Consistency and stability of hierarchical planning and control systems for autonomous driving

Philip Polack

THÈSE DE DOCTORAT

de l'Université de recherche Paris Sciences et Lettres
PSL Research University

Préparée à MINES ParisTech

Consistency and stability of hierarchical planning and control systems for autonomous driving
Cohérence et stabilité des systèmes hiérarchiques de planification et de contrôle pour la conduite automatisée

École doctorale n°432

SCIENCE DES MÉTIERS DE L'INGÉNIEUR

Spécialité MATHÉMATIQUES ET INFORMATIQUE TEMPS-RÉEL

COMPOSITION DU JURY :

M Alain Oustaloup
Univ. Bordeaux, Président du jury

M Thierry-Marie Guerra
Univ. Valenciennes, Rapporteur

M Philippe Martinet
Inria Sophia Antipolis, Rapporteur

M Ming Yang
Univ. Shanghai Jiao-Tong, Membre du jury

Mme Brigitte d'Andréa-Novel
Mines ParisTech, Membre du jury

M Arnaud de La Fortelle
Mines ParisTech, Membre du jury

Soutenue par **Philip POLACK**
le 29 octobre 2018

Dirigée par **Brigitte D'ANDRÉA-NOVEL ET ARNAUD DE LA FORTELLE**

MINES ParisTech | PSL

Abstract

Autonomous vehicles are believed to reduce the number of deaths and casualties on the roads while improving the traffic efficiency. However, before their mass deployment on open public roads, their safety must be guaranteed at all time. Therefore, this thesis deals with the motion planning and control architecture for autonomous vehicles and claims that the intention of the vehicle must match with its actual actions. For that purpose, the kinematic and dynamic feasibility of the reference trajectory should be ensured. Otherwise, the controller which is blind to obstacles is unable to track it, setting the ego-vehicle and other traffic participants in jeopardy. The proposed architecture uses Model Predictive Control based on a kinematic bicycle model for planning safe reference trajectories. Its feasibility is ensured by adding a dynamic constraint on the steering angle which has been derived in this work in order to ensure the validity of the kinematic bicycle model. Several high-frequency controllers are then compared and their assets and drawbacks are highlighted. Finally, some preliminary work on model-free controllers and their application to automotive control are presented. In particular, an efficient tuning method is proposed and implemented successfully on the experimental vehicle of ENSIAME in collaboration with the laboratory LAMIH of Valenciennes.

Contents

List of Figures

List of Tables

Acknowledgements

This work would not have been possible without the direct or indirect contributions of many people. First of all, I would like to thank my two advisors: Brigitte, who has always guided me thoroughfully in my research and read all my publications scrupulously, correcting even punctuation and accentuation errors; Arnaud, who trusted me even though I did not have a background in robotics/automation and encouraged me to nurture my knowledge on autonomous vehicles outside of my domain. This work would not have been possible also without Florent, whose strong scientific and computer science skills guided me throughout my thesis, and Sébastien, from whom I got the chance to learn a lot on control applications and scientific rigor during my numerous stays in Valenciennes.

I would also like to thank my lab mates with whom I have spent my three years of PhD: Xavier who introduced me to new board games especially King of Tokyo; Daniele who brought Italy to our doorstep; Edgar who aroused my curiosity with his music instrument “Embodme" combining mathematics and AI; Eva and Jun with whom I won the Valeo Innovation Challenge 2016; Jean-Emmanuel, Arthur and Michelle for visiting China together; Amaury and Hassan for our runs in the Luxembourg garden on Thursday mornings; Cyril for joining as a goalkeeper my soccer team; Philippe for our passionate debates about politics and life; Guillaume for visiting Hawaii with me; Hugues, Martin and Paul for the numerous board games; Aubrey for making me discover the slack line; Grégoire for an introduction to the virtual reality world; Mathieu for the rock’n’roll; Marion for the classical music; Marin for the crazy driving with Reinforcement Learning; Thomas for his help in C++. Thank you also to my colleagues François, Fabien, Silvère, Simon, and of course Christine and Christophe who easen my administrative tasks.

During my thesis, I also got the chance to meet a lot of researchers who took some time to discuss my work. I would like to thank in particular Michel for thorough discussion about model-free control, Eduardo about Model Predictive Control, Jean-Paul and Nicolas about motion planning, Lghani, Dominique, and people from the Chair, in particular Wei, Ching-Yao and Wei-Bin from UC Berkeley.

My three months in China have been quite of an adventure too! I have discovered a whole new culture with wonderful colleagues. In particular, I would like to thank Chaojie, mister “super outstanding", for showing me around and introducing me to the chinese culture; Wei, mister “sugar" (his lastname Tang means sugar in chinese), for helping me out with the administrative stuffs and driving me like an “old driver" to the airport; Lihong for his always positive energy; Yao for our controller contest.

I am very grateful to the jury members for assessing my work: Thierry-Marie whose nice comments about the manuscript persuaded me that it was worth spending a couple of months on this task; Philippe for his challenging comments about my work; Ming for his curiosity about my recent research on deep learning control and without whom my three-month experience in China would not have been possible; Alain for his course about PID control.

My family and friends have also played an important role in this work. In particular, I would like to thank my parents and my brother for their support, and my grand-mother for coming to my PhD defense all the way from Le Mans at the age of 95!

Last but not least, I would like to gratefully thank Tan-Mai, who always gave me her support, not only in the good times but also in the harder moments during this thesis... This work would not have been possible without her constant encouragements!

Foreword

The work undertaken during this thesis is part of the international research Chair MINES ParisTech - Peugeot-Citroën - Safran - Valeo on ground vehicle automation and coordinated by Professor Arnaud de La Fortelle[1]. This research Chair also comprises academic partners: University of California Berkeley (USA), Ecole Polytechnique Fédérale de Lausanne (Switzerland) and Shanghai Jiao-Tong University (China).

This five-year Chair with a funding of 3.7 million euros has three main objectives:

- Improve the knowledge on autonomous driving;
- Develop onboard intelligence devices;
- Run automated vehicle tests on three different continents (Asia, North America and Europe).

[1] http://chair-driveforyou.com/

Chapter 1

Introduction

“ Si l’on se préoccupait de l’achèvement des choses, on n’entreprendrait jamais rien [a] ”

[a] If we were concerned about the completion of things, we would never undertake anything
François Ier, roi de France

Contents

1.1 What is an autonomous vehicle?

In 1886, Karl Benz patented the first *motor vehicle* in the world. He benefitted from the technological advances entailed by the industrial revolution to make a vehicle move on its “own", based on an internal combustion engine, replacing progressively animal-drawn *carriages*. Interestingly, the french language distinguishes *automobiles* (motor vehicles) that are moving using their own energy from *hippomobiles* (carriages), horse-drawn vehicles [Laurgeau, 2009]. In less than half a century, motor vehicles completely changed the way people moved, destroying on its rise lots of jobs such as farriers to replace them by new ones, for example gas station attendants. Motor vehicles became accessible to the masses in 1908 with Ford’s Model T and became a symbol of the “American way of life" after World War II.

Nowadays, the digital revolution has replaced the industrial revolution, and in this “Second Machine Age" [Brynjolfsson and McAfee, 2014], the future of mobility might be completely reshaped: the car which was seen in the 1980s as a purely mechanical object is now becoming more of an electronic device. And autonomous vehicles that might have been seen as pure science fiction until not so long ago are about to become real. This paradigm shift has some important consequences on the automotive industry market as clients are becoming more interested in digital features (ex: connectivity, driver assistance systems) than in mechanical performances (ex: horsepowers, acceleration capacities).

1.1.1 History of autonomous vehicles

The early stages of autonomous vehicle with actual vehicle prototypes date back to the 1980s, with mainly three projects:

- NavLAB – National Autonomous Vehicle LABoratory – launched in 1984 at Carnegie Mellon University [Thorpe et al., 1991a, Thorpe et al., 1991b] which proposed successive eponymous vehicle prototypes: Navlab 1 in 1986 to Navlab 10 in 1996.

- Prometheus – PROgraM for a European Traffic of Highest Efficiency and Unprecedented Safety – a European project promoted by Ernst Dickermanns at the Bundeswehr University of Munich and Mercedes-Benz, launched in 1987, and which led to a prototype, the VaMP (Versuchsfahrzeug für autonome Mobilität und Rechnersehen[1]).

- California PATH – Partners for Advanced Transit and Highways – program founded in 1986 [Shladover, 2006] as a collaboration between California Department of Transportation (Caltrans), UC Berkeley and other public and private partners, and which has been a pioneer in Automated Highway Systems (AHS), in particular focusing on increasing the highway capacity and safety while reducing the traffic congestion, the air pollution and the energy consumption.

However, already back in 1925, the inventor Francis Houdina and his company Houdina Radio Control demonstrated a radio-controlled “driverless" car, the American Wonder, nicknamed the “Phantom Auto", which drove on Broadway and Fifth Avenue in New York without any human on-board[2]. It was controlled remotely by a human located in a second car following close behind using radio technology.

Many other projects have emerged afterwards from the 1990s until now, such as the ARGO project[3] (1996) led by the universities of Parma and Pavia in Italy, and the European projects

[1] Experimental vehicle for autonomous mobility and computer vision

[2] https://www.inverse.com/article/14438-remotely-controlled-autonomous-vehicles-are-coming-for-better-or-for-worse, visited April 2018.

[3] www.argo.ce.unipr.it, visited on December 2017.

CityMobil[4] (2006-2011), CityMobil2[5] (2012-2016), interactIVe[6] (2010-2013), AutoNET2030[7] (2013-2016), i-GAME[8] (2013-2016), iTeam[9] (2016-2019).

Research on Autonomous Vehicles really reached a milestone with the successive DARPA challenges, organized by the Defense Advanced Research Projects Agency part of the United States Department of Defense. These challenges were the first long distance competitions for driverless car in the world. After a first unsuccessfull edition in 2004, the 2005 DARPA Grand Challenge was won by the Stanley vehicle from Stanford University (see Figure 1.1a): it accomplished the 240km of route in the desert in 6 hours and 54 minutes. The 2007 DARPA Urban Challenge consisted in a 96km urban area course where the autonomous vehicles had to obey traffic rules, respect other traffic participants, and avoid obstacles. This time, the Boss vehicle from Carnegie Mellon University won the competition at an average speed of 22.53km/h (see Figure 1.1b).

(a) Stanley from Stanford University (b) Boss from Carnegie Mellon University

Figure 1.1 – The winners of the DARPA Grand Challenge in 2005 (a) and the DARPA Urban Challenge in 2007 (b).

More recently, other competitions took place. Among others, let's mention:

- the Intelligent Vehicles Future Challenge (IVFC) which has been held every year in China since 2009 and gathers teams from many major universities and scientific research institutions of the country. The competitors are required to finish not only a test on an actual road in order to examine the safety, comfort, agility and intelligence of the driverless vehicle, but also an offline test to check out the vehicles' basic cognitive ability by simulating an actual road environment.

- the Grand Cooperative Driving Challenges (GCDC) which were held in the Netherlands in 2011 [Ploeg et al., 2012] and in 2016 [Englund et al., 2016] and organized by the European project i-GAME. Their goal was to prove a basis for cooperative automated driving in an international context.

Some recent projects have confirmed the technical advances that have been made in the domain of autonomous vehicles: in 2010, VisLab Intercontinental Autonomous Challenge (VIAC) involved four driverless vehicles which drove an almost 16000km trip from Parma in Italy, to Shanghai in China. This challenge enabled to collect about 50 terabytes of data. Then in 2012, only 125 years after Karl Benz's wife Bertha complete the first journey in automotive history, a vehicle from Daimler in collaboration with Karlsruhe Institute of Technology and Forschungszentrum Informatik[10] completed the same journey from Mannheim to Pforzheim in fully autonomous mode [Ziegler et al., 2014b]. Finally, the Google car project[11] launched in 2009, is probably the project

[4] http://www.citymobil-project.eu/, visited on December 2017.
[5] http://www.citymobil2.eu/en/, visited on December 2017.
[6] http://www.interactive-ip.eu/, visited on December 2017.
[7] http://www.autonet2030.eu/, visited on December 2017.
[8] http://gcdc.net/en/i-game, visited on December 2017.
[9] https://iteam-project.net/, visited on December 2017.
[10] Research Center for Information Technology
[11] https://waymo.com/journey/, visited on April 2018.

that has gained most public attention. In February 2018, its fleet of vehicles had drove over 5 million miles[12] in self-driving mode, with a rate growing exponentially due to the expansion of the number of vehicles. In December 2016, Google decide to launch a subsidiary company, named Waymo, working specifically on autonomous vehicles. Other companies, such as Uber or NuTonomy have also launched a huge fleet of vehicles around the globe to test their technology. Notice that France has also some companies with their own fleet, such as Navya and Easymile.

1.1.2 From Advanced Driver Assistance Systems...

Safety has always been the main priority for both the car industry and legislators. Car accidents are responsible for over 3400 deaths and 70000 casualties in 2017 in France[13], 37000 deaths and 2000000 casualties in 2016 in the USA[14], and 1.2M deaths worldwide[15] in 2013. To reduce these figures, the automotive industry has been developing constantly new safety systems. Commonly, we distinguish *passive* and *active* safety systems:

- *Passive safety* is used to refer to components of the vehicle that help to protect occupants during a crash.
  Examples: airbags, seatbelts and the physical structure of the vehicle.

- *Active safety* is used to refer to technology assisting in the prevention of a crash.
  Examples: Anti-lock Braking System (ABS), Electronic Stability Program (ESP) and Advanced Driver Assistance Systems (ADAS).

First, research on active safety has been focusing on how to enhance the handling and stability of the vehicle, by compensating some undesired dynamic effects. They have lead to the development of new systems, in particular:

- *Anti-lock Braking System (ABS)* - 1978: braking assist system that prevents the wheels from locking up during severe braking.

- *Electronic Stability Program (ESP)* - 1995: antiskid system aimed at enhancing the control of the vehicle for a better road handling. It enables to detect the loss of grip in curves and counteract by braking one or several wheels.

These systems are now mandatory on all new vehicles in the European Union since respectively 2004 and 2014.

However 94% of car accidents are due to human factors[16], such as drowsiness, alcohol and speeding. Therefore with the emergence of technologies for autonomous vehicles, car manufacturers and Original Equipment Manufacturers (OEM) have developed increasingly efficient active safety systems, such that nowadays, the driver can be relegated to a role of simple supervisor of the driving task; the onboard intelligence is able to act directly on the steering, acceleration and braking. These systems, gathered under the acronym ADAS (for Advanced Driver Assistance Systems) are already available on high-end commercial vehicles. Among them, let's mention:

- *Adaptive Cruise Control (ACC)* - 2011: cruise control system that automatically adjusts the speed of the vehicle to maintain a safe distance with the preceding one.

- *Collision Avoidance System or Autonomous Emergency Braking (AEB)* : system that either provides a warning to the driver when there is an imminent collision risk or takes action autonomously, generally by braking at low vehicle speeds (below 50 km/h) and by steering at higher speeds.

[12] https://waymo.com/ontheroad/, visited on April 2018.
[13] http://www.securite-routiere.gouv.fr/la-securite-routiere/l-observatoire-national-interministeriel-de-la-securite-routiere, visited on April 2018.
[14] https://www.nhtsa.gov/press-releases/usdot-releases-2016-fatal-traffic-crash-data, visited on April 2018.
[15] http://www.who.int/gho/road_safety/mortality/traffic_deaths_number/en/, visited on April 2018.
[16] https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812115, visited on April 2018.

- *Lane-Departure Warning (LDW)* - 1992: system that warns the driver when the vehicle begins to move out of its lane (unless the turn signal is on in that direction).

- *Lane Keeping System (LKS)* : feature that in addition to the lane departure warning system automatically takes actions to ensure the vehicle stays in its lane.

- *Park Assist*: system that moves automatically a vehicle from a traffic lane into a parking spot by performing parallel, perpendicular or angle parking.

Finally, *eCall* launched by the European Union and mandatory since April 2018 on all new vehicles enables to automatically send an emergency call in the case of a crash providing its precise localization and its severity.

1.1.3 ... to fully autonomous vehicles

The automation of a vehicle has been classified into 6 different levels by the Society of Automotive Engineers[17] (SAE). These levels are summarized in Table 1.1, starting from level 0 corresponding to no automation to level 5 where the steering wheel and pedals are not necessary anymore. These levels are now standard in the car industry since their adoption by the National Highway Traffic Safety Administration (NHTSA), part of the United States Department of Transportation.

N°	SAE Levels	Description
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.
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 performs all remaining aspects of the dynamic driving task.
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 performs all remaining aspects of the dynamic driving task.
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.
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.
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.

Table 1.1 – The different SAE levels.

[17] https://www.sae.org/binaries/content/assets/cm/content/news/press-releases/pathway-to-autonomy/automated_driving.pdf, visited on April 2018.

Combining different ADAS presented in section 1.1.2 can provide vehicles up to level 2. This is the case for example of Tesla Motors' vehicles where the different assistance systems are gathered under the name "Autopilot". The vehicle is able to remain in its lane while adjusting its speed to the vehicle located in front. However, such vehicles are not autonomous vehicles as the driver might have to take over the control at anytime due for example to snow conditions hidding the lane marking[18] or unclear lane marking[19]. Today, only commercial vehicles up to level 2 are available on the market and allowed to drive on public roads. Audi has announced the release of the first commercial vehicle of level 3 by 2019: the Audi A8. But according to Vienna Convention (1968), the driver remains responsible in the case of an accident.

