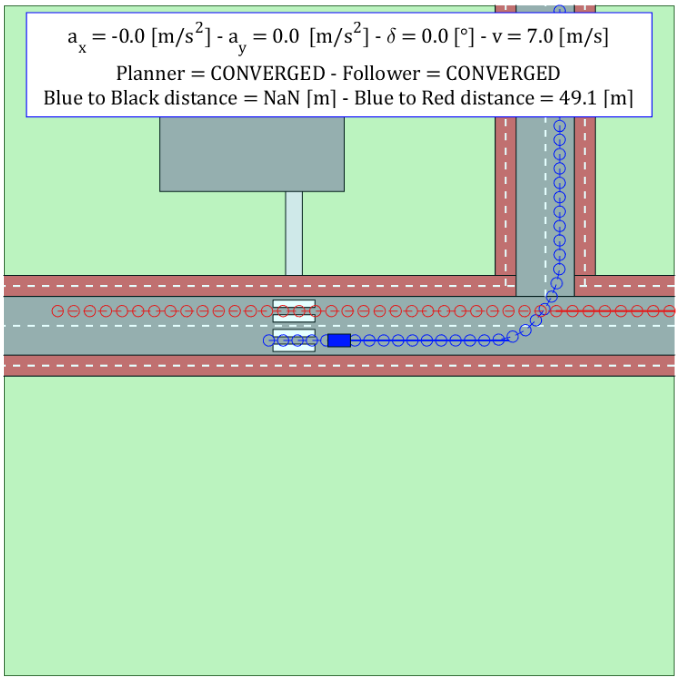
(b) Path Follower optimization results.

Figure 6.32: Scenario 3 with V2V communication optimization processes results.

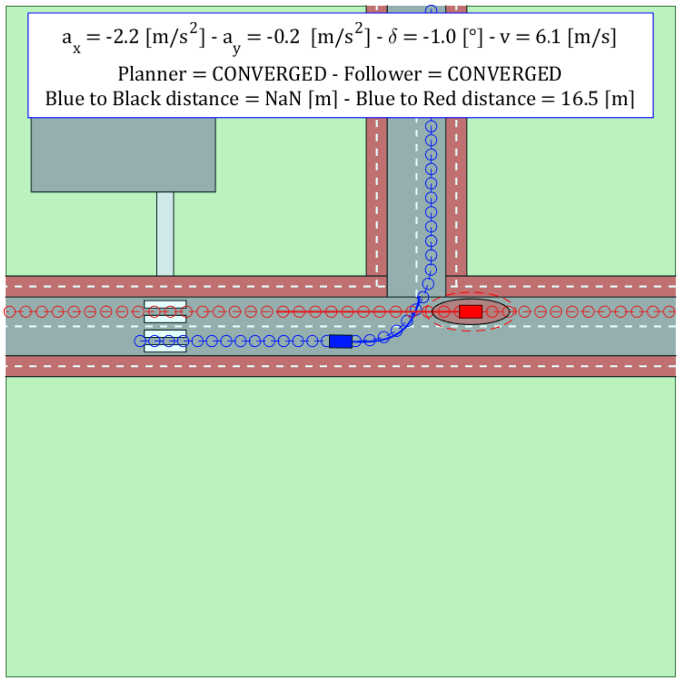
6.4 Scenario 4 - Sharp Turn

In this scenario the controlled vehicle has to turn to the left once the red agent has passed and, to do so, it follows a tight curve. Due to the fact that the velocity reaches small values the High Fidelity plant can not be used, therefore this simulation has been carried out with the dynamic plant. However, results should not be really different if the high-fidelity plant could have been used as the operating conditions are not "extreme".

The vehicle runs smoothly by reducing the velocity as necessary and does not show any kind of jerky behavior. Both the PP and the PF solvers always converge and mostly in reasonable time spans.

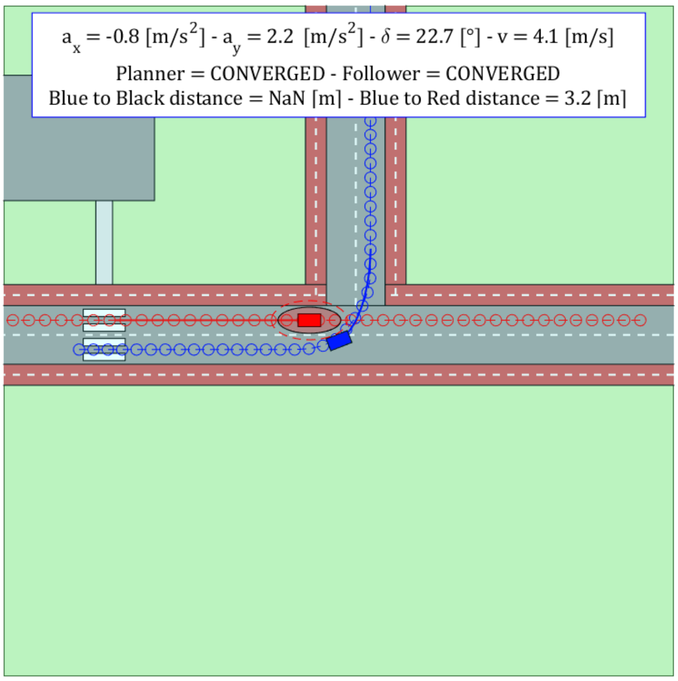


(a)

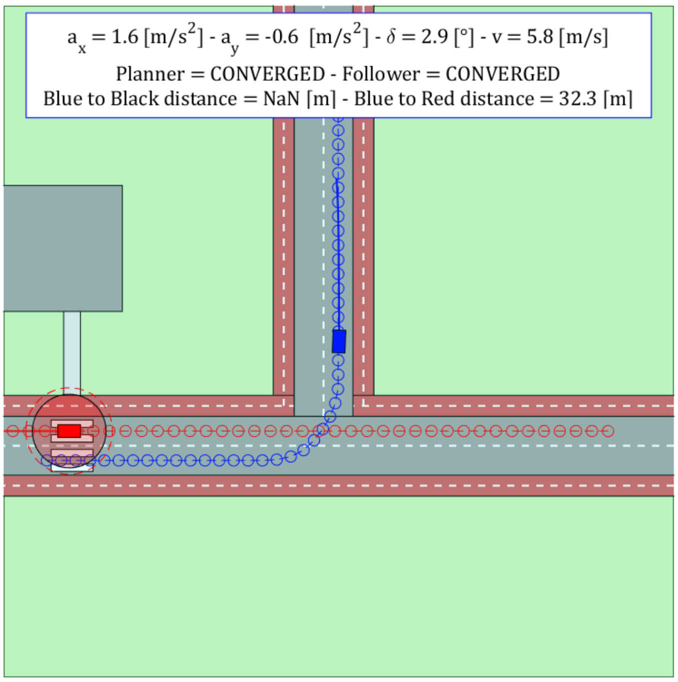


(b)

Figure 6.33: Scenario 4 frames.