Nowadays most carmakers and OEM have their own autonomous vehicle prototypes, sometimes up to level 4, pushed by disruptive actors stemming from the technological area such as Google, Tesla, Uber and even Apple. The prototype of the Valeo's hands-off tour is displayed as an example in Figure 1.2 and the Google car in Figure 1.3. However, these vehicles have often a very conservative driving strategy. Therefore, they can be unefficient in terms of traffic flow as they can get stuck by for example road works, stopped cars, traffic wardens or four-way stops. In particular, they are unable to take a decision when there is too much uncertainties[20]. Moreover, they often rely on a human intervention to perform a lane change (or at least to take the decision).

Figure 1.2 – Valeo's hand-off tour prototype.

Figure 1.3 – The Google Car.

1.2 Technical components of an autonomous vehicle

An autonomous vehicle is a wheeled robot. The hardware and software used in the early stages of autonomous vehicles were simple adaptation of the one used on small wheeled robots. These algorithms were suited for the low speed applications considered at the beginning, where dynamics effects for example could be neglected, as the goal was to build a Proof of Concept. As a robot, the

[18] https://www.youtube.com/watch?v=lTtLGP4cRaQ, visited on April 2018.
[19] https://www.youtube.com/watch?v=6QCF8tVqM3I&feature=youtu.be, visited on April 2018.
[20] https://www.youtube.com/watch?v=tiwVMrTLUWg, visited on April 2018.

organization of an autonomous vehicle can be divided into tasks that require more or less intelligence.

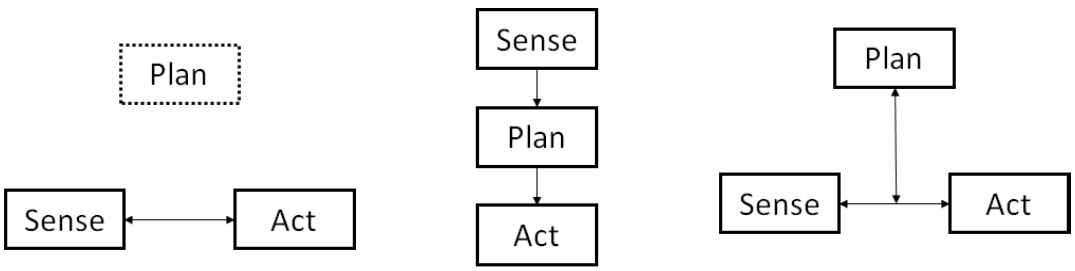
1.2.1 Sense-Model-Plan-Act and the different robotic paradigms

Like any system provided with Artificial Intelligence (AI), an autonomous vehicle relies on the four primitives of robotics presented in Table 1.2:

Sense	Gathers information from different sensors on the state of the robot and its surroundings. There exist two types of sensors: *proprioceptive sensors* which measure the internal state of the robot such as Inertial Measurement Units (IMU) and wheel encoders, and *exteroceptive sensors* which acquire information from the robot's environment like Global Positioning Systems (GPS), cameras, LIght Detection And Ranging (LIDAR) or RAdio Detection And Ranging (RADAR).
Model	Creates a world model from the sensor information, in order to determine (i) the position of the robot using Simulatenous Localization And Mapping (SLAM), (ii) the free-space where the robot can move and (iii) the intention of the dynamic obstacles.
Plan	Determines a goal for the robot using navigation and/or a behavioral layer and computes a safe path/trajectory to follow in order to reach it. A *path* consists in a list of positions $(X_i, Y_i)_{i \in [\![1,n]\!]}$ in the physical space while a *trajectory* is a path indexed by time (or equivalently, with a velocity encoded).
Act	Commands the actuators of the robot, namely the brake and gas pedals and the steering wheel in the case of an autonomous vehicle, in order to track the reference path/trajectory.

Table 1.2 – The four robotic primitives.

These primitives are organized according to a *paradigm*, defined as a philosophy or set of assumptions and/or techniques which characterizes an approach to a class of problems. There exist three main paradigms for robotic systems [Murphy, 2000]: the hierarchical paradigm, the reactive paradigm and the hybrid deliberative/reactive paradigm. Under the hierarchical paradigm, the robot performs at each step successively the Sense-Plan-Act tasks. The reactive paradigm relies on the Sense-Act tasks: the Plan task is relegated to the background, or even eliminated. Therefore, it has a faster execution time. Finally, the hybrid deliberative/reactive paradigm first performs the Plan task and then decomposes the planning goal into subtasks where the reactive paradigm Sense-Act is executed. The organization of the different paradigms are represented in Figure 1.4.



(a) Reactive paradigm (b) Hierarchical paradigm (c) Hybrid deliberative/reactive paradigm

Figure 1.4 – The different robotic paradigms [Murphy, 2000].

Remark 1. *Note that the primitive "Model" has been introduced only recently and is therefore not mentioned in some references, in particular in Figure 1.4.*

1.2.2 High-level cognitive tasks

High-level cognitive tasks require some kind of “intelligence" in order to be performed: they consists in tasks that humans are good at and that cannot be easily translated into algorithms.

Perception

Perception, which corresponds to the SENSE primitive of a robot, is very critical for the safety of an automous vehicle. It is responsible for gathering all the necessary information on the driving scene, *i.e.* the position and nature of the obstacles, the traffic signs, the drivable road corridors... This task is rather easy for a human but much harder for a computer as it requires to understand the scene, in particular its semantic: for example, to tell the difference between a biker and a pedestrian. As mentioned previously, it comprises proprioceptive and exteroceptive sensors. Usually, autonomous vehicles are equipped with four wheel encoders, one per wheel, to compute the rotational speed of each wheel; an IMU to get the three angular and three linear accelerations of the vehicle; a GPS to receive its position in the inertial coordinate system; one or several cameras to detect the lane marking, pedestrians, vehicles, traffic signs and other sementic information in the scene; one or several LIDARs and/or RADARs to obtain the distances to static and dynamic objets; ultrasounds for parking and tight maneuvers.

Simultaneous Localization And Mapping (SLAM)

After collecting information on the environment, the vehicle needs to build a local map of the scene, to understand where it is allowed to drive, what the other traffic participants are doing, and to localize itself accurately in the scene. This is done using Simultaneous Localization And Mapping (SLAM). These algorithms have been widely developed by the robotics community (see [Cadena et al., 2016] for a review of the different techniques). As the name indicates, they enable to localize the vehicle while creating a map of the environment, *i.e.* the roads, the static and dynamic obstacles. One of the main difficulty of such algorithms applied to autonomous vehicles compared to conventional robots is not only the large number of dynamic obstacles and thus of occlusions encountered, but also their higher velocities. Hence, SLAM requires to understand the semantic of the scene such as where is the vehicle allowed to drive and occlusions. It is thus a hard task for a computer.

Decision-making and planning

Finally, an autonomous vehicle needs to have a goal and plan how to get there. While the navigation task precomputes the route to get to the final destination, it is still necessary to compute local plans during the travel in order to perform lane change maneuvers (either to overtake a slow vehicle located in front or to take an exit on the highway), merging into a lane with other traffic participants, crossing a four-way stop and other similar actions. These maneuvers require a lot of decisions such as when it should do it and how, and thus a certain risk. Evaluating the risk can be rather difficult sometimes for a computer as it implies social behavior too and trade-off between drivers.

1.2.3 Low-level cognitive tasks

Unlike high-level cognitive tasks, low-level ones do not require any specific intelligence. These tasks are relatively easy to implement on computers and provide better results than human. Such tasks are either repetitive (keeping a constant speed for example), require a great accuracy or lot of computation.

Observation and filtering

In order to keep the costs low, some internal states of the vehicle such as its speed cannot be directly measured. Instead, observers stemming from control theory are used (see for example Luenberger observers for linear systems [Luenberger, 1971]). They provide an estimation of the internal state from the measurements of the input and output signals of the system. Moreover, due to the low costs of the sensors such as the IMU, the quality of the measurements is usually poor. Therefore, filters are necessary. In particular, Kalman filters [Kalman and Bucy, 1961] enable to improve the estimation of the internal states of the vehicle such as its localization or its velocity, despite model errors and measurement noises, by merging measurements provided by proprioceptive sensors such as the IMU with exteroceptive sensors such as the GPS.

Control and tracking

Controllers are ubiquitous in modern vehicles, in particular in autonomous one. They correspond to the ACT primitive. Their objective is to provide an input signal to the system, also referred to as a control law, in order to track a desired reference output while ensuring the stability of the system. They are used for example for generating the throttle to maintain a constant speed during cruise control, as well as for the different ADAS presented in section 1.1.2. Control laws are often simple to implement on a computer, in particular if a closed-loop form can be found, and are outperforming a human to track a reference signal. While the stability and the accuracy of the tracking are usually the most important criteria, some other ones can be considered too such as the smoothness of the control input, the time response or overshoots.

Observability and *controllability* are two dual concepts in control theory that are important in order to obtain a desired result. While adding new sensors to the system might improve its observability, adding new actuators might improve its controllability. In the case of autonomous vehicles, the actuators are the gas pedal, the brake pedal and the steering wheel, even though some systems such as the ESP can provide a differential torque to the wheels to improve the controllability of the vehicle. However, in some cases the system becomes not controllable: for example, during gear shift, the torque generated by the engine is not transmitted to the wheel as the clutch is opened. While this case is relatively non important regarding the vehicle safety, other cases are much more critical such as slip and skid, where the vehicle can loose control (the steering wheel has no impact on the direction of the vehicle anymore). These cases need to be avoided and will be explained in chapter 2.

1.3 Scope of the thesis

1.3.1 Main assumptions

This thesis deals with *autonomous vehicles*. Throughout this work, a vehicle will be said to be *autonomous* if it compels with the levels 4 or 5 defined by the SAE standards. In the literature, see for example [Qian et al., 2017], some authors distinguish an *autonomous* vehicle from a *cooperative* vehicle: in the first case, the vehicle relies solely on its own sensors while in the second case, the vehicle also gets information from other vehicles and the infrastructure, through respectively Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications. Such a distinction will not be made in this work. Therefore, an *autonomous* vehicle can rely both on its own sensors or on communication.

As research on autonomous vehicles is getting more and more mature, the question of consistency between its different layers, namely perception (SENSE), localization and mapping (MODEL), motion planning (PLAN) and control (ACT), is becoming crucial to ensure the safety of the vehicle at all time. An ill-designed vehicle architecture might be very critical for its safety, even though each layer is well-designed independently. Our work focuses on the *Plan-Act* primitives defined

in section 1.2.1, which will be referred to as respectively the motion planning and the low-level control tasks, in order to define a “good" motion planning and control architecture. Therefore, we assume that (i) all the information needed to take a decision is known (the environment is fully observable), (ii) the world model is known and given as input of the planning layer, (iii) a navigation task or a behavioral layer provides a goal to the vehicle. More precisely, the inputs comprise a map with the coordinates of the road centerlane and its boundaries, the reference lane assigned by the navigation task, and the position and velocity of all the obstacles. The detection of obstacles will not be considered at any stage during this work.

1.3.2 Some important notions

Control

The word *control* refers to the fact of bringing a system to a desired state by acting on its inputs. In control theory, it is common to consider single input single output (SISO) systems and to derive a closed-loop form for the control signal where stability proofs for the system can be demonstrated. In the robotic community, the large number of degrees of freedom and/or actuators compels to use “optimal control" formulation in order to find a solution, *i.e.* an optimization problem where the goal is to minimize/maximize a criterion while respecting some constraints on the system.

Feasibility

The word *feasibility* will play a major role in this work but its meaning is ambiguous and depends on the context. It is defined as “capable of being done or carried out"[21]. Among the motion planning community, people claim that their stated problem is feasible if there exists a mathematical solution: this means that there exists a collision-free path or trajectory between two configurations for example. This will be referred to as the *mathematical feasibility*. However, the solution obtained might not be feasible, this time in the physical sense, because it does not respect some laws of motion of the vehicle. This problem arises for example if the vehicle model used in the motion planning problem is not adapted to the situation. This will be referred to as the *dynamic feasibility*.

Consistency

The word *consistency* is probably the most important one in this thesis, or more precisely its antonym *inconsistency*. The Merriam-Webster Dictionary[22] defines the later as the quality or state of being: (i) not compatible with another fact or claim; (ii) containing incompatible elements; (iii) incoherent or illogical in thought or actions.

Inconsistency problems between the motion planning and control layers may arise due to some assumption mismatches, corresponding to definition (i) or (ii): for example, it is common to assume that the reference trajectory provided to the controller is C^2 in order to have a continuous curvature. However, if no precaution is taken, the motion planning has no reasons to deliver a trajectory with such properties (not even continuous)! However, we will focus on definition (iii) in this work: choosing an improper level of modeling when defining a motion planning or control problem might lead to some illogical actions.

Model

A model is a mathematical representation of a real world system. It enables not only to describe and understand its behavior but also to predict its evolution. Models play a fundamental role in a large variety of robotic applications such as simulation, control theory, motion planning or prediction intention. However, a model for a given system is neither unique nor universal. It is a

[21] https://www.merriam-webster.com/dictionary/feasible, visited on April 2018.
[22] https://www.merriam-webster.com/dictionary/inconsistent, visited on April 2018.

trade-off between conflicting objectives: the computational efficiency to solve it, its accuracy, its validity and the a priori knowledge of the system. One particularity about models is that they can be true but irrelevant for some applications, either because they are too complex for the application considered or because they do not model certain phenomena that cannot be neglected in this case. Therefore, defining a proper model [Ersal et al., 2008] is not straightforward and depends on the level of abstraction of the problem considered. Thus, when a model is used, it is important to have in mind the underlying assumptions that have been made. Otherwise, the mismatch with reality can be huge. Using a model that is *consistent* with the problem considered is therefore critical for the safety of the vehicle. In a sense, a model is like a drawing: the more details are included, the longer it takes to execute it.

1.3.3 Consistency of the architecture

In her inaugural lecture on “Algorithms" at College de France[23], Claire Mathieu underlines that an algorithm might lead to bad solutions not because the resolution is wrong but because the stated problem is not relevant or well-posed. In the case of an autonomous vehicle, such a problem may arise in the motion planning and control architecture due to *inconsistencies*. They can be found at three different levels: at the planning level, due to an improper level of modeling of the vehicle dynamics; at the control level, as its operational domain must be able to track at least all planned trajectories from the motion planner and also be robust to noise and disturbances; at the junction between the motion planning and control layers, due to some assumption mismatches between the two layers, such as the continuity or smoothness of the planned trajectory.

Motion planning and control problems are two different but highly related problems. The first one consists in computing a dynamically feasible trajectory for the vehicle, avoiding the surrounding obstacles such as other vehicles, pedestrians, or non moving objects. The second is acting on the actuators, *i.e.* the gas pedal, brake pedal and steering wheel, in order to track the trajectory obtained by the motion planner, while ensuring the stability of the system and, if possible, a smooth drive. Therefore, the properties of the design of each layer are quite different.

At the motion planning layer, the algorithm is searching a trajectory that respects some safety and comfort constraints while optimizing some criteria. Such a search is computationally expensive and requires therefore both a simple model of the vehicle and a low computational frequency (around 5-10Hz). On the contrary, control usually operates at high-frequency (around 100Hz) to ensure a good tracking of the reference trajectory. Moreover, the level of abstraction of the motion planning and the control layers also differs. For example, dealing with obstacles is one of the main task of the motion planner while the controller usually completely ignores them: the trajectory given by the motion planner is assumed to be safe within a certain margin and the goal of the controller is then to follow as well as possible the given trajectory, without considering the obstacles.

In most studies the low-level control is not considered at the motion planning phase; instead, the general assumption is that, provided the planned trajectory satisfies a certain set of constraints, the low-level control will be able to track this trajectory with a bounded error. In some actual implementations, the low-level controller is provided by the OEM’s test car as a black-box, with hard limits on the acceptable inputs of this control to prevent the vehicle from leaving its handling envelope. Such “black-box” behavior effectively fully isolates motion planning and control.

Conversely, research on low-level vehicle control often assumes that a predefined reference trajectory is known in advance and does not change over time. However, due to the presence of dynamic obstacles on the road, notably other traffic participants, this assumption is in general untrue.

[23] https://www.college-de-france.fr/site/claire-mathieu/inaugural-lecture-2017-11-16-18h00.htm, visited on May 2018

1.4 Contributions

In this thesis, we claim that the *intention of the motion planner, expressed as the future trajectory of the vehicle, should match with the one actually executed by the low-level controllers.*

In other words, the motion planning and control architecture should be *consistent* as defined previously by choosing proper models and studying the influence of the interaction between the two layers on the overall performance.

Therefore, we will investigate some fundamental questions concerning the safety of an autonomous vehicle, in particular on the guarantees that can be obtained. This starts at the motion planning level by answering the following question:

Question 1: Can we guarantee the kinematic and dynamic feasiblity of a trajectory computed by the motion planner?

Therefore, after a reminder about vehicle dynamics and modeling in chapter 2 and a review of state-of-the-art motion planning methods in chapter 3, chapter 4 deals with the question of consistency of the model used to plan a feasible trajectory avoiding obstacles while driving the vehicle towards its goal. In fact, the motion planner proposed is based on a kinematic bicycle model.