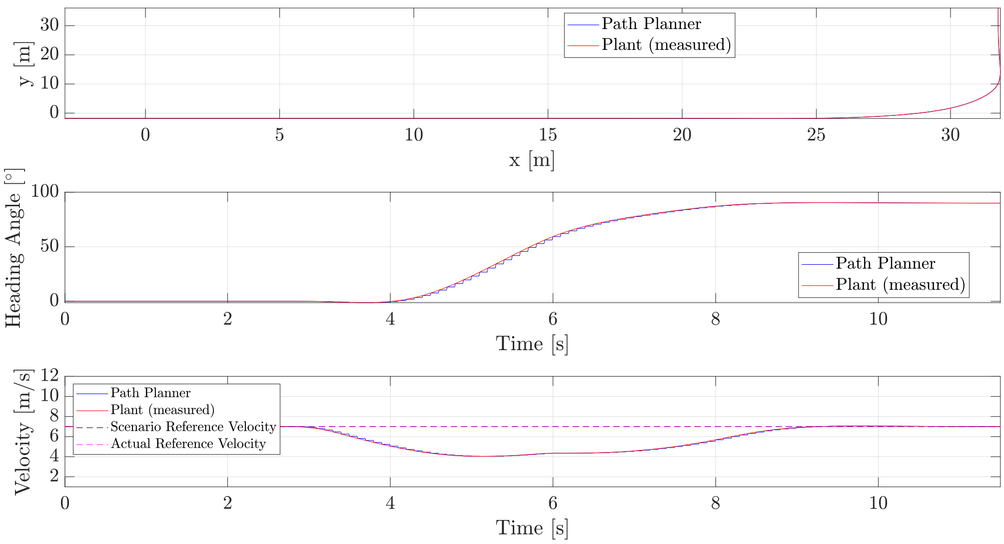


(c)

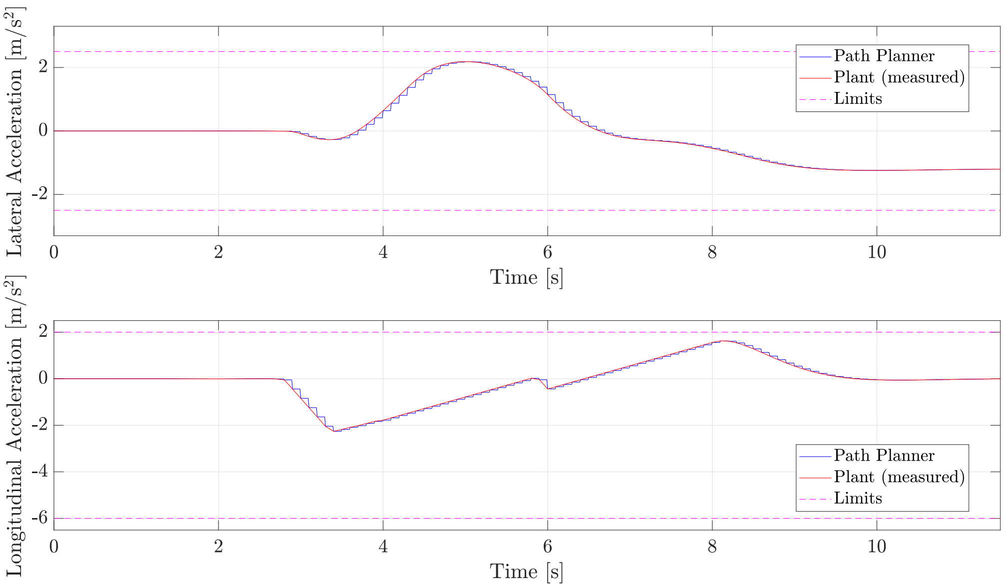


(d)

Figure 6.33: Scenario 4 frames.

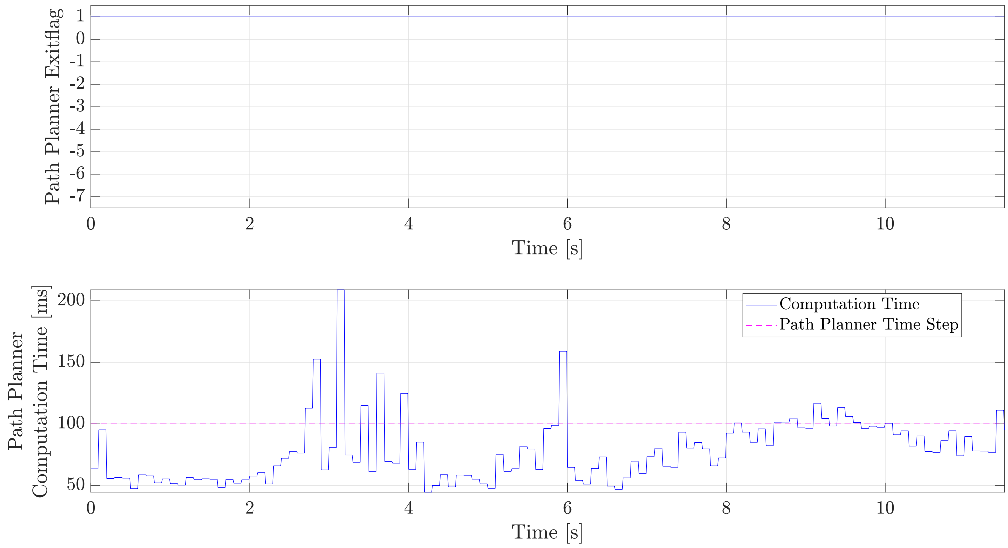


(a) Position, yaw and velocity of the controlled vehicle from both Path Planner (predicited) and Plant (measured).

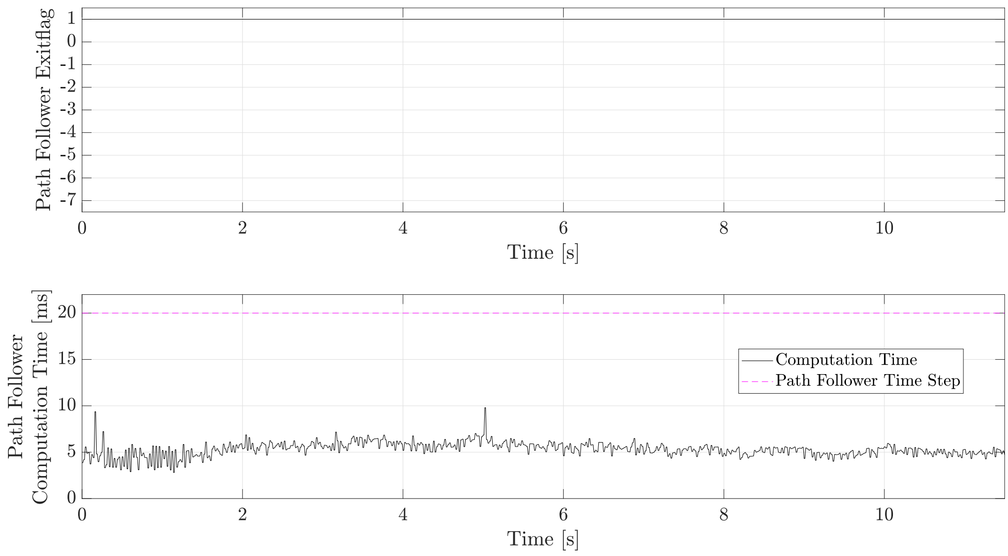


(b) Lateral and longitudinal acceleration of the controlled vehicle.