The second part of this thesis will focus on low-level control problems, stemming from control theory. As such, the algorithms described therein are not specific to an autonomous vehicle but can be used in the design of some ADAS such as an ACC or a LKS. Many longitudinal and lateral controllers have been proposed in the literature for such purposes. However, these controllers have an operational limit: thus, chapter 5 will recall some of the commonly used lateral controllers and answer the following critical question regarding safety:

Question 2: Given a reference trajectory, can we ensure the performance of a lateral controller?

Finally, the last part of this thesis will focus on a newly introduced model-free control paradigm, proposed by [Fliess and Join, 2013], and its application to vehicle control. Due to the complexity of vehicle dynamics, in particular of tire dynamics, and the large number of parameters that not only vary from one situation to another but also are difficult to estimate, the model-free control appears as a promising solution. One reason therefore is that it estimates in real-time the dynamics of the vehicle. However, the following questions need to be addressed:

Question 3: How to implement model-free controllers on an autonomous vehicle? And what guarantees can we have on the results?

We will try to give an answer to these questions in chapter 6. This chapter is completely independent from the rest of this thesis.

Part I

Modeling of the motion of a vehicle

Chapter 2

Vehicle dynamics and motion modeling

" A theory has only the alternative of being right or wrong.
A model has a third possibility: it may be right, but irrelevant "

Manfred Eigen

Contents

A physical system can be represented by a wide range of models, each one having some assets and drawbacks. The choice of a model is usually a trade-off between its accuracy, its computational efficiency, its robustness to modeling errors or uncertain parameters, and the level of abstraction necessary. The more detailed the model is, the more computational time is needed to solve it. Moreover, more details often require more information about the system: therefore, if only a few vehicle parameters are known, a complex model that requires a lot of information about the vehicle might be inappropriate, in particular if the model is not very robust to parameter uncertainties. Conversely, a detailed model can capture some phenomena that should be taken into account in certain situations, for instance slip and skid in the case of an autonomous vehicle.

In this chapter, we are interested in developing a high-fidelity vehicle simulator for testing our motion planning and control algorithms. The model used in the simulator needs to be as accurate as possible, taking into account phenomena such as slip or skid. The computational time is not the main issue and we suppose that we know all the information necessary about the vehicle. Therefore, a review of the different dynamic models which are the only models that comprise slip and skid are presented in a first time. The dynamics of a vehicle can be decomposed into two parts: a slowly-evolving dynamic (characteristic time of 0.1s to 1s) corresponding to the movement of the carbody and a fast dynamic (characteristic time of 1 to 10ms) corresponding to the movement of the wheels/tires. They will be presented respectively in secion 2.1 and 2.2. Section 2.3 presents the vehicle simulator we have derived from the dynamic models and that will be used in the next chapters. Finally, section 2.4 introduces some simpler models that are used for motion planning and control.

Most of the explanations given in this chapter as well as complementary informations on vehicle dynamics can be found in [Rajamani, 2012, Kiencke and Nielsen, 2005, Gillespie, 1997]. All the main notations are given in Tables 2.1, 2.2, 2.3, 2.4, 2.5 and 2.6. Figures 2.1 and 2.2 show the three different frames that will be used in the rest of this chapter: the inertial frame $(\vec{e}_X, \vec{e}_Y, \vec{e}_Z)$ associated to the ground, the vehicle frame $(\vec{e}_x, \vec{e}_y, \vec{e}_z)$ and the pneumatic/tire frame $(\vec{p}_{xi}, \vec{p}_{yi}, \vec{p}_{zi})$ associated to each wheel $i = 1..4$.

Figure 2.1 – Inertial frame $(\vec{e}_X, \vec{e}_Y, \vec{e}_Z)$ and vehicle frame $(\vec{e}_x, \vec{e}_y, \vec{e}_z)$.

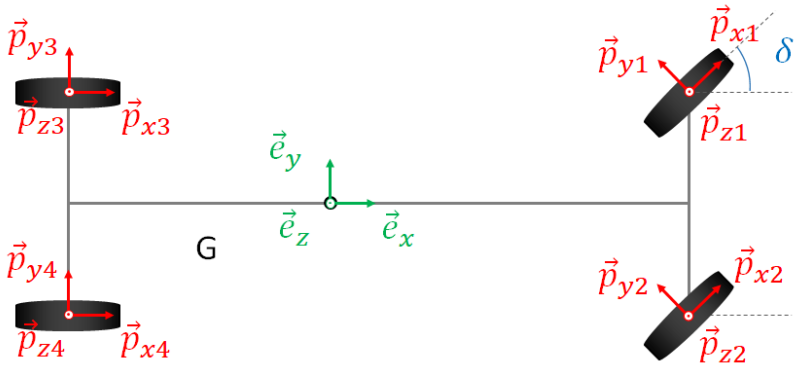


Figure 2.2 – Vehicle frame $(\vec{e}_x, \vec{e}_y, \vec{e}_z)$ and pneumatic frame $(\vec{p}_{xi}, \vec{p}_{yi}, \vec{p}_{zi})$ for each wheel $i = 1..4$.

Variables	Meaning
M_T	Total mass of the vehicle
M_S	Suspended mass of the vehicle
I_x, I_y, I_z	Inertia of the vehicle around its roll, pitch and yaw axis
l_f, l_r	Distance between the front (resp. rear) axle and the center of gravity
l_w	Half-track of the vehicle
h_{CoG}	Height of the center of gravity of the vehicle

Table 2.1 – Notations for vehicle parameters.

Variables	Meaning
X, Y	Position of the center of gravity of the vehicle in the inertial frame $(\vec{e}_X, \vec{e}_Y, \vec{e}_Z)$
θ, ϕ, ψ	Roll, pitch and yaw angles of the carbody
V_x, V_y, V_z	Longitudinal, lateral and vertical speed of the vehicle in its own frame $(\vec{e}_x, \vec{e}_y, \vec{e}_z)$
β	Slip angle of the vehicle at center of gravity
δ_f	Steering angle of the wheels located on the front axle
δ_f	Steering angle of the wheels located on the rear axle
T_{ω_i}	Total torque applied to wheel i
T_{mi}	Motor torque applied to wheel i
T_{bi}	Brake torque applied to wheel i
T_{tot}	Total torque applied to all the wheels

Table 2.2 – Notations for states and inputs of the vehicle.

Variables	Meaning
μ	Friction coefficient of the road
μ_c	Coulomb friction coefficient of the road
μ_s	Static friction coefficient of the road
p_x, p_y	Slope and road-bank angle of the road

Table 2.3 – Notations for road characteristics.

Variables	Meaning
I_r	Inertia of the wheel
m_r	Mass of the wheel
r_{stat}	Radius of the wheel when the vehicle is static
r_0	Radius of the wheel without any load
r_{eff}	Effective radius of the wheel
ω_i	Angular velocity of the wheel i
V_{xpi}	Longitudinal speed of the center of rotation of wheel i expressed in the tire frame $(\vec{p}_{xi}, \vec{p}_{yi}, \vec{p}_{zi})$
τ_{xi}	Longitudinal slip ratio at wheel i
α_i	Slip angle at wheel i
τ_{total}	Total (longitudinal+ lateral) slip ratio at a wheel
γ	Camber angle

Table 2.4 – Notations for states and characteristics of the wheels.

Variables	Meaning
F_{xpi}, F_{ypi}	Longitudinal and lateral tire forces generated by the road on the wheel i expressed in the tire frame $(\vec{p}_{xi}, \vec{p}_{yi}, \vec{p}_{zi})$
F_{xi}, F_{yi}	Longitudinal and lateral tire forces generated by the road on the wheel i expressed in the vehicle frame $(\vec{e}_x, \vec{e}_y, \vec{e}_z)$
F_{zi}	Normal reaction forces on wheel i
ΔF_{si}	Additional spring force of the suspension of wheel i compared to the situation where the suspended carbody is at equilibrium
F_{aero}	Aerodynamic drag forces applied on the vehicle
g	Gravitational constant

Table 2.5 – Notations for forces and torques.

Variables	Meaning
z_{ti}	Heigth of wheel i in the inertial frame
z_{si}	Heigth of the suspension of wheel i in the inertial frame
z_{ri}	Heigth of the road at wheel i in the inertial frame
k_s	Stiffness coefficient of the spring between the tire and the suspended carbody
k_t	Stiffness coefficient of the tire
d_s	Damping coefficient between the tire and the suspended carbody

Table 2.6 – Notations for suspensions.

2.1 Carbody dynamics - the slow dynamics

In this section, we consider two dynamic models of the carbody of the vehicle, namely the dynamic bicycle model and the four-wheel vehicle model. We are interested in the evolution of the state of the carbody, which is composed of up to six different variables, namely the longitudinal V_x, lateral V_y and vertical V_z velocities and the roll $\dot{\theta}$, pitch $\dot{\phi}$ and yaw $\dot{\psi}$ angular velocities. The inputs of the models are the longitudinal and lateral forces F_{xi} and F_{yi} applied by the road on the different wheels i in the vehicle frame (or equivalently F_{xpi} and F_{ypi} in the pneumatic frame) as shown in Figure 2.3.

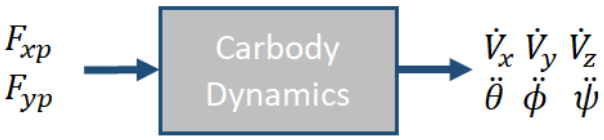


Figure 2.3 – Modeling of the carbody dynamics.

The force generated by the road on the wheel is denoted by $F_{..}$, where the first subscript $\{x, y, z\}$ indicates the direction of the force and the second one the wheel i. By default, the forces are expressed in the vehicle frame. Forces expressed in the pneumatic frame are denoted by an extra subscript p. For the wheel subscript i, we consider $i = 1..4$ (or equivalently $i \in \{fl, fr, rl, rr\}$) if we consider a four-wheel vehicle model and $i = 1,2$ (or equivalently $i \in \{f, r\}$) if we consider a bicycle model. Note that fl, fr, rl, rr denotes respectively the front-left, front-right, rear-left and rear-right wheels, and f, r the front and rear wheels.

Examples: F_{xi} *refers to the longitudinal force applied on wheel* i *in the vehicle frame and* F_{ypi} *refers to the lateral force applied on wheel* i *in the pneumatic frame.*

All the models of the carbody dynamics presented afterwards are expressed in the vehicle frame $(\vec{e}_x, \vec{e}_y, \vec{e}_z)$. However, the position (X,Y) of the vehicle in the inertial frame $(\vec{e}_X, \vec{e}_Y, \vec{e}_Z)$ can be obtained by Equations (2.1):

$$\dot{X} = V_x \cos\psi - V_y \sin\psi \tag{2.1a}$$
$$\dot{Y} = V_x \sin\psi + V_y \cos\psi \tag{2.1b}$$

2.1.1 Dynamic bicycle model

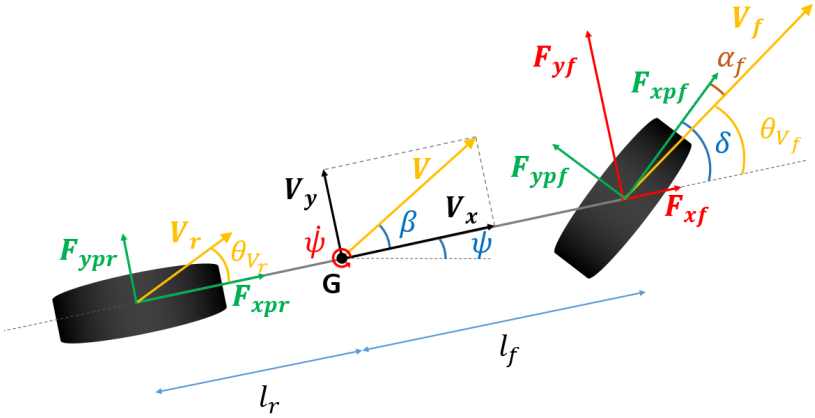


Figure 2.4 – The dynamic bicycle model.

In this subsection, we assimilate the vehicle to a bicycle model, where the two front wheels (resp. rear wheels) are lumped into a unique wheel, located at the center of the front axle (resp.

rear axle) as illustrated in Figure 2.4. In that case, the fundamental principle of dynamics applied to the vehicule leads to Equations (2.2) (see Figure 2.5). The expression of the aerodynamic drag forces is given by Equation (2.3).

$$\begin{aligned} M_T(\dot{V}_x - \dot{\psi}V_y) &= F_{xf} + F_{xr} - F_{aero} - M_T g \sin p_x && (2.2a)\\ M_T(\dot{V}_y + \dot{\psi}V_x) &= F_{yf} + F_{yr} && (2.2b)\\ I_z\ddot{\psi} &= l_f F_{xf} - l_r F_{xr} && (2.2c) \end{aligned}$$

$$F_{aero} = \frac{1}{2}\rho_{air} C_x S (V_x + W)^2 \quad (2.3)$$

where g is the gravitational constant, ρ_{air} is the mass density of air, C_x the aerodynamic drag coefficient, S the frontal area of the vehicle and W the speed of the wind in the longitudinal direction.

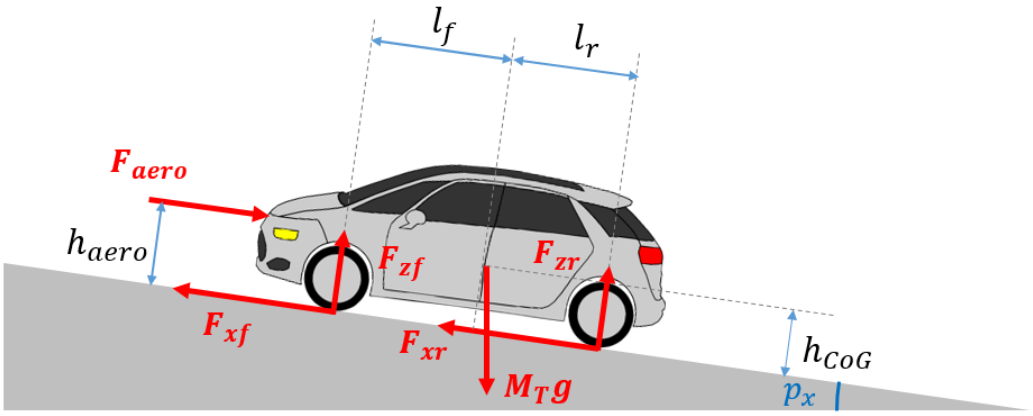


Figure 2.5 – Forces applied on a vehicle.

In order to obtain Equations (2.2), the following assumptions were made:

- The road-bank angle is neglected.
- The pitch and roll motions as well as the vertical dynamics of the vehicle are neglected.

Therefore, one of the main characteristics of this model is that it does not take into account the lateral load transfer between wheels, which has an impact on the tire forces that can be generated as we will see in section 2.2.

At last, Equations (2.4) show how to transform the tire forces expressed in the pneumatic frame to tire forces expressed in the vehicle frame:

$$\begin{aligned} F_{xf} &= F_{xpf}\cos\delta_f - F_{ypf}\sin\delta_f && (2.4a)\\ F_{xr} &= F_{xpr} && (2.4b)\\ F_{yf} &= F_{ypf}\cos\delta_f + F_{xpf}\sin\delta_f && (2.4c)\\ F_{yr} &= F_{ypr} && (2.4d) \end{aligned}$$

2.1.2 Four-wheel vehicle model

We consider again a four-wheel vehicle model as shown in Figure 2.6. Compared to the dynamic bicycle model introduced by Equation (2.2), both the slope and the road-bank angle are considered, as well as the roll, pitch and vertical motions. The full demonstrations of Equations (2.5) and (2.6) are given respetively in Appendices A.1 and A.2.

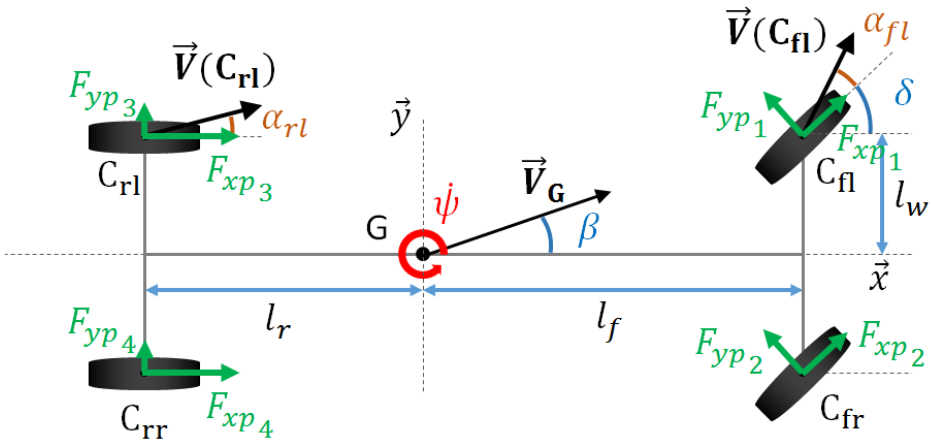


Figure 2.6 – Four-wheel vehicle model.