Figure 6.34: Scenario 4 simulation results.



(a) Path Planner optimization results.



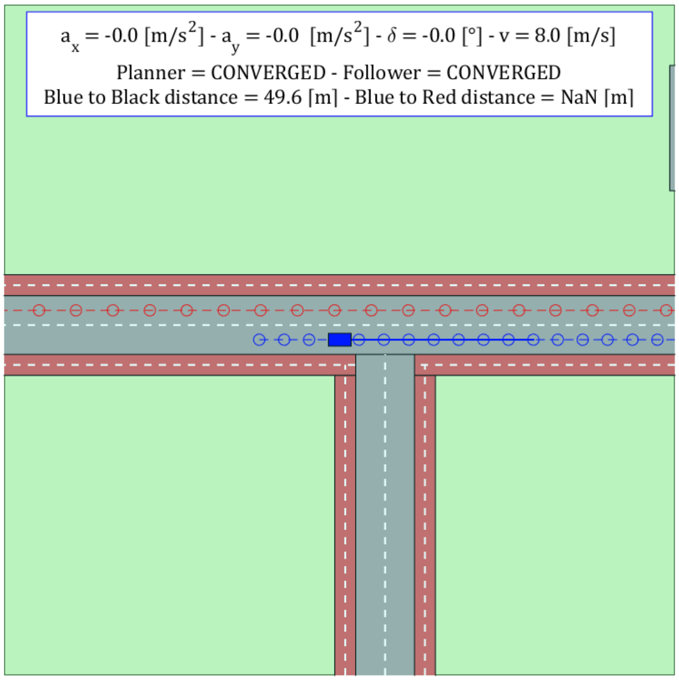
(b) Path Follower optimization results.

Figure 6.35: Scenario 4 optimization processes results.

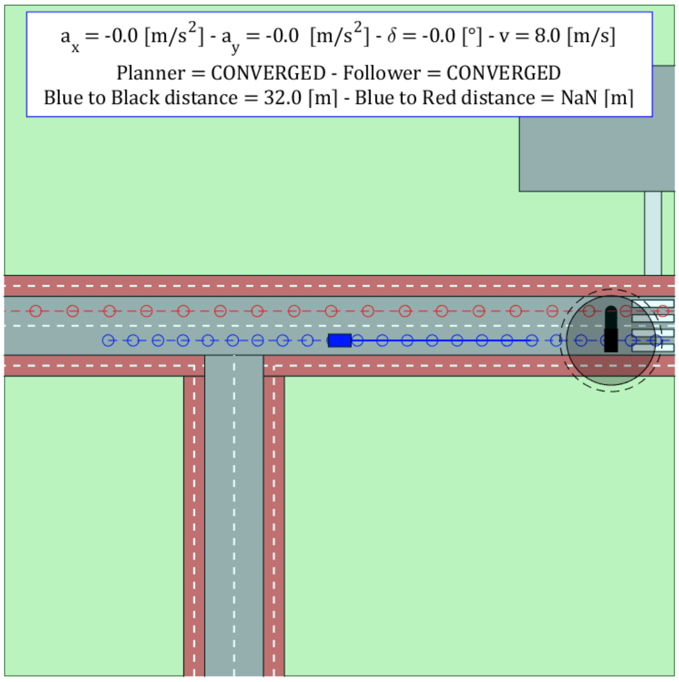
6.5 Scenario 5 - Complete Stop

In this scenario the controlled vehicle faces an obstacle in the middle of the road and has no way to proceed further. As before, due to the low velocity the dynamic plant needs to be used.

The vehicle gradually reduces its speed to then completely stop. The acceleration profiles are perfectly coherent with this behavior (considering that it tries to overtake the obstacle once really close to it). From a computational perspective, the PP shows a bit of uncertainty when the overtaking is (unsuccessfully) evaluated, but works perfectly before and after having realized a complete stop needs to be undertaken. The MPC-based PF constantly operate really well.

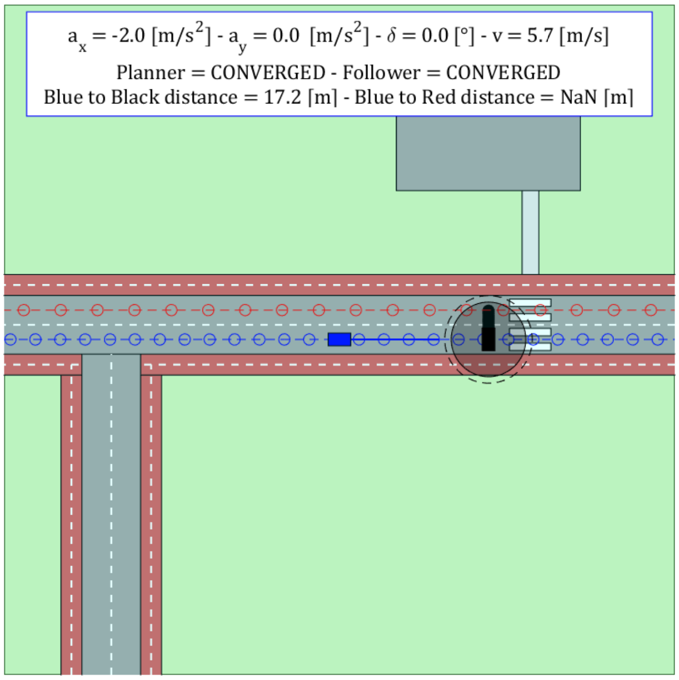


(a)

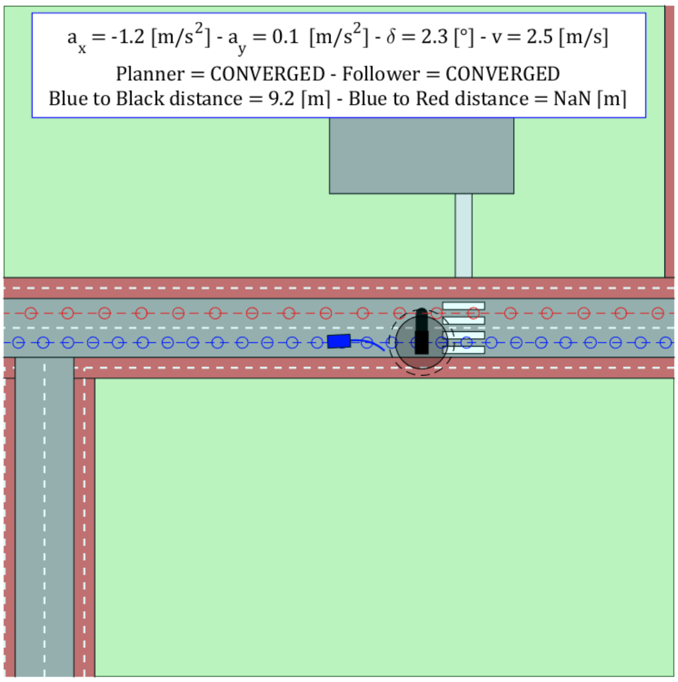


(b)

Figure 6.36: Scenario 5 frames.