The evolution of the longitudinal, lateral and vertical velocities of the vehicle frame are ruled by the fundamental principle of dynamics:

$$M_T(\dot{V}_x - \dot{\psi}V_y) = \sum_{i=1}^{4} F_{xi} - F_{aero}\cos\phi + M_T g\sin(\phi - p_x) \tag{2.5a}$$

$$M_T(\dot{V}_y + \dot{\psi}V_x) = \sum_{i=1}^{4} F_{yi} - M_T g\sin(\theta - p_y)\cos(\phi - p_x) \tag{2.5b}$$

$$M_S(\dot{V}_z - \dot{\phi}V_x) = \sum_{i=1}^{4} F_{zi} - M_S g\cos(\theta - p_y)\cos(\phi - p_x) + F_{aero}\sin\phi \tag{2.5c}$$

where F_{aero} is given by Equation (2.3). Note that the crossterms θV_z, ϕV_z and θV_y have been neglected in Equations (2.5).

The dynamics of the roll, pitch and yaw angular velocities in the inertial frame are given by:

$$I_x\ddot{\theta} = l_w(\Delta F_{s1} + \Delta F_{s3} - \Delta F_{s2} - \Delta F_{s4}) + \sum_{i=1}^{4} h_{CoG}F_{yi} \tag{2.6a}$$

$$I_y\ddot{\phi} = -l_f(\Delta F_{s1} + \Delta F_{s2}) + l_r(\Delta F_{s3} + \Delta F_{s4}) - \sum_{i=1}^{4} h_{CoG}F_{xi} + h_{CoG}F_{aero} \tag{2.6b}$$

$$I_z\ddot{\psi} = l_f(F_{y1} + F_{y2}) - l_r(F_{y3} + F_{y4}) + l_w(F_{x2} + F_{x4} - F_{x1} - F_{x3}) \tag{2.6c}$$

The variation of the suspension force ΔF_{si} applied at wheel i depends on the variation of the length of the suspension $\Delta z_{si} = z_{si} - z_{ti}$ (see Equation (2.7a)) which itself depends only on the roll θ and pitch ϕ angles (see Equation (2.7b)). The normal reaction F_{zi} force applied by the road on wheel i is given by Equation (2.7c). The forces acting on the suspension are displayed in Figure 2.7 where P_r is the weight of the wheel (see Appendix A.3 for more details).

$$\Delta F_{si} = -k_s\Delta z_{si}(\theta,\phi) - d_s(\Delta\dot{z}_{si}(\theta,\phi)) \tag{2.7a}$$

$$\Delta z_{si}(\theta,\phi) = \epsilon_i l_w\sin\theta - l_i\cos\theta\sin\phi \tag{2.7b}$$

$$F_{zi} = F_{zi_0} + \Delta F_{si} \tag{2.7c}$$

where $\epsilon_i = \begin{cases} 1 & \text{if } i = \{2,4\} \\ -1 & \text{if } i = \{1,3\} \end{cases}$ and $l_i = \begin{cases} l_f & \text{if } i = \{1,2\} \\ -l_r & \text{if } i = \{3,4\} \end{cases}$.

The nominal normal reaction force F_{zi_0} at equilibrium applied on wheel i is given by Equation (2.8). The demonstration is given in Appendix A.4.

$$F_{zi_0} = \begin{cases} \frac{l_r M_T g}{2(l_f+l_r)} & \text{if } i = \{1,2\} \\ \frac{l_f M_T g}{2(l_f+l_r)} & \text{if } i = \{3,4\} \end{cases} \tag{2.8}$$

Remark 2. *The dynamics of the suspensions are ruled by Equation (2.9) but are not necessary for the modeling:*

$$M_s \ddot{z}_s = -k_s(z_s - z_t) - d_s(\dot{z}_s - \dot{z}_t) \tag{2.9}$$

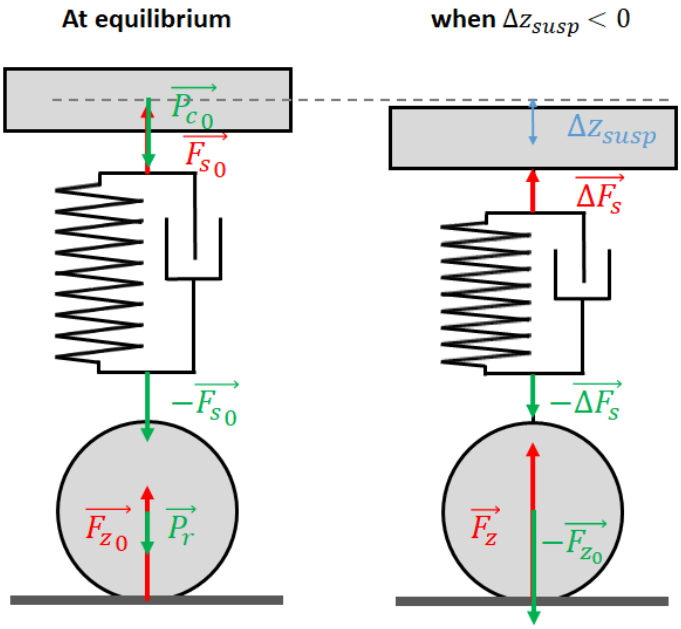


Figure 2.7 – Variation of the suspension forces: $\vec{P}_r$ is the wheel weight, $\vec{F}_{z_0}$ is the normal reaction force of the road at equilibrium, $\vec{P}_{c_0}$ is the weight of a quarter carbody at equilibrium, $\vec{F}_{s_0}$ is the suspension spring and damping force at equilibrium, $\Delta\vec{F}_s$ the variation from equilibrium of the spring and damping forces.

At last, Equations (2.10) show how to transform the tire forces expressed in the pneumatic frame to ones expressed in the vehicle frame:

$$F_{xi} = (F_{xpi}\cos\delta_i - F_{ypi}\sin\delta_i)\cos\phi - F_{zi}\sin\phi \tag{2.10a}$$

$$F_{yi} = (F_{xpi}\cos\delta_i - F_{ypi}\sin\delta_i)\sin\theta\sin\phi + (F_{ypi}\cos\delta_i + F_{xpi}\sin\delta_i)\cos\theta + F_{zi}\sin\theta\cos\phi \tag{2.10b}$$

2.2 Tire dynamics - the fast dynamics

In the previous section, we have seen how the longitudinal F_{xp} and lateral F_{yp} tire forces act on the carbody dynamics. However, these forces are not given directly by the driver (or controller in the case of an autonomous vehicle) as inputs. Therefore, we still need to model how the actions on the steering wheel, the gas pedal and the brake pedal generate these tire forces.

We consider more precisely the inputs of the driver to be the steering angle δ_f at the front wheels and the motor T_{mi} and the brake T_{bi} torques at each wheel i. Considering the wheel torques avoids going through the modeling of the internal combustion engine and the brake systems. Moreover electric and gas vehicles can then be handled indifferently. Note that the steering angle at the front wheel is directly proportional to the steering wheel through the steering column.

The longitudinal F_{xp} and lateral F_{yp} forces generated by the road on each tire expressed in the pneumatic frame are obtained from four variables: the longitudinal slip ratio τ_x, the lateral slip angle α, the normal reaction force F_z of the road on the wheel, and the road friction coefficient μ. Therefore, tire models can be expressed in the form of Equations (2.11) (see Figure 2.8).

$$F_{xp} = f_x(\tau_x, \alpha, F_z, \mu) \tag{2.11a}$$

$$F_{yp} = f_y(\alpha, \tau_x, F_z, \mu) \tag{2.11b}$$

Figure 2.8 – Modeling of tire dynamics.

The longitudinal slip ratio τ_{xi} of wheel i describes how much the wheel slips in the longitudinal direction of the wheel. It depends on the sliding velocity of the wheel on the road, *i.e.* $r_{eff}\omega_i - V_{xpi}$, which is the difference between the wheel speed $r_{eff}\omega_i$ at contact with the road and the actual longitudinal speed of the wheel V_{xpi} expressed in the tire frame $(\vec{p}_{xi}, \vec{p}_{yi}, \vec{p}_{zi})$ (see Figure 2.9). The longitudinal slip ratio τ_{xi} belongs to $[-1;1]$, where $\tau_{xi} = -1$ corresponds to the situation where $r_{eff}\omega_i > 0$ and $V_{xpi} < 0$, and $\tau_{xi} = 1$ to the situation where $r_{eff}\omega_i < 0$ and $V_{xpi} > 0$. It is defined by Equation (2.12).

$$\tau_{x_i} = \begin{cases} \frac{r_{eff}\omega_i - V_{xpi}}{r_{eff}|\omega_i|} & \text{if } r_{eff}\omega_i \geq V_{xpi} \quad \text{(Traction phase)} \\ \frac{r_{eff}\omega_i - V_{xpi}}{|V_{xpi}|} & \text{if } r_{eff}\omega_i < V_{xpi} \quad \text{(Braking phase)} \end{cases} \tag{2.12}$$

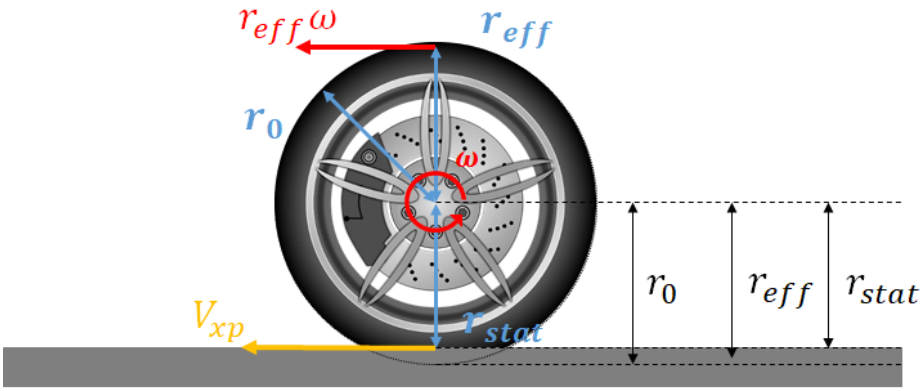


Figure 2.9 – Slip ratio and the different wheel radius presented in Table 2.4.

The lateral slip angle α_i is the angle between the orientation vector of the wheel i and the velocity vector of the same wheel as shown in Figure 2.10. Its expression is given by Equation (2.13) (see Appendix A.5 for demonstration):

$$\alpha_i = \begin{cases} \delta_f - \text{atan}\left(\frac{V_y + l_f\dot{\psi}}{V_x + \epsilon_i l_w \dot{\psi}}\right) & \text{if } i \in \{1,2\} \text{ (front wheels)} \\ -\text{atan}\left(\frac{V_y - l_r\dot{\psi}}{V_x + \epsilon_i l_w \dot{\psi}}\right) & \text{if } i \in \{3,4\} \text{ (rear wheels)} \end{cases} \tag{2.13}$$

In the case of a dynamic bicycle model, Equation (2.13) becomes:

$$\alpha_f = \delta_f - \text{atan}\left(\frac{V_y + l_f\dot{\psi}}{V_x}\right) \tag{2.14}$$

$$\alpha_r = -\text{atan}\left(\frac{V_y - l_r\dot{\psi}}{V_x}\right) \tag{2.15}$$

The normal reaction force $\vec{F}_z$ depends directly on the load transfer between wheels as illustrated in Figure 2.7: if the weight on a wheel increases, the normal reaction force increases accordingly. This has a strong impact on the magnitude of the combined forces $\vec{F}_{xp}$ and $\vec{F}_{yp}$ that can be generated in the road plane due to the laws of friction given by Equation (2.16). Therefore, the

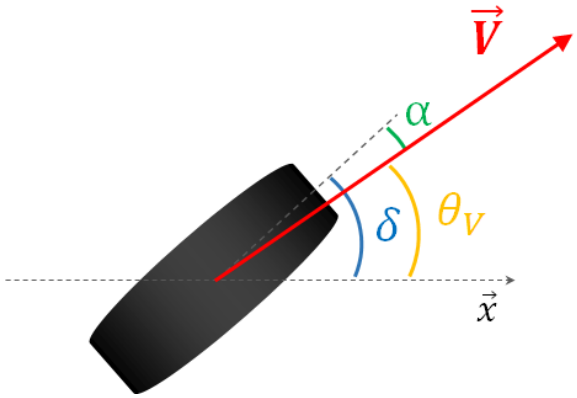


Figure 2.10 – Slip angle: the angle θ_V depends on the position of the wheel and on V_x, V_y and ψ.

norm of the normal reaction force $||\vec{F}_z||$ and the road friction coefficient μ define a friction circle in which the resulting force $\vec{F}_{xp} + \vec{F}_{yp}$ has to remain inside (see Figure 2.11).

$$||\vec{F}_{xp} + \vec{F}_{yp}|| \leq \mu ||\vec{F}_z|| \tag{2.16}$$

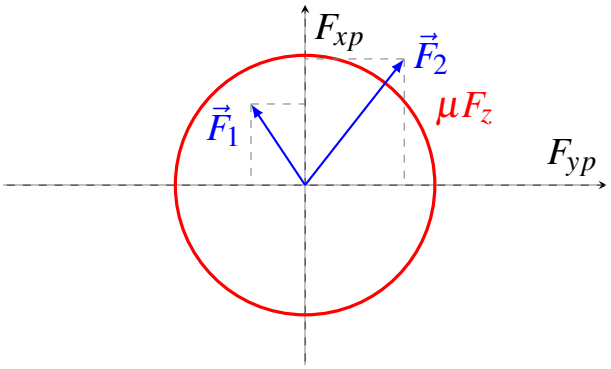


Figure 2.11 – Friction circle: the laws of Coulomb impose that $\vec{F} = \vec{F}_{xp} + \vec{F}_{yp}$ remains inside the red circle of radius μF_z. Therefore, $\vec{F}_1$ is an admissible force while $\vec{F}_2$ is impossible to generate.

Hence, in order to use a tire model, it is necessary to first compute the longitudinal slip ratio τ_x and the lateral slip angle α from the inputs of the vehicle, namely δ_f, T_m and T_b. While Equation (2.13) gives a clear relation between the steering angle δ_f and the different slip angles α_i, the relation between the wheel torques T_{mi} and T_{bi} and the different slip ratios τ_{xi} is more subtle: both Equation (2.12) and the equation of the wheel dynamics given by Equation (2.17) for wheel i should be considered.

$$I_r \dot{\omega}_i = T_{mi} - T_{bi} - r_{eff} F_{xpi} \tag{2.17}$$

Taking into account Equations (2.5), (2.6) and (2.17) with a tire model leads to the 10 Degrees of Freedom vehicle model (10 DoF). For a better accuracy, the dynamics of the tire deflection for each wheel given by Equation (2.18) can also be added to the model; this leads to the 14 Degrees of Freedom (14 DoF) vehicle model. In that case, the wheel/road interaction is represented by a spring with a very high stiffness coefficient k_t as shown in Figure 2.12.

$$m_t \ddot{z}_t = -k_t(z_t - z_r) - k_s(z_s - z_t) - d_s(\dot{z}_s - \dot{z}_t) \tag{2.18}$$

Remark 3. *As $I_r << \{I_x, I_y, I_z\}$, the dynamic of the wheel in Equation (2.17) is much faster than the one of the carbody. This has a strong impact on simulation and control problems as the time response of the wheel is much faster than the one of the carbody. Therefore, it is necessary to choose*

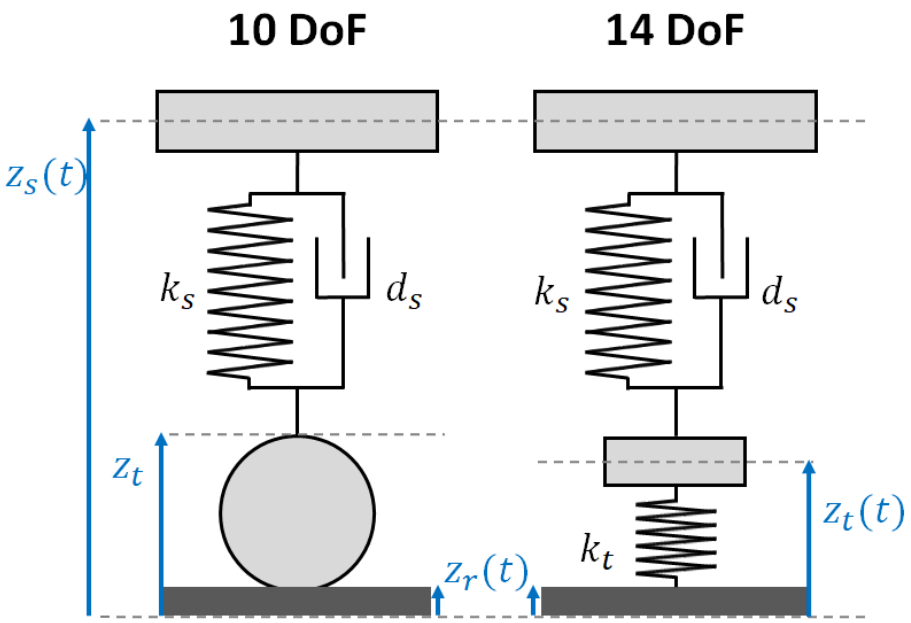


Figure 2.12 – Differences between the 10 DoF and 14 DoF vehicle model (on a quarter vehicle model).

a small time step (around 1 *ms) for simulations. For control purposes, singular perturbation techniques such as Tikhonov's theorem can be useful: see [Khalil, 1996] or [Kokotović et al., 1999] for more details. Moreover, note that the wheel dynamics is directly responsible for slip and skid; therefore, in order to improve the vehicle stability, it is necessary to monitor and control the wheels using for example an Anti-lock Braking System (ABS).*

In order to model the functions f_x and f_y of Equations (2.11), a wide range of tire models can be found in the literature. Each model has its own assets and drawbacks, in particular regarding accuracy and complexity as mentioned in introduction of this chapter. While some models such as the complete Pacejka's model or the LuGre are only suited for simulation purpose due to their complexity, other models are well suited for control application (in particular the linear tire model) but have a limited validity range. We will present the different models in the rest of this section.

2.2.1 Pacejka's Magic Formula

One of the most widespread modeling for tire/road interaction is the Pacejka model, also referred to as the "Pacejka's Magic Formula". It is a semi-empirical model as it is based partly on an analytical model assuming a parabolic distribution of the tire forces on the tire/road contact zone and partly on phenomenological observations of trial results [Pacejka, 2002].

Pure slip

In its simplest form, the longitudinal tire force F_{xp} only depends on the longitudinal slip ratio τ_x and the lateral tire force F_{yp} only depends on the lateral slip angle α. This model is valid only in the case of a pure longitudinal or lateral slip, which is usually the case in normal driving situations.

In this case, Equation (2.11) is modeled as following [Pacejka and Besselink, 1997]:

$$F_{xp} = D_x \sin\left(C_x \arctan(B_x\tau_x - E_x(B_x\tau_x - \arctan(B_x\tau_x)))\right) + S_{vx} \quad (2.19a)$$

$$F_{yp} = D_y \sin\left(C_y \arctan(B_y\alpha_y - E_y(B_y\alpha_y - \arctan(B_y\alpha_y)))\right) + S_{vy} \quad (2.19b)$$

In a more general form, with Y being the longitudinal (resp. lateral) tire force and X the longi-

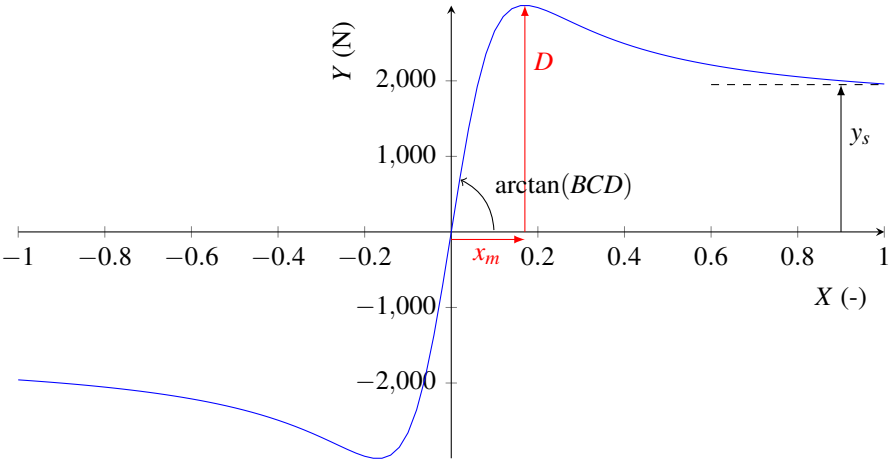


Figure 2.13 – Pacejka model in the case of pure slip.

tudinal slip ratio (resp the slip angle):

$$y = \mathrm{D}\sin(\mathrm{C}\arctan(\mathrm{B}x - \mathrm{E}(\mathrm{B}x - \arctan(\mathrm{B}x)))) \tag{2.20a}$$
$$\mathrm{Y} = y + \mathrm{S}_v \tag{2.20b}$$
$$x = \mathrm{X} + \mathrm{S}_h \tag{2.20c}$$

where the different constants, namely the stiffness factor B, the shape factor C, the peak value D and the curvature factor E can be determined as following (see Figure 2.13):

1. D is the peak value of the force;
2. $\mathrm{C} = 2 - \frac{2}{\pi}\arcsin(\frac{y_s}{\mathrm{D}})$ with y_s the asymptotic value of y;
3. B is determined through the slope at origin equal to BCD;
4. $\mathrm{E} = \frac{\mathrm{B}x_m - \tan(\frac{\pi}{2\mathrm{C}})}{\mathrm{B}x_m - \arctan(\mathrm{B}x_m)}$ where x_m is the slip value where the peak is reached, *i.e.* $y(x_m) = \mathrm{D}$;

Finally, S_h and S_v are respectively the horizontal and vertical shift.

In reality, these “constants" are not real constants and depend on some other parameters, such as the normal reaction force F_z (ans thus the mass transfer $\mathrm{dF}_z = \frac{\mathrm{F}_z - \mathrm{F}_{z_0}}{\mathrm{F}_{z_0}}$), the road friction coefficient μ, and the camber angle γ as explained in [Pacejka, 2002] (see Equations (2.21) and (2.22)).

The model given by Equation (2.20) also applies to the self-aligning torque M_z, with Y being the self-aligning torque and X the slip angle. A finer modeling can be obtained using Equation (2.23). However, the self-aligning torque will not be considered in this thesis.

- Longitudinal forces (pure longitudinal slip)

$$\mathrm{S}_{\mathrm{H}x} = p_{\mathrm{H}x_1} \tag{2.21a}$$
$$\tau_{shift} = \tau_x + \mathrm{S}_{\mathrm{H}x} \tag{2.21b}$$
$$\mathrm{C}_x = p_{\mathrm{C}x_1} \tag{2.21c}$$
$$\mu_x = \mu(p_{\mathrm{D}x_1} + p_{\mathrm{D}x_2}\mathrm{dF}_z) \tag{2.21d}$$
$$\mathrm{D}_x = \mu_x \mathrm{F}_z \tag{2.21e}$$
$$\mathrm{E}_x = (p_{\mathrm{E}x_1} + p_{\mathrm{E}x_2}\mathrm{dF}_z + p_{\mathrm{E}x_3}\mathrm{dF}_z{}^2)(1 - p_{\mathrm{E}x_4} sign(\tau_{shift_x})) \tag{2.21f}$$
$$\mathrm{K}_{xk} = \mathrm{F}_z(p_{\mathrm{K}x_1} + p_{\mathrm{K}x_2}\mathrm{dF}_z)\exp^{(p_{\mathrm{K}x_3}\mathrm{dF}_z)} \tag{2.21g}$$
$$\mathrm{B}_x = \frac{\mathrm{K}_{xk}}{\mathrm{C}_x \mathrm{D}_x} \tag{2.21h}$$
$$\mathrm{S}_{\mathrm{V}x} = \mathrm{F}_z(p_{\mathrm{V}x_1} + p_{\mathrm{V}x_2}\mathrm{dF}_z) \tag{2.21i}$$

$$\mathrm{F}_{xp_0} = \mathrm{D}_x \sin\left(\mathrm{C}_x \arctan(\mathrm{B}_x\tau_{shift} - \mathrm{E}_x(\mathrm{B}_x\tau_{shift} - \arctan(\mathrm{B}_x\tau_{shift})))\right) + \mathrm{S}_{\mathrm{V}x} \tag{2.21j}$$

- Lateral forces (pure lateral slip)

$$
\begin{aligned}
S_{Hy} &= (p_{Hy_1} + p_{Hy_2} dF_z) + p_{Hy_3}\gamma && (2.22a)\\
\alpha_{shift} &= \alpha + S_{Hy} && (2.22b)\\
C_y &= p_{Cy1} && (2.22c)\\
\mu_y &= \mu(p_{Dy_1} + p_{Dy_2} dF_z)(1 - p_{Dy_3}\gamma^2) && (2.22d)\\
D_y &= \mu_y F_z && (2.22e)\\
E_y &= (p_{Ey_1} + p_{Ey_2} dF_z)(1 - (p_{Ey_3} + p_{Ey_4}\gamma sign(\alpha_{shift}))) && (2.22f)\\
K_{ya_0} &= p_{Ky_1} F_{z_0} \sin(2\arctan(F_z/(p_{Ky_2} F_{z_0}))) && (2.22g)\\
K_{ya} &= K_{ya_0}(1 - p_{Ky_3}\gamma^2) && (2.22h)\\
B_y &= \frac{K_{ya}}{C_y D_y} && (2.22i)\\
S_{Vy} &= F_z((p_{Vy_1} + p_{Vy_2} dF_z) + (p_{Vy_3} + p_{Vy_4} dF_z)\gamma) && (2.22j)\\
\\
F_{yp_0} &= D_y \sin(C_y \arctan(B_y\alpha_{shift} - E_y(B_y\alpha_{shift} - \arctan(B_y\alpha_{shift})))) + S_{Vy} && (2.22k)
\end{aligned}
$$

- Self-aligning torques (pure lateral slip)

$$
\begin{aligned}
S_{Ht} &= q_{Hz_1} + q_{Hz_2} dF_z + (q_{Hz_3} + q_{Hz_4} dF_z)\gamma && (2.23a)\\
\alpha_t &= \alpha + S_{Ht} && (2.23b)\\
B_t &= (q_{Bz_1} + q_{Bz_2} dF_z + q_{Bz_3} dF_z^2)(1 + q_{Bz_4}\gamma + q_{Bz_5}|\gamma|) && (2.23c)\\
C_t &= q_{Cz_1} && (2.23d)\\
D_{t_0} &= F_z\left(\frac{r_0}{F_{z_0}}\right)(q_{Dz_1} + q_{Dz_2} dF_z) && (2.23e)\\
D_t &= D_{t_0}(1 + q_{Dz_3}\gamma + q_{Dz_4}\gamma^2) && (2.23f)\\
E_t &= \left(q_{Ez_1} + q_{Ez_2} dF_z + q_{Ez_3} dF_z^2\right)\left(1 + (q_{Ez_4} + q_{Ez_5}\gamma)\frac{2}{\pi}\arctan(B_t C_t \alpha_t)\right) && (2.23g)\\
t_0 &= D_t \cos(C_t \arctan(B_t\alpha_t - E_t(B_t\alpha_t - \arctan(B_t\alpha_t)))) \cos(\alpha) && (2.23h)\\
M'_{z_0} &= -t_0 F_{y_0} && (2.23i)\\
\\
\epsilon_k &= 0.01 && (2.23j)\\
K'_{ya} &= K_{ya} + \epsilon_k && (2.23k)\\
S_{Hf} &= S_{Hy} + \frac{S_{Vy}}{K'_{ya}} && (2.23l)\\
B_r &= q_{Bz_{10}} B_y C_y && (2.23m)\\
C_r &= 1 && (2.23n)\\
D_r &= F_z r_0((q_{Dz_6} + q_{Dz_7} dF_z) + (q_{Dz_8} + q_{Dz_9} dF_z)\gamma)\cos(\alpha) && (2.23o)\\
\alpha_r &= \alpha + S_{Hf} && (2.23p)\\
M_{zr_0} &= D_r \cos(C_r \arctan(B_r\alpha_r)) && (2.23q)\\
\\
M_{z_0} &= M_{zr_0} + M'_{z_0} && (2.23r)
\end{aligned}
$$

Combined slip

In the case where longitudinal and lateral slips occur simultaneously, Equations (2.21), (2.22) and (2.23) do not respect the laws of friction anymore given by Equation (2.16). Therefore, less forces

are generated compared to the pure slip case: this is modeled by a ponderation factor G_{xa} (resp. G_{yk}) on the pure longitudinal (resp. lateral) force (see [Pacejka, 2002]) such as given by Equations (2.24), and (2.25). The longitudinal and lateral tire forces generated in the case of combined slip are displayed respectively in Figure 2.14 and 2.15. The self-aligning moment in case of combined slip is given by Equations (2.26).

- Longitudinal forces (combined slip)

$$
\begin{aligned}
B_{xa} &= r_{Bx_1}\cos(\arctan(r_{Bx_2}\tau_x)) && (2.24a)\\
C_{xa} &= r_{Cx_1} && (2.24b)\\
E_{xa} &= r_{Ex_1} + r_{Ex_2} dF_z && (2.24c)\\
S_{Hxa} &= r_{Hx_1} && (2.24d)\\
\alpha_{shift_x} &= \alpha + S_{Hxa} && (2.24e)\\
G_{xa} &= \frac{\cos(C_{xa}\arctan(B_{xa}\alpha_{shift_x} - E_{xa}(B_{xa}\alpha_{shift_x} - \arctan(B_{xa}\alpha_{shift_x}))))}{\cos(C_{xa}\arctan(B_{xa}S_{Hxa} - E_{xa}(B_{xa}S_{Hxa} - \arctan(B_{xa}S_{Hxa}))))} && (2.24f)\\
F_{xp} &= G_{xa}F_{xp_0} && (2.24g)
\end{aligned}
$$

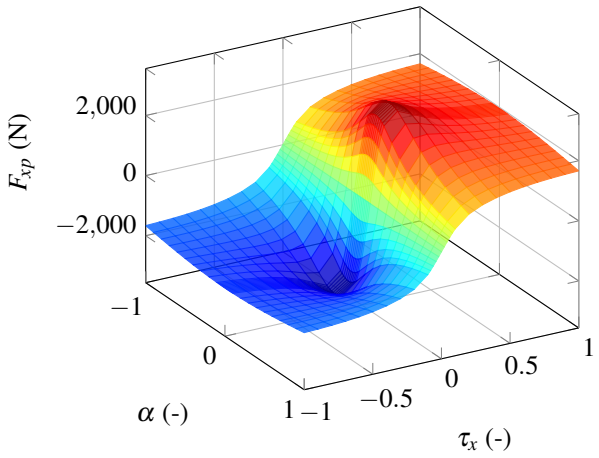


Figure 2.14 – Longitudinal tire forces in case of combined slip.

- Lateral forces (combined slip)

$$
\begin{aligned}
B_{yk} &= r_{By_1}\cos(\arctan(r_{By_2}(\alpha - r_{By_3}))) && (2.25a)\\
C_{yk} &= r_{Cy_1} && (2.25b)\\
E_{yk} &= r_{Ey_1} + r_{Ey_2} dF_z && (2.25c)\\
S_{Hyk} &= r_{Hy_1} + r_{Hy_2} dF_z && (2.25d)\\
D_{Vyk} &= \mu_y F_z(r_{Vy_1} + r_{Vy_2} dF_z + r_{Vy_3}\gamma)\cos(\arctan(r_{Vy_4}\alpha)) && (2.25e)\\
S_{Vyk} &= D_{Vyk}\sin(r_{Vy_5}\arctan(r_{Vy_6}\tau_x)) && (2.25f)\\
\tau_{shift_y} &= \tau_x + S_{Hyk} && (2.25g)\\
G_{yk} &= \frac{\cos(C_{yk}\arctan(B_{yk}\tau_{shift_y} - E_{yk}(B_{yk}\tau_{shift_y} - \arctan(B_{yk}\tau_{shift_y}))))}{\cos(C_{yk}\arctan(B_{yk}S_{Hyk} - E_{yk}(B_{yk}S_{Hyk} - \arctan(B_{yk}S_{Hyk}))))} && (2.25h)\\
F_{yp} &= G_{yk}F_{yp_0} + S_{Vyk} && (2.25i)
\end{aligned}
$$

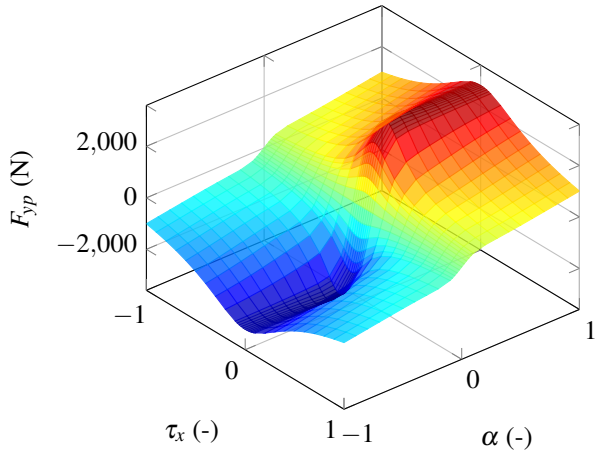


Figure 2.15 – Lateral tire forces in case of combined slip.

- Self-aligning torques (combined slip)

$$\alpha_{t_{eq}} = \sqrt{\alpha_t^2 + \left(\frac{K_{xk}}{K'_{ya}}\right)^2 \tau_x^2}\, sign(\alpha_t) \tag{2.26a}$$

$$\alpha_{r_{eq}} = \sqrt{\alpha_r^2 + \left(\frac{K_{xk}}{K'_{ya}}\right)^2 \tau_x^2}\, sign(\alpha_r) \tag{2.26b}$$

$$t = D_t \cos(C_t \arctan(B_t \alpha_{t_{eq}} - E_t(B_t \alpha_{t_{eq}} - \arctan(B_t \alpha_{t_{eq}})))) \cos(\alpha) \tag{2.26c}$$

$$F'_y = F_{yp} - S_{Vyk} \tag{2.26d}$$

$$M'_z = -tF'_y \tag{2.26e}$$

$$M_{zr} = D_r \cos(C_r \arctan(B_r \alpha_{r_{eq}})) \tag{2.26f}$$

$$s = r_0 \left(s_{sz_1} + s_{sz_2} \frac{F_{yp}}{F_{z_0}} + (s_{sz_3} + s_{sz_4} dF_z)\gamma \right) \tag{2.26g}$$

$$M_z = M'_z + M_{zr} + sF_{xp} \tag{2.26h}$$

Influence of F_z and μ

The simple version of “Pacejka’s Magic Formula" given by Equations (2.19) takes neither into account the normal reaction force F_z applied by the road on the vehicle, nor the friction coefficient μ of the road. However, in Equations (2.21d), (2.21e), (2.22e), (2.22d), (2.21g), (2.21h), (2.22g), (2.22h) and (2.22i) we observe that the coefficients D_x, D_y, as well as B_x and B_y are influenced by the value of μ and F_z. Thus, the longitudinal and lateral forces generated by the road on the tire depends strongly on these two values as illustrated in Figures 2.16 and 2.17.