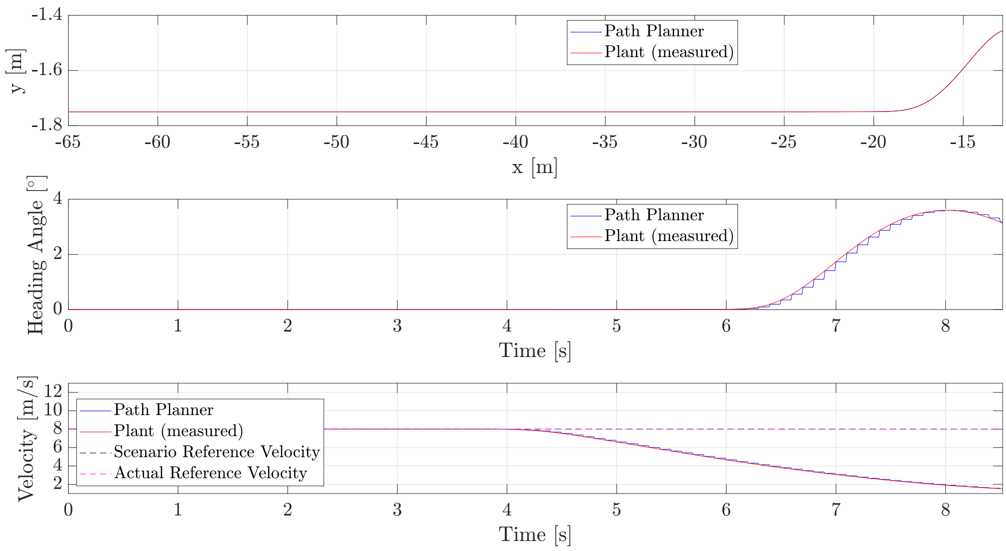


(c)

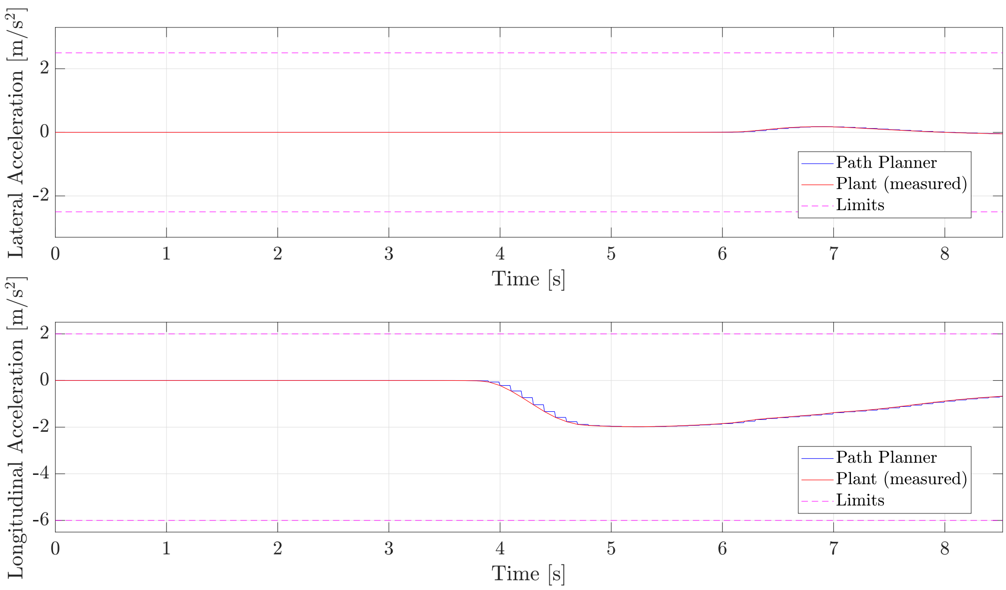


(d)

Figure 6.36: Scenario 5 frames.

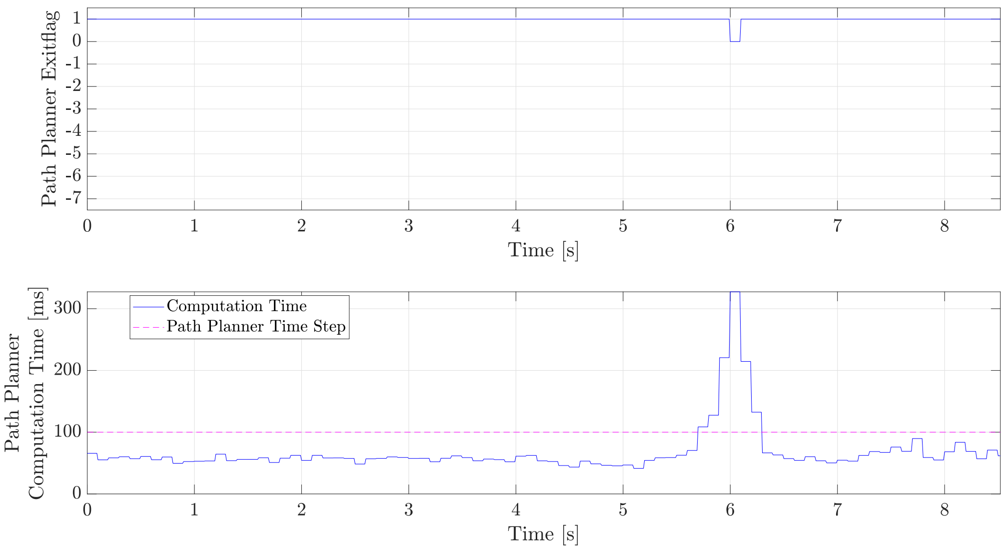


(a) Position, yaw and velocity of the controlled vehicle from both Path Planner (predicited) and Plant (measured).