2.2.2 The linear tire model

The simplified version of “Pacejka’s Magic Formula" for pure slip given by Equations (2.19) can be linearized if the longitudinal slip ratio or the lateral slip angle remain small as shown in Figure 2.18. In this case, the longitudinal and lateral forces are given by Equations (2.27).

$$F_{xp} = C_\tau \tau_x \tag{2.27a}$$

$$F_{yp} = C_\alpha \alpha \tag{2.27b}$$

where $C_\tau = B_x C_x D_x$ is the longitudinal stiffness and $C_\alpha = B_y C_y D_y$ the cornering stiffness.

This model is very convenient for control purpose as we will see in chapter 5.

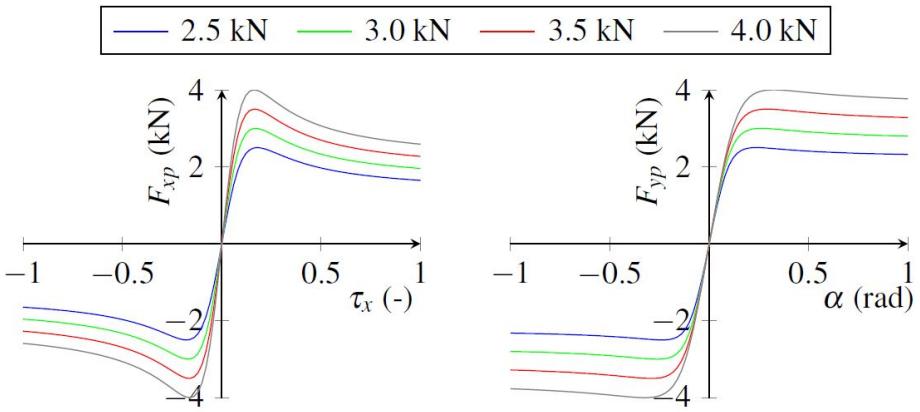


Figure 2.16 – Longitudinal (left) and lateral (right) tire forces generated by Pacejka's Magic Formula for different values of F_z, with $F_{z0} = 3kN$ and $\mu = 1$.

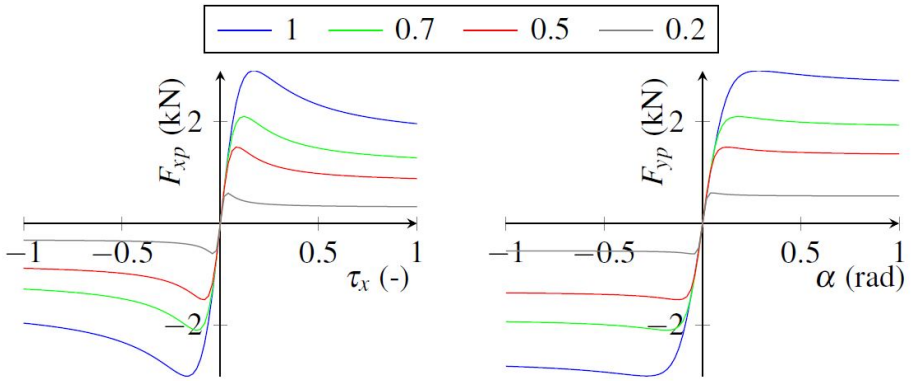


Figure 2.17 – Longitudinal (left) and lateral (right) tire forces generated by Pacejka's Magic Formula for different values of μ, with $F_z = F_{z0} = 3kN$.

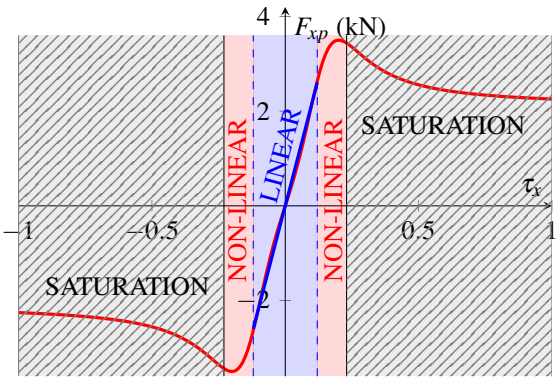


Figure 2.18 – Linearization of the Pacejka model.

2.2.3 Dugoff model

The Dugoff tire model was introduced in [Dugoff et al., 1969]. It is an analytical model assuming a uniform distribution of the tire forces on the tire/road contact zone. [Guntur and Sankar, 1980] have included the friction circle concept to the model to ensure that the laws of friction are respected. Unlike the Pacejka model, the Dugoff model deals easily with the road friction coefficient

and the normal reaction force as shown in Equations (2.28):

$$F_{xp} = C_\tau \frac{\tau_x}{1-\tau_x} f(\lambda) \tag{2.28a}$$

$$F_{yp} = C_\alpha \frac{\tan(\alpha)}{1-\tau_x} f(\lambda) \tag{2.28b}$$

where $\lambda = \frac{\mu F_z (1+\tau_x)}{2\sqrt{(C_\tau \tau_x)^2 + (C_\alpha \tan(\alpha))^2}}$ and:

$$f(\lambda) = \begin{cases} (2-\lambda)\lambda & \text{if } \lambda < 1 \\ 1 & \text{if } \lambda \geq 1 \end{cases} \tag{2.29}$$

2.2.4 Other models

The models presented in this section are less used in the literature. In particular, the LuGre and Burckhardt-Kiencke model present a slight difference with the models mentioned aforehead: they cannot be expressed by Equations (2.11), as they use some other states of the tire as input to their models.

LuGre model

The LuGre friction model, introduced in [Canudas de Wit et al., 1995] has been applied to tire-road friction in [Canudas de Wit and Tsiotras, 1999, Canudas-de Wit and Horowitz, 1999]. The model is given by Equations (2.30). It is named after the contraction of Lund-Grenoble, as the results were obtained from a collaboration between Lund Institute of Technology and Institut National Polytechnique de Grenoble.

$$\dot{\xi} = v_r - \frac{\sigma_0 |v_r|}{g(v_r)} \tag{2.30a}$$

$$F_{xp} = (\sigma_0 \xi + \sigma_1 \dot{\xi} + \sigma_2 v_r) F_z \tag{2.30b}$$

$$g(v_r) = \mu_c + (\mu_s - \mu_c) \exp^{-\sqrt{|v_r/v_s|}} \tag{2.30c}$$

where σ_0 is the normalized rubber longitudinal lumped stiffness, σ_1 the normalized rubber longitudinal lumped damping, σ_2 the normalized viscous relative damping, μ_c the normalized Coulomb friction coefficient, μ_s the normalized static friction coefficient, v_s the Stribeck relative velocity, $v_r = (r_{eff}\omega - V_{xp})$ the relative velocity, V_{xp} the longitudinal velocity of the wheel and ξ the internal friction state. Examples of values taken from [Canudas de Wit and Tsiotras, 1999] are shown in Table 2.7. Note that $\mu_c \leq \mu_s$.

Parameter	Value	Units
σ_0	40	[1/m]
σ_1	4.9487	[s/m]
σ_2	0.0018	[s/m]
μ_c	0.5	[-]
μ_s	0.9	[-]
v_s	12.5	[m/s]

Table 2.7 – Example of values of the different coefficients of the LuGre model [Canudas de Wit and Tsiotras, 1999].

Some improvements have been made in [Canudas-de Wit et al., 2003] to capture the transient behavior of the traction force observed during transitions between braking and acceleration.

Burckhardt-Kiencke model

The Burckhardt-Kiencke tire model computes the longitudinal and lateral tire forces from the friction coefficient or adhesion coefficient μ, the speed of the vehicle at center of gravity V and the normal reaction force F_z. The friction coefficient μ is defined as the ratio of the frictional force acting in the wheel plane F_{fric} and the wheel ground contact force F_z: $\mu = \frac{F_{fric}}{F_z}$ (see Figure 2.19).

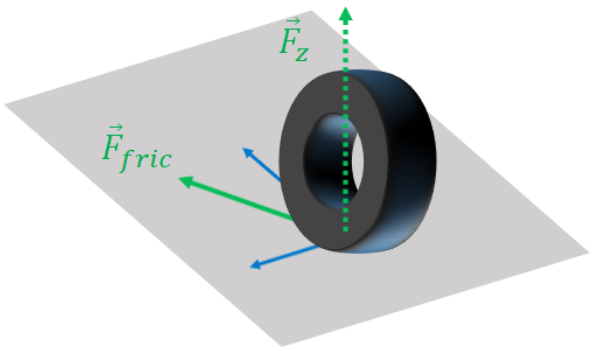


Figure 2.19 – Friction force.

In its simple version originally proposed by Burckhardt [Burckhardt, 1993], the friction coefficient depends only on the total slip ratio τ_{total} as given by Equation (2.31):

$$\mu(\tau_{total}) = c_1\left(1-\exp^{-c_2\tau_{total}}\right) - c_3\tau_{total} \tag{2.31}$$

The total slip ratio corresponds to the norm of the slip ratios:

$$\tau_{total} = \sqrt{\tau_x^2+\tau_y^2} \tag{2.32}$$

where τ_y is the lateral slip ratio.

The parameters c_1, c_2, c_3 are characteristic of the road surface, in particular the coating. Their values are given in Table 2.8.

	c_1	c_2	c_3
Asphalt, dry	1.2801	23.99	0.52
Asphalt, wet	0.857	33.822	0.347
Concrete, dry	1.1973	25.168	0.5373
Cobblestones, dry	1.3713	6.4565	0.6691
Cobblestones, wet	0.4004	33.7080	0.1204
Snow	0.1946	94.129	0.0646
Ice	0.05	306.39	0

Table 2.8 – Values of the different coefficients of the Burckhardt model.

Equation (2.31) can be extended to a more precise model taking into account the speed V at the center of gravity of the vehicle and the normal reaction forces F_z:

$$\mu(\tau_{total}) = c_1\left(1-\exp^{-c_2\tau_{total}}\right) - c_3\tau_{total}\exp^{-c_4\tau_{total}V}\left(1-c_5F_z^2\right) \tag{2.33}$$

The parameters c_4 et c_5 are both positive values and cannot exceed 1. They describe respectively the influence of a higher drive velocity and of a higher wheel load. More precisely, parameter c_4 lies in the range 0.002s/m to 0.004s/m. More details can be found in [Kiencke and Nielsen, 2005].

At last, the friction forces can be computed using Equations (2.34):

$$F_x(\tau_{total},\tau_x) = \frac{\tau_x}{\tau_{total}}\mu(\tau_{total})F_z \tag{2.34a}$$

$$F_y(\tau_{total},\alpha) = \frac{\alpha}{\tau_{total}}\mu(\tau_{total})F_z \tag{2.34b}$$

Fiala model

The Fiala model has been introduced in [Fiala, 1954]. In order to take into account the friction circle, [Hindiyeh and Gerdes, 2014] have proposed a slightly modified version. It is a model of the lateral tire force F_{yp} only, taking into account both the load transfer through F_z, the road conditions μ and the slip angle α as shown in Equation (2.35). Note that the longitudinal forces F_{xp} can not be obtained using the Fiala model. However, this model is very convenient for designing controllers in the case of varying F_z or μ (see for example [Goh and Gerdes, 2016]).

$$F_{yp} = \begin{cases} -C_\alpha z + \frac{C_\alpha^2}{3\xi\mu F_z}|z|z - \frac{C_\alpha^3}{27\xi^2\mu^2 F_z^2}z^3 & \text{if } |z| < \tan\alpha_{sl} \\ -\xi\mu F_z \operatorname{sign}\alpha & \text{if } |z| \geq \tan\alpha_{sl} \end{cases} \tag{2.35a}$$

$$z = \tan\alpha \tag{2.35b}$$

$$\alpha_{sl} = \arctan\frac{3\xi\mu F_z}{C_\alpha} \tag{2.35c}$$

$\xi \in [0;1]$ is a derating factor given by Equaton (2.36) that accounts for a reduction in lateral force capacity when longitudinal force is applied due to the friction circle:

$$\xi = \frac{\sqrt{(\mu F_z)^2 - F_{xp}^2}}{\mu F_z} \tag{2.36}$$

2.3 Vehicle simulator

Now that we have presented how to model the vehicle and tire dynamics, we are going to present the vehicle simulator that we have developed in order to test motion planning and control algorithms. For that purpose, we chose the 10 Degrees of Freedom (10 DoF) vehicle model, comprising Equations (2.5), (2.6) and (2.17) with a combined Pacejka tire model.

2.3.1 The simulator engine

At each time step, the inputs of our simulator are the actual state $X = \{x_1, ..., x_{16}\}$ of the vehicle and the wheels, and the control input $U = \{u_1, ..., u_5\}$ to be applied on the vehicle. The states $\{x_1, ..., x_{16}\}$ correspond respectively to $\{X, V_x, Y, V_y, Z, V_z, \theta, \dot{\theta}, \phi, \dot{\phi}, \psi, \dot{\psi}, \omega_1, \omega_2, \omega_3, \omega_4\}$ and the control inputs $\{u_1, ..., u_5\}$ to $\{T_{fl}, T_{fr}, T_{rl}, T_{rr}, \delta_f\}$. The simulator outputs the new state of the vehicle as shown in Figure 2.20, after having computed the forces applied to the system. The road friction coefficient is an external parameter that depends on the road conditions.

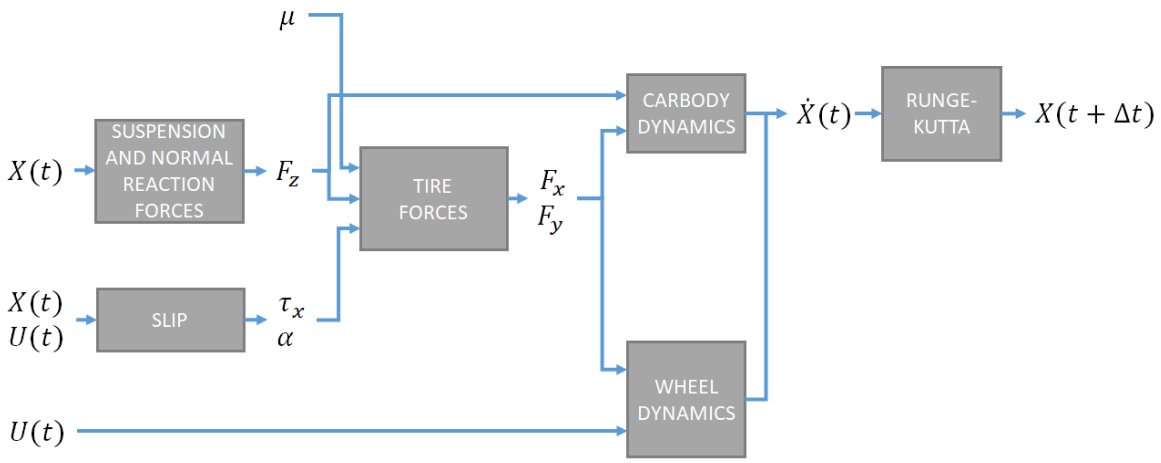


Figure 2.20 – Description of the simulator engine.

Computation of the forces

The first step of the simulation iteration consists in computing all the forces that are applied on the vehicle:

1. Compute the longitudinal slip τ_{xi} using Equation (2.12) and the slip angle α_i using Equations (2.13) for each wheel i;
   Note that V_{xpi} is the velocity of the wheel i in the pneumatic frame i:

$$V_{xpi} = \left[\begin{pmatrix}\cos\delta_i & -\sin\delta_i \\ \sin\delta_i & \cos\delta_i\end{pmatrix}\begin{pmatrix}V_x + \epsilon_i \dot{\psi} l_w \\ V_y + \dot{\psi} l_i\end{pmatrix}\right]\begin{pmatrix}1 \\ 0\end{pmatrix} \tag{2.37}$$

   where $\epsilon_i = \begin{cases} 1 & \text{if } i = \{2,4\} \\ -1 & \text{if } i = \{1,3\} \end{cases}$ and $l_i = \begin{cases} l_f & \text{if } i = \{1,2\} \\ -l_r & \text{if } i = \{3,4\} \end{cases}$.

2. Compute the spring displacement Δz_{si} for each wheel i based on the value of the roll θ and pitch ϕ angles as shown by Equation (2.7b);

3. Compute the variation of the suspension forces ΔF_{si} at each wheel as in Equation (2.7a);

4. Compute the normal reaction forces F_{zi} at each wheel as in Equation (2.7c);

5. Compute the longitudinal F_{xpi} and lateral F_{ypi} tire forces in the tire frame for each wheel using Equations (2.21), (2.22), (2.24) and (2.25). The coefficients used are reported in Table 2.9 and are taken from [Pacejka, 2002];

6. Compute the longitudinal F_{xi} and lateral F_{yi} tire forces in the vehicle frame for each wheel by using Equation (2.10);

7. Compute the aerodynamic drag forces F_{aero} using Equation (2.3).

$p_{Cx1} = 1.65$	$p_{Dx1} = 1$	$p_{Dx2} = 0$	$p_{Ex1} = -0.5$	$p_{Ex2} = 0$	$p_{Ex3} = 0$	$p_{Ex4} = 0$
$p_{Kx1} = 12$	$p_{Kx2} = 10$	$p_{Kx3} = -0.6$	$p_{Hx1} = 0$	$p_{Hx2} = 0$	$p_{Vx1} = 0$	$p_{Vx2} = 0$
$p_{Cy1} = 1.3$	$p_{Dy1} = 1$	$p_{Dy2} = 0$	$p_{Dy1} = 0$			
$p_{Ey1} = -1$	$p_{Ey2} = 0$	$p_{Ey3} = 0$	$p_{Ey4} = 0$			
$p_{Ky1} = 10$	$p_{Ky2} = 1.5$	$p_{Ky3} = 0$	$p_{Hy1} = 0$	$p_{Hy2} = 0$	$p_{Hy3} = 0.25$	
$p_{Vy1} = 0$	$p_{Vy2} = 0$	$p_{Vy3} = 0.15$	$p_{Vy4} = 0$			
$q_{Bz1} = 6$	$q_{Bz2} = -4$	$q_{Bz3} = 0.6$	$q_{Bz4} = 0$	$q_{Bz5} = 0$	$q_{Bz9} = 0$	$q_{Bz10} = 0.7$
$q_{Cz1} = 1.05$	$q_{Dz1} = 0.12$	$q_{Dz2} = -0.03$	$q_{Dz3} = 0$	$q_{Dz4} = -1$	$q_{Dz6} = 0$	$q_{Dz7} = 0$
$q_{Dz8} = 0.6$	$q_{Dz9} = 0.2$	$q_{Ez1} = -10$	$q_{Ez2} = 0$	$q_{Ez3} = 0$	$q_{Ez4} = 0$	$q_{Ez5} = 0$
$q_{Hz1} = 0$	$q_{Hz2} = 0$	$q_{Hz3} = 0$	$q_{Hz4} = 0$			
$r_{Bx1} = 5$	$r_{Bx2} = 8$	$r_{Cx1} = 1$	$r_{Hx1} = 0$			
$r_{By1} = 7$	$r_{By2} = 2.5$	$r_{By3} = 0$	$r_{Cy1} = 1$	$r_{Hy1} = 0.02$		
$r_{Vy1} = 0$	$r_{Vy2} = 0$	$r_{Vy3} = -0.2$	$r_{Vy4} = 14$	$r_{Vy5} = 1.9$	$r_{Vy6} = 10$	
$s_{sz1} = 0$	$s_{sz2} = -0.1$	$s_{sz3} = -1.0$	$s_{sz4} = 0$			

Table 2.9 – Coefficients used for the complete Pacejka model [Pacejka, 2002].

Computing the dynamics

The next step of the simulation iteration consists in computing the state vector X at time $t + \Delta t$, where Δt is the time step of the simulation. For that purpose, we first compute the derivative of each state as shown in Equation (2.38) which stems from Equations (2.1), (2.5), (2.6) and (2.17). These equations are then integrated using a Runge-Kutta solver [Dormand and Prince, 1980]. Note that a forward Euler solver does not work except if the discretization step Δt is really small (less than 1ms) as the wheel dynamic is much faster than the one of the carbody (cf. Remark 3).

$$\dot{x}_1 = x_2 \cos x_{11} - x_4 \sin x_{11} \tag{2.38a}$$
$$\dot{x}_2 = x_{12} x_4 - x_{10} x_6 + \frac{1}{\mathrm{M_T}} \left(\sum_{i=1}^{4} \mathrm{F}_{xi} - \mathrm{F}_{aero} \right) \tag{2.38b}$$
$$\dot{x}_3 = x_2 \sin x_{11} + x_4 \cos x_{11} \tag{2.38c}$$
$$\dot{x}_4 = -x_{12} x_2 + x_8 x_6 + \frac{1}{\mathrm{M_T}} \sum_{i=1}^{4} \mathrm{F}_{yi} \tag{2.38d}$$
$$\dot{x}_5 = x_6 \tag{2.38e}$$
$$\dot{x}_6 = \frac{1}{\mathrm{M_S}} \sum_{i=1}^{4} \mathrm{F}_{si} - g \cos x_7 cos x_9 \tag{2.38f}$$
$$\dot{x}_7 = x_8 \tag{2.38g}$$
$$\dot{x}_8 = \frac{1}{\mathrm{I}_x} \left(l_w \left(\mathrm{F}_{s1} + \mathrm{F}_{s3} - \mathrm{F}_{s2} - \mathrm{F}_{s4} \right) + x_5 \sum_{i=1}^{4} \mathrm{F}_{yi} \right) \tag{2.38h}$$
$$\dot{x}_9 = x_{10} \tag{2.38i}$$
$$\dot{x}_{10} = \frac{1}{\mathrm{I}_y} \left(-l_f \left(\mathrm{F}_{s1} + \mathrm{F}_{s2} \right) + l_r \left(\mathrm{F}_{s3} - \mathrm{F}_{s4} \right) + x_5 \sum_{i=1}^{4} \mathrm{F}_{xi} \right) \tag{2.38j}$$
$$\dot{x}_{11} = x_{12} \tag{2.38k}$$
$$\dot{x}_{12} = \frac{1}{\mathrm{I}_z} \left(l_f \left(\mathrm{F}_{y1} + \mathrm{F}_{y2} \right) - l_r \left(\mathrm{F}_{y3} - \mathrm{F}_{y4} \right) + l_w \left(\mathrm{F}_{x2} + \mathrm{F}_{x4} - \mathrm{F}_{x1} - \mathrm{F}_{x3} \right) \right) \tag{2.38l}$$
$$\dot{x}_{13} = \frac{u_1 - r_{eff} \mathrm{F}_{xp1}}{\mathrm{I}_r} \tag{2.38m}$$
$$\dot{x}_{14} = \frac{u_2 - r_{eff} \mathrm{F}_{xp2}}{\mathrm{I}_r} \tag{2.38n}$$
$$\dot{x}_{15} = \frac{u_3 - r_{eff} \mathrm{F}_{xp3}}{\mathrm{I}_r} \tag{2.38o}$$
$$\dot{x}_{16} = \frac{u_4 - r_{eff} \mathrm{F}_{xp4}}{\mathrm{I}_r} \tag{2.38p}$$

2.3.2 The graphical interface - coupling to PreScan

Finally, in order to generate traffic scenarios and visualize the output, we coupled our 10 DoF vehicle model to PreScan, a software developed by [TASS International,]. It is a physics-based simulation platform that aims to help the development of intelligent vehicles, by enabling to test for example Advanced Driver Assistance Systems (ADAS), vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. The software provides both a user-friendly Graphical User Interface (GUI) to develop quickly a traffic scenario (see Figure 2.21), a simulation engine and a visualization tool (see Figure 2.22).

The vehicle model by default used for the simulation engine does not take into account slip and skip. However, it is very easy to provide its own Matlab/Simulink model and thus get the level of modeling expected. Therefore, we designed and visualized our scenarios using the PreScan software, where the simulation engine corresponds to the vehicle model presented previously.

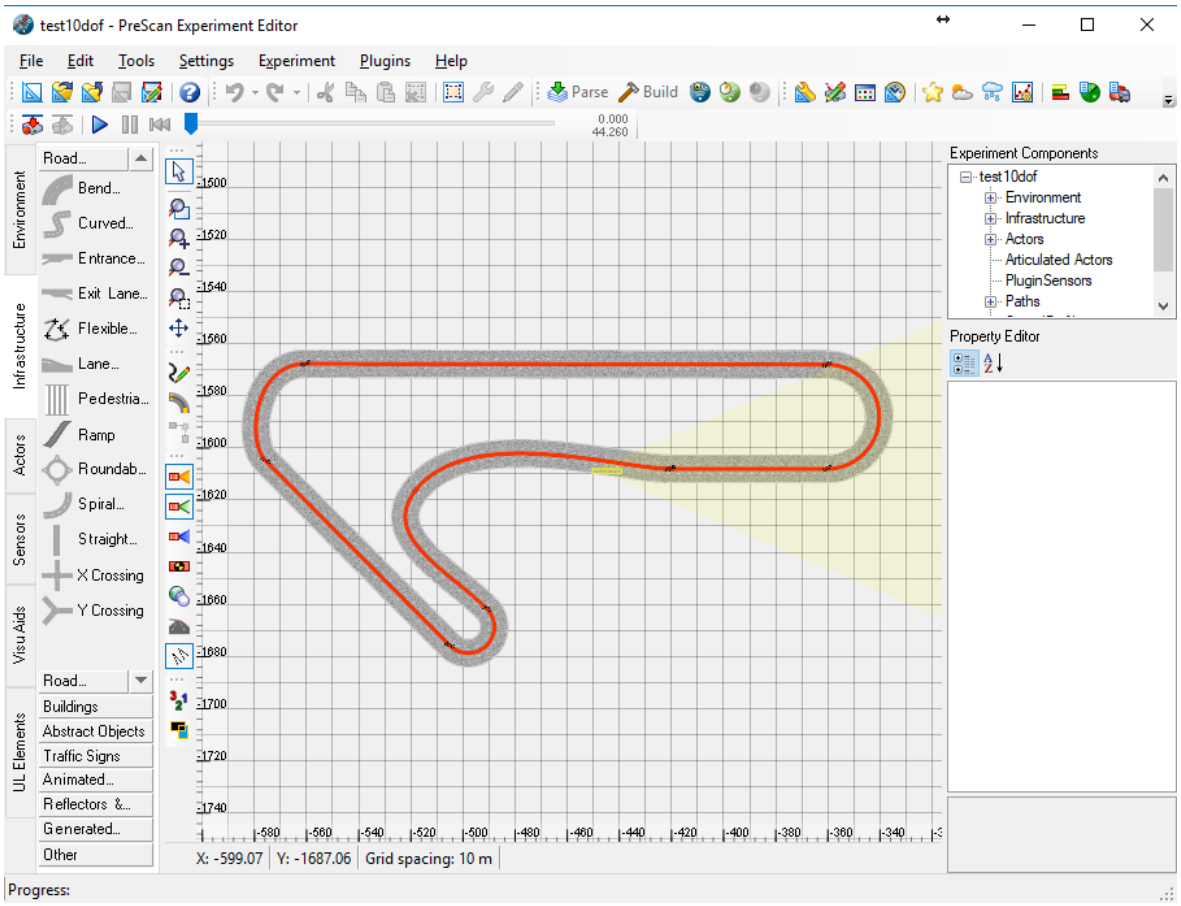


Figure 2.21 – Screenshot of PreScan's GUI.

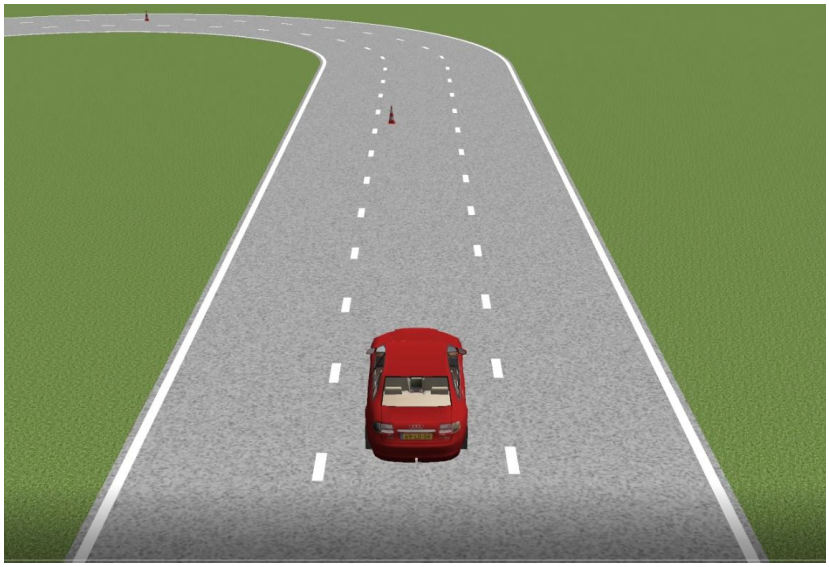


Figure 2.22 – Screenshot of PreScan's visualization tool.

2.4 Simplified models for motion planning and control

In this section, some simplified vehicle models that can be used for motion planning and control purpose are presented, namely the point-mass model, the unicycle model, the kinematic bicycle model and the dynamic bicycle model.

2.4.1 The point-mass model

The point-mass model is the simplest modeling of a vehicle: the latter is assimilated to a point-mass where the control inputs are for example for a second-order point-mass model the accelerations in the inertial frame a_X and a_Y. The system obtained is a linear one, given by Equation (2.39).

$$\dot{\xi}_{pm} = A\xi_{pm} + BU_{pm} \tag{2.39}$$

For a second-order point-mass model, $\xi_{pm} = (X, \dot{X}, Y, \dot{Y})^T$ is the state of the vehicle, $U_{pm} = (a_X, a_Y)^T$ is the control input and matrices A and B are given by Equation (2.40):

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad ; \quad B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{2.40}$$

The variables X and Y represent the positions of the vehicle in the inertial frame, $\dot{X}$ and $\dot{Y}$ the speed of the vehicle in the inertial frame.

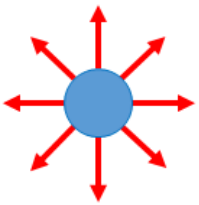


Figure 2.23 – The point-mass model.

The point-mass model is very efficient from a computational point of view but has a poor accuracy as the vehicle can move in any direction, as shown in Figure 2.23, even perpendicularly to its orientation. Some constraints can be added in order to improve its accuracy: for example, in [Qian et al., 2016], the authors added the constraint given by Equation (2.41) in order to avoid pure lateral motion of the model.

$$|\dot{Y}| \leq k|\dot{X}| \quad \text{with} \quad 0 < k = \tan(\psi_{max}) < 1 \tag{2.41}$$

Remark 4. *Both the first-order [Nilsson et al., 2015b], second-order [Murgovski and Sjöberg, 2015] and third-order [Qian et al., 2016] point-mass models exist in the literature. The control inputs are respectively the velocities, the accelerations and the jerks.*

2.4.2 The unicycle model

The unicycle model corresponds to a single-wheel kinematic model given by Equations (2.42) and illustrated by Figure 2.24. The control inputs are the velocity V and the angular velocity u_ψ. For more smoothness, the linear acceleration of the wheel $u_{\dot{V}}$ can be used instead of V by adding Equation (2.42d). Due to its simplicity, the unicycle model is used mostly at the motion planning level and is also referred to as the “simple kinematic car model". The angular speed u_ψ is limited

by the maximal curvature the vehicle can follow.

$$\dot{X} = V\cos(\psi) \tag{2.42a}$$
$$\dot{Y} = V\sin(\psi) \tag{2.42b}$$
$$\dot{\psi} = u_{\dot{\psi}} \tag{2.42c}$$
$$\dot{V} = u_{\dot{V}} \tag{2.42d}$$

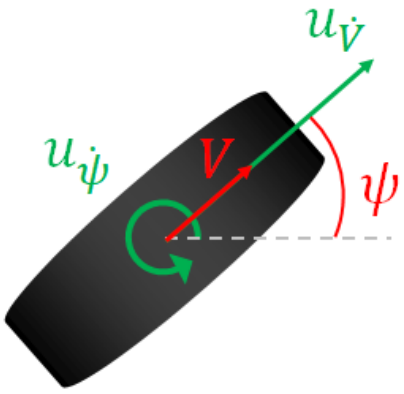


Figure 2.24 – Unicycle model of the vehicle.

The unicycle model satisfies the non-holonomic constraint on the vehicle motion which will be explained in chapter 3.3. However, the model can spin around without moving forwards or backwards.

2.4.3 The kinematic bicycle model

The word *kinematic* (from the Greek κινημα, "motion") in classical mechanics refers to the motion of systems such as points or bodies, without considering the masses or forces involved in it. Hence, the kinematic models describes the geometry of motion of the system.

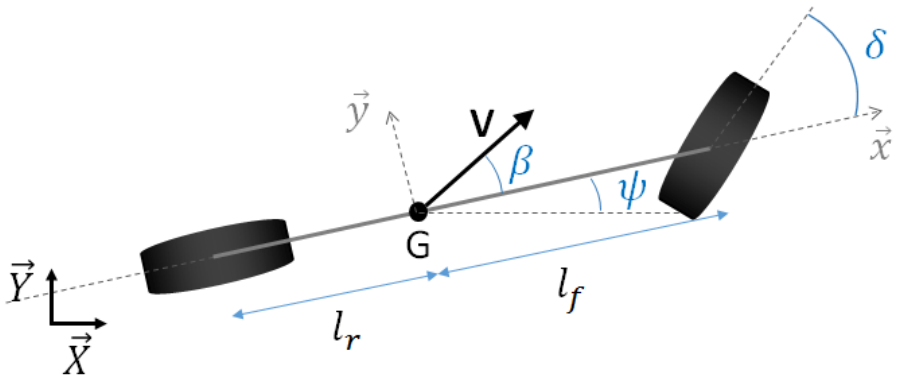


Figure 2.25 – Kinematic bicycle model of the vehicle.