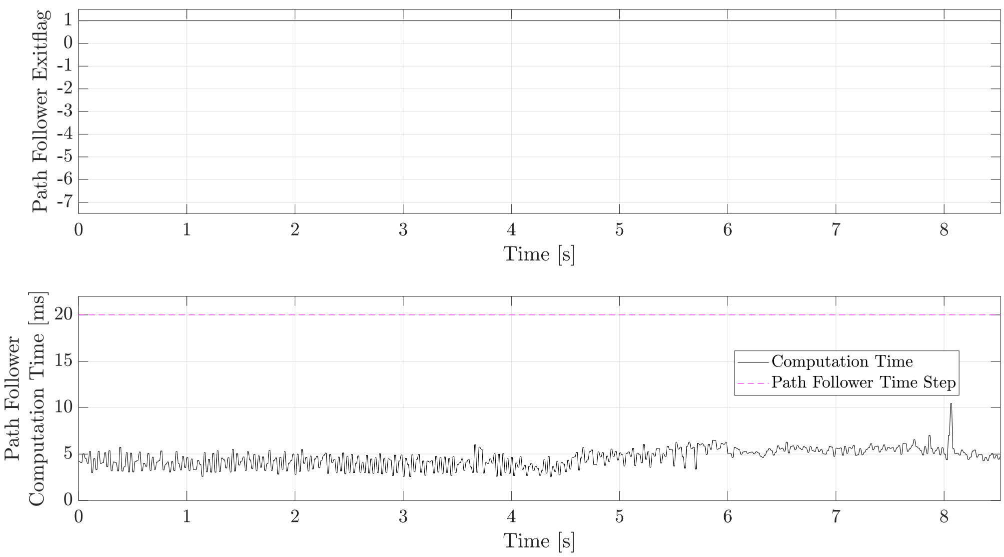


(b) Lateral and longitudinal acceleration of the controlled vehicle.

Figure 6.37: Scenario 5 simulation results.



(a) Path Planner optimization results.



(b) Path Follower optimization results.

Figure 6.38: Scenario 5 optimization processes results.

Conclusions and Future Work

In this chapter the main conclusions regarding the realized Local Motion Planner capacities and performances are drawn. Since some limitations are still present, indications for Future Work are also provided.

Conclusions

The main goal of this thesis was to design an innovative Local Motion Planner for Autonomous Vehicles compatible with the SafeVRU platform. This means that it is expected to operate in real-time and with the other modules present on the same platform. The result is a controller able to ensure collision avoidance in various situations with multiple moving obstacles, whose performances have been proved by means of tests in a dedicated simulation environment.

Firstly, a new Path Planner has been developed through an enriched vehicle prediction model that allows to control comfort and dynamics. Thanks to that the vehicle actually follows the design trajectories and avoids moving obstacles while respecting road rules. This is done via Model Predictive Control whose optimization process is fast enough to operate in real time.

Secondly, planning capabilities have been further improved via a dedicated combined Path Follower, that is based on both MPC and PID techniques. This component adjusts the Path Planner inputs by checking the vehicle actual states at higher frequency, and modeling them with more accuracy. It also considers the road friction coefficient to limit tire forces in case of slippery road (within certain limits). A feedback to the Path Planner is present to correct obstacle dimensions by analyzing estimated trajectory variance.

Thirdly, the complete Local Motion Planner has been enriched with multiple safety features and tested in some representative urban situations. Overall, performances are quite good since the vehicle smoothly follows the optimal trajectory to avoid obstacles and reaches the desired destination (or not when it is expected), in all simulations.

Fourthly, the proposed Local Motion Planner has been repeatedly tested via the created simulation environment. Common urban traffic situations have been reproduced and lately analyzed through a user-friendly process: they all demonstrate the versatility of this unit and how each feature contributes to a successful execution of the expected maneuver.

"Driving in mixed traffic involves numerous interactions with diverse pedestrians, animals, bicyclists and vehicles, and so is more complex than flying an airplane."
— Keith Shaw [63]

Future Work

There are some changes that could be applied to further validate the realized Local Motion Planner and basically test its performances in even more realistic conditions.

Uncertainty could be added in agents localization and future movements predictions in order to reproduce noise in sensor measurements. The number of agents itself could be increased to see how the computational burden changes. Since the employed Behavioral Layer is now user-based, an automatic version could be designed. Basically, it should be able to set the optimization problem, perhaps dynamically limiting the car states (accelerations, jerks etc.), so that the vehicle operates like a real driver would do, which means according to his/her driving preferences and the situation faced. This need emerges especially when dealing with a "smooth" urban context and emergency maneuvers at the same time. Other work could consist in implementing a different tire model, introducing a delay between PP and PF and simulating even more scenarios.

Experimental validation was planned since the beginning of this research project but due to the COVID-19 pandemic it could not take place. Knowing how important it is, experiments are now considered part of future work.

Considerations concerning Technology Diffusion

Regardless the great effort and remarkable investments companies and universities have put in place so far, there are several issues that still need to be addressed prior to Autonomous Driving adoption on a large scale and an interdisciplinary approach is necessary [64].

Purchasing costs represent a big barrier that prevent consumers to benefit of the already available technology, as reported in [65]. Incentives and mass-production can make AVs more affordable over time and this is definitely necessary for technology diffusion.

The *traffic law* itself needs to be adapted to new (probable) litigation that may arise: as the automation level increases, human drivers will be expected to take actions in less situation. Because of the fact that AVs have hardware and software designed, tested and sold to process information and operate quite fast, their decisions may be questioned in a court of law. Insurances itself will need to be modified basically following the responsibility shifting.

People *(cyber)safety* is another key issue. A lot of different nasty scenarios belong to this group, ranging from data theft to loss of vehicle control, which can eventually lead to collisions and traffic disruption. Even though these events may not happen, they can prevent technology diffusion basically because consumers feel in danger. In order to reduce this risk, mitigation techniques commonly used in comparable critical infrastructure systems may be helpful.

Even if consumers are nowadays (unconsciously) compliant with sharing personal data of any kind to benefit of free services (like social media), *privacy* is something policymakers should take care of. Efficient traffic management, route planning, software updates and other potential benefits will be built upon massive data collection (the so called Big Data). Who should own vehicle/passenger data? What type of data will be stored? For what ends will they be used? These questions have not simple answers.

Last but not least there are *ethical issues*. AVs are expected to reduce road accidents up to 90%, but this means there will still be crashes and cars might have to face hard decisions from an ethical perspective, as pointed out in [66]. Potentially infinite unpredictable situations, although quite unlikely to happen, may be encountered. They are often associated to questions like "Who have to be hurt between two passengers and five pedestrians?" (assuming someone seriously has to be hurt, because reality and science fiction are two different things). However, what principles have to be respected when programming AVs is a truly thorny dilemma.

Appendix A

The SafeVRU Platform

In this Appendix, the SafeVRU platform architecture is described in all its modules.

This platform is basically a self-driving vehicle (the converted Toyota Prius depicted in Figure 1.1) able to plan collision-free trajectories in the presence of VRUs (Vulnerable Road Users). It has been designed at TU Delft [15]. It should be noted as among the 1.3 million of road death registered in 2018, more than half are among vulnerable road users: pedestrians, cyclists and motorcyclists [4].

System Architecture

The architecture overview shows how the modules are connected each other.

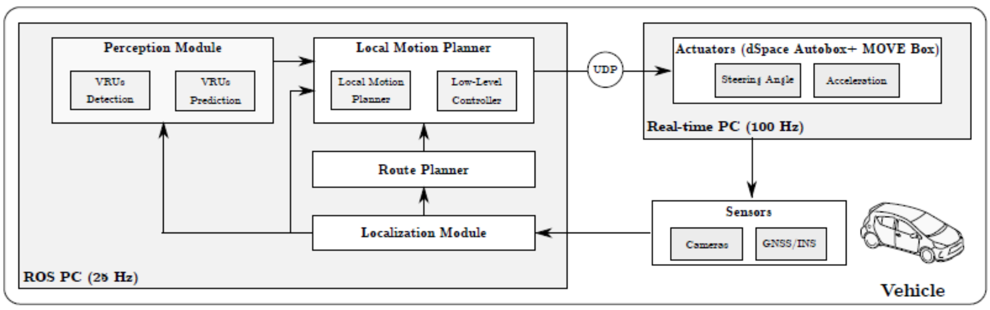


Figure A.1: SafeVRU platform architecture overview.

Route Planner

The route planner is the module designated to provide the global path, that is the sequence of way-points (like points on a map) the vehicle should to pass through in order to get from the starting position to the final destination. Respecting the road rules and other information that may be available (like the road condition which can influence the handling), the presence of static and moving obstacles, it also provides the velocity at each point.

Localization Module

The localization module is basically responsible for localizing the vehicle itself in the world-fixed coordinates system. Information like current pose and speed is sent to the perception module, in order to perform a tracker-based intent recognition, and to the motion planning module to plan the acceleration and steering angle commands.

On SafeVRU, a nonlinear state estimation through sensor fusion consisting of an unscented Kalman filter is used to estimate the vehicle state, which consists of pose, velocity, and angular velocity. See the paper [15] for a more detailed explanation.

Perception Module

The perception module provides to the Local Motion Planner a probabilistic prediction of the future location of the detected objects in the world-fixed coordinate frame (according to the localization module). The whole activity can be split into detection (finding objects) and prediction (future positions estimation).

Local Motion Planner

The Local Motion Planner exploits the information coming from the previous modules in order to compute a collision-free trajectory, and sending the relative acceleration and steering inputs to the car through a real time PC. To achieve this objective, the platform relies on an MPCC formulation. Loosely speaking, it uses constraints and potential fields to ensure obstacle avoidance and the best tracking of the given path.

Low-level Controller

It basically consists of the MOVE Box, which allows to remove the driver from the loop. It enables the longitudinal control by exploiting the existing adaptive cruise control and the lateral control by exploiting the electric power steering system.

Appendix B

Emergency Braking Modeling and Horizon Definition

In this Appendix, the way in which the emergency braking has been modeled is explained, as well as supported with assumptions and practical considerations coherent with the tested situations. Nevertheless, it must be said that, due to the hypothesis behind this formulation, the reality might be quite different.

The actual input variables controlled through the LMP are longitudinal jerk (longitudinal acceleration derivative) and steering acceleration, which are then integrated over time and injected into the plant. This means, for example, that it is necessary to make acceleration negative before reducing vehicle velocity, obviously while respecting LMP constraints. With regard to a hard braking, the worst case scenario is therefore when the vehicle approaches the maximum acceleration ($2\,m/s^2$ according to the current settings) but all of a sudden needs to aggressively slow down. This is the situation taken into account. The vehicle velocity is $10\,m/s$ since it is the reference in most of the simulated scenarios (see Section 6). No slipping is also supposed. These assumptions impose to split the emergency braking maneuver in two phases which are respectively modeled as two types of motion:

1. Constant jerk from maximum acceleration ($2\,m/s^2$) to the minimum one ($-6\,m/s^2$);

2. Constant (negative) acceleration from the velocity reached in phase 1 to $0\,m/s$.

The equations describing phase 1 are:

$$s = s_0 + v_0\, t + \frac{1}{2}\, a_0\, t^2 + \frac{1}{6}\, j\, t^3 \tag{B.1}$$

$$v = v_0 + a_0\, t + \frac{1}{2}\, j\, t^2 \tag{B.2}$$

$$a = a_0 + j\, t \tag{B.3}$$

while the equations describing phase 2 are:

$$s = s_0 + v_0\, t + \frac{1}{2}\, a_0\, t^2 \tag{B.4}$$

$$v = v_0 + a_0\, t \tag{B.5}$$

If both phases are present, that is the vehicle does not stop before reaching minimum acceleration, the results of phase 1 are used as initial values of phase 2. Otherwise, optimal values of dt (PP time step) and N (PP horizon length) are recalculated considering phase 1 only.

According to this model $3\,s$ and $21.7\,m$ are necessary to stop the vehicle with the current settings ($dt = 0.1\ s$, $N = 30$) and reference velocity $v = 10\,m/s$. The current horizon is $3\,s$ long which is equivalent to $30\,m$ at $10\,m/s$.

Bibliography

[1] D.J. Fagnant and K. Kockelman. "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations". In: *Transportation Research Part A: Policy and Practice* 77 (2015), pp. 167–181. ISSN: 09658564. DOI: 10.1016/j.tra.2015.04.003. URL: http://dx.doi.org/10.1016/j.tra.2015.04.003 (cit. on p. 1).

[2] *The future of mobility is at our doorstep.* URL: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-future-of-mobility-is-at-our-doorstep (cit. on p. 1).

[3] *Autonomous Vehicles & Traffic Safety 2018.* Report. European Commission, 2018 (cit. on pp. 1, 2).

[4] World Health Organization. *Global status report on road safety 2018.* Road Safety. 2018, p. 424 (cit. on pp. 1, 141).

[5] *2018 fatal motor vehicle crashes: Overview.* Report. National Center for Statistics and Analysis, 2019 (cit. on p. 2).

[6] W. David Montgomery. *Public and Private Benefits of Autonomous Vehicles.* Publication. Securing America's Future Energy, 2018 (cit. on p. 2).

[7] *Road Safety: Functional limitations, diseases and medication.* 2021. URL: https://ec.europa.eu/transport/road_safety/specialist/knowledge/old/functional_limitations_and_physical_vulnerability/functional_limitations_diseases_and_medication_en (cit. on p. 2).

[8] Payton Chang. *Self-Driving Cars and Their Environmental Impact.* 2018. URL: http://large.stanford.edu/courses/2017/ph240/chang-p2/ (cit. on p. 4).

[9] *Largest autonomous driving patent owners in the United States as of November 2020, by number of active patent families.* 2020. URL: https://www.statista.com/statistics/1051924/autonomous-driving-patent-owners-united-states-authority/ (cit. on p. 6).

[10] *Waymo: passeggeri paganti sulle auto a guida autonoma.* 2018. URL: https://www.alvolante.it/news/waymo-passeggeri-paganti-sulle-auto-guida-autonoma-359934 (cit. on p. 6).

[11] *Argo AI Gets A New Generation Of Automated Test Vehicles As It Moves Into Detroit.* 2019. URL: https://www.forbes.com/sites/samabuelsamid/2019/06/12/argo-ai-gets-a-new-generation-of-automated-test-vehicles-as-it-moves-into-detroit/?sh=4efeffc969cf (cit. on p. 7).

[12] *Toyota says its 'in the game' on autonomous technology.* 2017. URL: https://www.autonews.com/article/20171023/MOBILITY/171029940/toyota-says-its-in-the-game-on-autonomous-technology (cit. on p. 7).

[13] *Cruise Begins First Driverless Automated Tests In San Francisco.* 2020. URL: https://www.forbes.com/sites/samabuelsamid/2020/12/09/cruise-begins-first-driverless-automated-tests-in-san-francisco/?sh=686aa3bc735b (cit. on p. 8).

[14] *Daimler starts pilot testing of self-driving Mercedes taxis in the U.S.* 2019. URL: https://www.autonews.com/mobility-report/daimler-starts-pilot-testing-self-driving-mercedes-taxis-us (cit. on p. 8).

[15] L. Ferranti, B. Brito, E. Pool, Y. Zheng, R. M. Ensing, et al. "SafeVRU: A Research Platform for the Interaction of Self-Driving Vehicles with Vulnerable Road Users". In: *2019 IEEE Intelligent Vehicles Symposium (IV).* 2019, pp. 1660–1666. DOI: 10.1109/IVS.2019.8813899 (cit. on pp. 9, 11, 17, 30, 141, 142).

[16] Y. Kanayama and B. I. Hartman. "Smooth local path planning for autonomous vehicles". In: *Proceedings, 1989 International Conference on Robotics and Automation.* 1989, 1265–1270 vol.3. DOI: 10.1109/ROBOT.1989.100154 (cit. on p. 11).

[17] Z. Shiller and Y. -. Gwo. "Dynamic motion planning of autonomous vehicles". In: *IEEE Transactions on Robotics and Automation* 7.2 (1991), pp. 241–249. DOI: 10.1109/70.75906 (cit. on p. 11).

[18] *TU Delft Vehicle Demo at the IV Conference 2019*. 2019. URL: http://intelligent-vehicles.org/posts/tu-delft-vehicle-demo-at-the-iv-conference-2019/ (cit. on p. 12).

[19] D. González, J. Pérez, V. Milanés, and F. Nashashibi. "A Review of Motion Planning Techniques for Automated Vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* 17.4 (2016), pp. 1135–1145. DOI: 10.1109/TITS.2015.2498841 (cit. on pp. 12, 16).

[20] Christos Katrakazas, Mohammed Quddus, Wen-Hua Chen, and Lipika Deka. "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions". In: *Transportation Research Part C: Emerging Technologies* 60 (2015), pp. 416–442. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2015.09.011. URL: http://www.sciencedirect.com/science/article/pii/S0968090X15003447 (cit. on pp. 12, 17).

[21] Jon Bohren, Tully Foote, Jim Keller, Alex Kushleyev, Daniel Lee, et al. "Little Ben: The Ben Franklin Racing Team's Entry in the 2007 DARPA Urban Challenge". In: Nov. 2009, pp. 231–255. ISBN: 978-3-642-03990-4. DOI: 10.1007/978-3-642-03991-1_6 (cit. on p. 14).

[22] *The Grand Challenge*. URL: https://www.darpa.mil/about-us/timeline/-grand-challenge-for-autonomous-vehicles (cit. on p. 14).

[23] Michael Montemerlo, Jan Becker, Suhrid Bhat, Hendrik Dahlkamp, Dmitri Dolgov, et al. "Junior: The Stanford Entry in the Urban Challenge". In: *Journal of Field Robotics* 25 (Sept. 2008), pp. 569–597. DOI: 10.1002/rob.20258 (cit. on p. 14).

[24] Lydia Kavraki, Petr Svestka, J.C. Latombe, and M.H. Overmars. "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces". In: *Robotics and Automation, IEEE Transactions on* 12 (Sept. 1996), pp. 566–580. DOI: 10.1109/70.508439 (cit. on p. 15).

[25] Steven LaValle and James Kuffner. "Randomized Kinodynamic Planning." In: *I. J. Robotic Res.* 20 (Jan. 2001), pp. 378–400 (cit. on p. 15).

[26] T. Faulwasser, B. Kern, and R. Findeisen. "Model predictive path-following for constrained nonlinear systems". In: *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly*

with 2009 28th Chinese Control Conference. 2009, pp. 8642–8647. DOI: 10.1109/CDC.2009.5399744 (cit. on p. 17).

[27] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman, and D. Rus. "Safe Nonlinear Trajectory Generation for Parallel Autonomy With a Dynamic Vehicle Model". In: *IEEE Transactions on Intelligent Transportation Systems* 19.9 (2018), pp. 2994–3008. DOI: 10.1109/TITS.2017.2771351 (cit. on pp. 17, 42).

[28] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora. "Model Predictive Contouring Control for Collision Avoidance in Unstructured Dynamic Environments". In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 4459–4466. DOI: 10.1109/LRA.2019.2929976 (cit. on pp. 17, 38, 43, 46, 47).

[29] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang. "Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control With Multiconstraints". In: *IEEE Transactions on Vehicular Technology* 66.2 (2017), pp. 952–964. DOI: 10.1109/TVT.2016.2555853 (cit. on pp. 17, 46).

[30] F. Borrelli, Paolo Falcone, Tamas Keviczky, J. Asgari, and D. Hrovat. "MPC–based approach to active steering for autonomous vehicle systems". In: *International Journal of Vehicle Autonomous Systems* 3 (Jan. 2005), pp. 265–291. DOI: 10.1504/IJVAS.2005.008237 (cit. on p. 17).

[31] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovsky. "The development of Model Predictive Control in automotive industry: A survey". In: *2012 IEEE International Conference on Control Applications*. 2012, pp. 295–302. DOI: 10.1109/CCA.2012.6402735 (cit. on p. 17).

[32] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles". In: *IEEE Transactions on Intelligent Vehicles* 1 (Apr. 2016). DOI: 10.1109/TIV.2016.2578706 (cit. on p. 19).

[33] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker. "A Predictive Controller for Autonomous Vehicle Path Tracking". In: *IEEE Transactions on Intelligent Transportation Systems* 10.1 (2009), pp. 92–102. DOI: 10.1109/TITS.2008.2011697 (cit. on p. 20).

[34] E. Kim and Myoungho Sunwoo. “Model predictive control strategy for smooth path tracking of autonomous vehicles with steering actuator dynamics”. In: *International Journal of Automotive Technology* 15 (Dec. 2014), pp. 1155–1164. DOI: `10.1007/s12239-014-0120-9` (cit. on p. 20).

[35] Paolo Falcone, F. Borrelli, Eric Tseng, J. Asgari, and D. Hrovat. “Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation”. In: *International Journal of Robust and Nonlinear Control* 18 (May 2008), pp. 862–875. DOI: `10.1002/rnc.1245` (cit. on p. 20).

[36] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat. “Predictive Active Steering Control for Autonomous Vehicle Systems”. In: *IEEE Transactions on Control Systems Technology* 15.3 (2007), pp. 566–580. DOI: `10.1109/TCST.2007.894653` (cit. on p. 20).

[37] R. Craig Coulter. *Implementation of the Pure Pursuit Path Tracking Algorithm*. Tech. rep. CMU-RI-TR-92-01. Pittsburgh, PA: Carnegie Mellon University, Jan. 1992 (cit. on p. 20).

[38] Moveh Samuel, Mohamed Hussein, and Maziah Binti. “A Review of some Pure-Pursuit based Path Tracking Techniques for Control of Autonomous Vehicle”. In: *International Journal of Computer Applications* 135.1 (Feb. 2016), pp. 35–38. DOI: `10.5120/ijca2016908314` (cit. on p. 20).

[39] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi. “A stable tracking control method for an autonomous mobile robot”. In: *Proceedings., IEEE International Conference on Robotics and Automation*. 1990, 384–389 vol.1. DOI: `10.1109/ROBOT.1990.126006` (cit. on p. 20).

[40] Alexander Winter and Simone Baldi. “Real-Life Implementation of a GPS-Based Path-Following System for an Autonomous Vehicle”. In: *Sensors* 18 (Nov. 2018), p. 3940. DOI: `10.3390/s18113940` (cit. on p. 20).

[41] I. Rivals, L. Personnaz, G. Dreyfus, D. Cañas, and Sagem Eragny. “Real-time control of an autonomous vehicle: a neural network approach to the path following problem”. In: 1993 (cit. on p. 20).

[42] Tom J.J. van den Boom. *Model Predictive Control*. 2016 (cit. on p. 29).

[43] Sergio Grammatico. "Model Predictive Control - SC42125. Lecture Notes". 2020. URL: sites.google.com/site/grammaticosergio/teaching/mpc (cit. on p. 32).

[44] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli. "Kinematic and dynamic vehicle models for autonomous driving control design". In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. 2015, pp. 1094–1099. DOI: 10.1109/IVS.2015.7225830 (cit. on pp. 33, 56).

[45] Jose Matute-Peaspan, Mauricio Marcano, Sergio Diaz, and Joshué Pérez. "Experimental Validation of a Kinematic Bicycle Model Predictive Control with Lateral Acceleration Consideration". In: vol. 52. July 2019. DOI: 10.1016/j.ifacol.2019.08.085 (cit. on p. 33).

[46] Philip Polack, Florent Altché, Brigitte Novel, and Arnaud de La Fortelle. *Guaranteeing Consistency in a Motion Planning and Control Architecture Using a Kinematic Bicycle Model.* Apr. 2018 (cit. on p. 34).

[47] Il Bae, J. Moon, Junekyo Jhung, H. Suk, T. Kim, et al. *Self-Driving like a Human driver instead of a Robocar: Personalized comfortable driving experience for autonomous vehicles.* Jan. 2020 (cit. on p. 35).

[48] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, et al. "FaSTrack: A modular framework for fast and guaranteed safe motion planning". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. 2017, pp. 1517–1522. DOI: 10.1109/CDC.2017.8263867 (cit. on p. 39).

[49] L. Svensson and Jenny Casey Eriksson. "Tuning for Ride Quality in Autonomous Vehicle: Application to Linear Quadratic Path Planning Algorithm". In: 2015 (cit. on pp. 39, 41).

[50] W. Schwarting, J. Alonso-Mora, L. Pauli, S. Karaman, and D. Rus. "Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 1928–1935. DOI: 10.1109/ICRA.2017.7989224 (cit. on p. 42).

[51] Sebastian Brechtel, Tobias Gindele, and Rüdiger Dillmann. "Probabilistic MDP-Behavior Planning for Cars". In: Oct. 2011. DOI: 10.1109/ITSC.2011.6082928 (cit. on p. 47).

[52] Matteo Pirovano. "Implementation on a Prototype Vehicle of a NMPC Trajectory Planner for Urban Scenarios". Thesis. Politecnico di Milano, 2020 (cit. on p. 50).

[53] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari. "FORCES NLP: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs". In: *International Journal of Control* 93.1 (2020), pp. 13–29. ISSN: 13665820. DOI: `10.1080/00207179.2017.1316017`. URL: `https://doi.org/10.1080/00207179.2017.1316017` (cit. on pp. 50, 54).

[54] Alexander Domahidi Jerez and Juan. *FORCES Professional* (cit. on pp. 50, 54).

[55] D.Q. Mayne, James Rawlings, Christopher Rao, and P. Scokaert. "Constrained Model Predictive Control: Stability and Optimality". In: *Automatica* 36 (June 2000), pp. 789–814. DOI: `10.1016/S0005-1098(99)00214-9` (cit. on p. 54).

[56] Mingyuan Bian, Long Chen, Yugong Luo, and Keqiang Li. "A Dynamic Model for Tire/Road Friction Estimation under Combined Longitudinal/Lateral Slip Situation". In: vol. 1. Apr. 2014. DOI: `10.4271/2014-01-0123` (cit. on p. 57).

[57] Massachusetts: The MathWorks Inc. Natick. *MATLAB*. 2019 (cit. on p. 69).

[58] Massachusetts: The MathWorks Inc. Natick. *Simscape Multibody*. 2018 (cit. on pp. 69, 70).

[59] TNO. *Delft Tyre*. 2017. URL: `http://www.delft-tyre.nl` (cit. on p. 69).

[60] Z. Lu, B. Shyrokau, B. Boulkroune, S. van Aalst, and R. Happee. "Performance benchmark of state-of-the-art lateral path-following controllers". In: *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*. 2018, pp. 541–546. DOI: `10.1109/AMC.2019.8371151` (cit. on p. 69).

[61] Quirinus Wilhelmus Adrianus van der Slot. "Comfort oriented nonlinear model predictive control for autonomous vehicles". Thesis. Delft University of Technology, 2020 (cit. on p. 69).

[62] O. de Groot, B. Brito, L. Ferranti, D. Gavrila, and J. Alonso-Mora. "Scenario-Based Trajectory Optimization in Uncertain Dynamic Environments". In: *arXiv e-prints*, arXiv:2103.12517 (Mar. 2021), arXiv:2103.12517. arXiv: `2103.12517 [cs.RO]` (cit. on p. 73).

[63] Todd Litman. *Autonomous Vehicle Implementation Predictions*. Victoria Transport Policy Institute, 2020. URL: `https://www.vtpi.org/avip.pdf` (cit. on p. 138).

[64] P. Koopman and M. Wagner. "Autonomous Vehicle Safety: An Interdisciplinary Challenge". In: *IEEE Intelligent Transportation Systems Magazine* 9.1 (2017), pp. 90–96. DOI: 10.1109/MITS.2016.2583491 (cit. on p. 139).

[65] Daniel J. Fagnant and Kara Kockelman. "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations". In: *Transportation Research Part A: Policy and Practice* 77 (2015), pp. 167–181. ISSN: 0965-8564. DOI: https://doi.org/10.1016/j.tra.2015.04.003. URL: http://www.sciencedirect.com/science/article/pii/S0965856415000804 (cit. on p. 139).

[66] Jean-François Bonnefon, Azim Shariff, and Iyad Rahwan. "The Social Dilemma of Autonomous Vehicles". In: *Science* 352 (June 2016). DOI: 10.1126/science.aaf2654 (cit. on p. 139).