In the kinematic bicycle model, the two front wheels (resp. the two rear wheels) of the vehicle are lumped into a unique wheel located at the center of the front axle (resp. of the rear axle) such as illustrated on Figure 2.25. The control inputs are the speed at center of gravity V and the front steering angle δ_f when assuming that only the front wheel can be steered. The model, expressed at the center of gravity of the vehicle, is given by Equation (2.43a) to (2.43c). The demonstration

can be found in [Rajamani, 2012].

$$\dot{X} = V\cos\left(\psi+\beta(\delta_f)\right) \tag{2.43a}$$
$$\dot{Y} = V\sin\left(\psi+\beta(\delta_f)\right) \tag{2.43b}$$
$$\dot{\psi} = \frac{V}{l_r}\sin\left(\beta(\delta_f)\right) \tag{2.43c}$$

where X and Y are the coordinates of the center of gravity of the vehicle in the inertial frame, ψ is the yaw angle and l_f and l_r are two geometric parameters of the vehicle, respectively the distance from the center of gravity to the front and rear axle. The slip angle β at the center of gravity is a function of δ_f defined by Equation (2.43d).

$$\beta(\delta_f) = \operatorname{atan}\left(\tan(\delta_f)\frac{l_r}{l_f+l_r}\right) \tag{2.43d}$$

In the motion planning literature, it is common to use as control inputs the total acceleration u_1 of the vehicle and the steering rate u_2 of the front wheel. In this case, Equations (2.43e) and (2.43f) are added to the model. By abuse of language, this model is also referred to as the "kinematic bicycle model":

$$\dot{V} = u_1 \tag{2.43e}$$
$$\dot{\delta}_f = u_2 \tag{2.43f}$$

In order to obtain Equations (2.43), the following assumptions were made:

- Only the front wheel is steerable ($\delta_r = 0$)
- The slip angle at each wheel is assumed to be zero. Therefore, the model is only valid at low speed.

The kinematic bicycle model takes into account the non-holonomic constraint on the vehicle motion which will be defined in section 3.3. However, the slip and skid of the vehicle are neglected.

Remark 5. *The kinematic bicycle model expressed at the center of the rear axle is equivalent to the unicycle model presented in section 2.4.2. However, the steering angle can no longer be used as an input of the model in that case.*

2.4.4 The dynamic bicycle model

The dynamic bicycle model, see Figure 2.26, has already been presented in section 2.1.1. As in the case of the kinematic bicycle model, the two front wheels (resp. the two rear wheels) of the vehicle are lumped into a unique wheel located at the center of the front axle (resp. of the rear axle). However, this model is based on the forces and inertias of the vehicle and dynamic phenomena are therefore considered.

There exists two versions of the dynamic bicycle model in the literature depending on the control variables chosen:

V1: Considering F_{xf}, F_{xr}, F_{yf} and F_{yr} as control inputs leads to Equation (2.44). In this case, the forces are limited by a friction circle (see section 2.2).

$$M_T(\dot{V}_x - \dot{\psi}V_y) = F_{xf}+F_{xr} \tag{2.44a}$$
$$M_T(\dot{V}_y + \dot{\psi}V_x) = F_{yf}+F_{yr} \tag{2.44b}$$
$$I_z\ddot{\psi} = l_fF_{yf} - l_rF_{yr} \tag{2.44c}$$

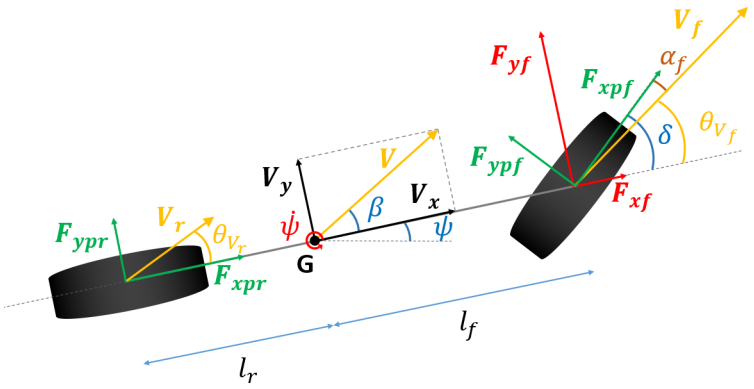


Figure 2.26 – The dynamic bicycle model.

V2: Considering a_x and δ_f as control inputs leads to Equation (2.45). The lateral forces F_{yf} and F_{yr} are computed from the steering angle δ_f using the linear tire model for example (see section 2.2.2). This method increases the computational time but leads to more intuitive control inputs.

$$\dot{V}_x - \dot{\psi} V_y = a_x \tag{2.45a}$$
$$M_T(\dot{V}_y + \dot{\psi} V_x) = F_{yf} + F_{yr} \tag{2.45b}$$
$$I_z \ddot{\psi} = l_f F_{yf} - l_r F_{yr} \tag{2.45c}$$

In both cases, the position of the center of gravity of the vehicle in the inertial frame follows Equations (2.46):

$$\dot{X} = V_x \cos\psi - V_y \sin\psi \tag{2.46a}$$
$$\dot{Y} = V_x \sin\psi + V_y \cos\psi \tag{2.46b}$$

While the first version is relatively simple from a computational point of view, the feasible limits of the control inputs F_{xf}, F_{xr}, F_{yf} and F_{yr} are difficult to obtain. Moreover, if $F_{yf} >> F_{xf}$, the non-holonomic constraint might not be respected. In the second version, it is important to have a good estimation of the cornering stiffness of the tire. Moreover, the linear tire model is not defined at very low speeds.

Part II

Motion planning for autonomous vehicles

Chapter 3

Motion planning for autonomous vehicles

“ Yes, there are two paths you can go by, but in the long run
There’s still time to change the road you’re on. ”

Led Zeppelin

Contents

The motion planning problem has been studied in robotics since the 1980s with the introduction of the *piano-mover problem* [Schwartz and Sharir, 1981]: given an initial and final configurations of a piano (or a robot) and a list of obstacles $o \in O$ in the physical space, can we find a continuous motion between these two configurations that avoids the obstacles. The motion can either be a *path*, i.e. a list of positions $(X_i^r, Y_i^r)_{i\in[\![1;n]\!]}$, or a *trajectory*, i.e. a list of positions indexed by time $(X_i^r(t), Y_i^r(t))_{i\in[\![1;n]\!]}$. The first case is referred to as a *path planning problem* while the second one corresponds to a *trajectory planning problem*. The main difference is that in the second case, a reference velocity for the robot is defined. The piano-mover problem is particularly evoking as the size of a piano, a three-dimensional object, is usually close to the one of the door, window or staircase it has to pass through. Not only the piano-mover needs to know if it is possible to move the piano to its new configuration, but also how it can be done.

The motion planning problem takes place in the 3D physical space (resp. 2D if only planar motions are considered): however, as the robot or piano is a three-dimensional (resp. two-dimensional) object with up to 6 degrees of freedom (resp. 3 DoF), determining if a solution exists is difficult even for a human. Moreover, one of the main difficulties is to translate a continuous problem into a combinatorial one that can be solved by a computer.

The methods for solving the motion planning problem can be classified into two main categories. The first one solves the problem in the configuration space, denoted by CS, which will be defined in section 3.1; these methods were proposed in the 1990s and beginning of the 2000s. The second category solves the problem in the control space and consists usually in solving an optimal control problem (OCP) which will be presented in section 3.2. These methods have been applied for motion planning problems more recently as they require a higher computational power.

Section 3.3 will introduce the notion of *non-honolomic constraint* which limits the admissible motions of the vehicle. Its impact on the motion planning problem will be analyzed. Finally, section 3.4 highlights the impacts of dynamic effects on the motion planning layer and outlines some solutions to face such issues.

More details can be found on the motion planning problem in [Lavalle, 2006, Latombe, 2013] or in the course given by Jean-Paul Laumond at College de France[1]. For a review of motion planning techniques applied specifically to autonomous vehicles, we refer the reader to one of the following articles: [Katrakazas et al., 2015, Paden et al., 2016, Gonzalez et al., 2016].

3.1 Methods based on the configuration space

Solving the motion planning problem in the physical space is a complicated task due to the dimensions of the robot. Therefore, [Lozano-Perez, 1983] proposed to change the problem formulation by introducing the configuration space denoted by CS, *i.e.* the space of all the possible configurations q of the system[2]:

Proposition 1. *Any admissible motion for a 3D mechanical system appears as a collision-free path for a point in the configuration space [Lozano-Perez, 1983].*

In CS, the configuration of the robot is characterized by a simple point and the configurations where the robot collides with an obstacle are denoted CS_{occ}. Thus, the motion planning problem boils down to capturing the topology (in particular the connected components) of the free configuration space, denoted CS_{free}, which is the set of all the configurations that do not collide with an obstacle (see Figure 3.1). If there exists a path joining the initial and final configurations, they have to be in the same connected component. Otherwise, no solution exists. Hence, the motion planning problem becomes easier to solve in the CS. CS_{free} is determined from the collision configuration space CS_{occ} as following: $CS_{free} = \{q \in CS, q \notin \overline{CS}_{occ}\}$. CS_{occ} can be determined

[1] Available at https://www.college-de-france.fr/site/jean-paul-laumond/course-2011-2012.htm, visited June 2018.

[2] A configuration defines in a unique way the physical space occupied by the robot. For a vehicle for example, the configuration $q = (X, Y, \psi)$ characterizes the robot's position and orientation in the physical space.

by computing for example for each obstacle o the Minkowski sum $R + B_o$, where R is the physical space occupied by robot and B_o the one occupied by obstacle o.

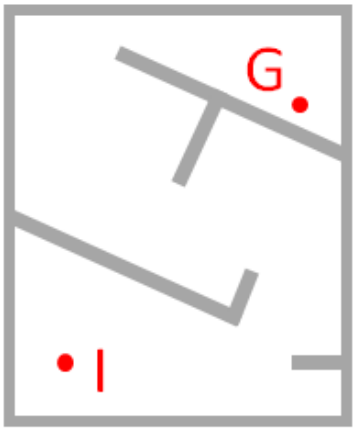


Figure 3.1 – Configuration space CS, free configuration space CS_{free} (white), occupied configuration space CS_{occ} (grey): the initial configuration is I and the final one is G.

However, the converse of Proposition 1 is not true: finding a path in the CS between two configurations does not guarantee that a feasible solution exists in the physical space (it is only a necessary condition) due to some restrictions on the motion of the robot. In particular, this is the case for a vehicle due to the non-holonomic constraints which will be discussed in section 3.3. However, if the robot is a *free-flying* object, *i.e.* that there is no constraints on its motion, Proposition 1 becomes also a sufficient condition.

Once the motion planning problem has been reformulated in the configuration space, it is still necessary to translate the continuous problem of exploring connected component into a combinatorial formulation that can be solved numerically by a computer. [Schwartz and Sharir, 1983] proved that the "piano-mover" problem was decidable (*i.e.* the computer can determine if there is a solution or not) using methods introduced by [Tarski, 1951], and [Reif, 1979] showed the problem is PSPACE-hard. [Schwartz and Sharir, 1983] came up with a method that has double-exponential growth in time with the dimension of the CS, which was later simplified by [Canny, 1988] to a solution that has single-exponential growth in time. Thus, the complexity of analytical solutions makes them unsuitable for real-time applications. Therefore, numerical approximation methods are used: this implies that CS_{free} is an open set (which is the case as the set of obstacles is a union of closed sets) in order to take into account numerical approximations.

3.1.1 Potential field method

[Khatib, 1986] introduced an artificial potential field method derived from classical physics to find a path between an initial and final configurations. The robot is moving in a force field where the configuration to be reached is an attractive pole and the obstacles are repulsive poles (see Figure 3.2). Let $q \to U_g(q)$ be the attractive potential associated to the goal and $q \to U_o(q)$ be the repulsive potential associated to the obstacle o. The total potential field U_{tot} is then given by Equation (3.1):

$$U_{tot} = U_g + \sum_{o=1}^{n_{obs}} U_o \tag{3.1}$$

The total force applied to the robot in configuration q is then $F_{tot}(q) = -\nabla_q(U_{tot})(q)$. The robot moves according to the laws of physics $\ddot{q} = F_{tot}$. It is also possible to apply directly gradient descent on the potential field U_{tot}.

This method provides smooth trajectories but can be trapped by local minima. Therefore there is no completeness for this method, meaning that there is no guarantee to find a solution even if

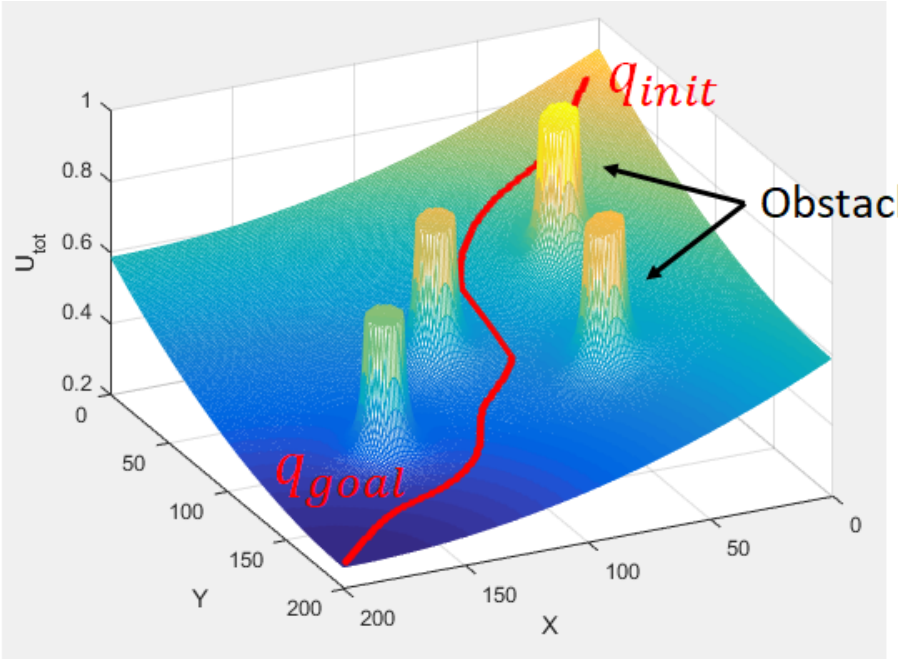


Figure 3.2 – Potential field methods.

one exists. [Barraquand et al., 1996] proposed to add random walks to the potential field method in order to avoid getting stuck in local minima. However, the completeness of this method can still not be guaranteed.

3.1.2 Deterministic sampling-based methods

Deterministic sampling-based methods consist in decomposing the configuration space into cells. They comprise two parts: first a cell-decomposition of the configuration space is performed; then graph-search algorithms such as A* [Peter E. Hart et al., 1968] or Dijkstra [Dijkstra, 1959] are used in order to obtain the shortest path (or the one with minimum cost) between the initial and final configurations. A wide range of methods exists for cell decomposition, each one having different performances in terms of computational complexity, accuracy and completeness. Deterministic sampling-based methods are well suited for low dimensional configuration space (typically 2 sometimes 3) but do not scale well for higher dimensions.

Grid-cell decomposition

The configuration space is decomposed into regular grid-cells as shown in Figure 3.3: these cells are either marked as occupied (in black), meaning that part of an obstacle is covering the cell, or as free (in white). The number of cells is $O(n^d)$ where d is the dimension of the configuration space (assuming that all the dimensions are decomposed into n cells).

The grid-cell search is complete: if a path exists among the free cells, it will be found. However, due to the cell resolution, the original problem might have a safe path while the grid-cell problem does not. Therefore, it is important to chose the cells relatively small for scenarios with a narrow driving corridor. However, this increases the computational burden as it increases the number of cells. In order to improve the computational efficiency while having a good resolution, the cells can be chosen smaller around obstacles, by using for example a quadtree.

Trapezoidal decomposition

This method consists in decomposing the free configuration space into trapezoidal and triangular cells as shown in Figure 3.4. This decomposition is done using a sweepline algorithm in $O(n \log n)$

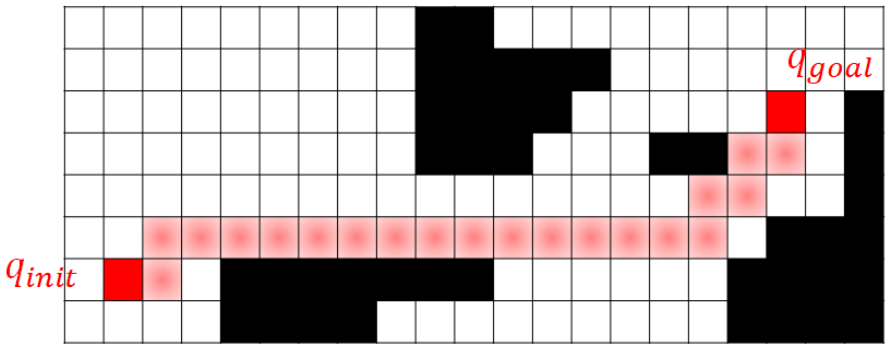


Figure 3.3 – Grid-cell decomposition of the configuration space.

where n is the number of vertices of the obstacles. Then, a connectivity graph is derived, representing the trapezoidal cells that are adjacent to one another, as shown in Figure 3.5.

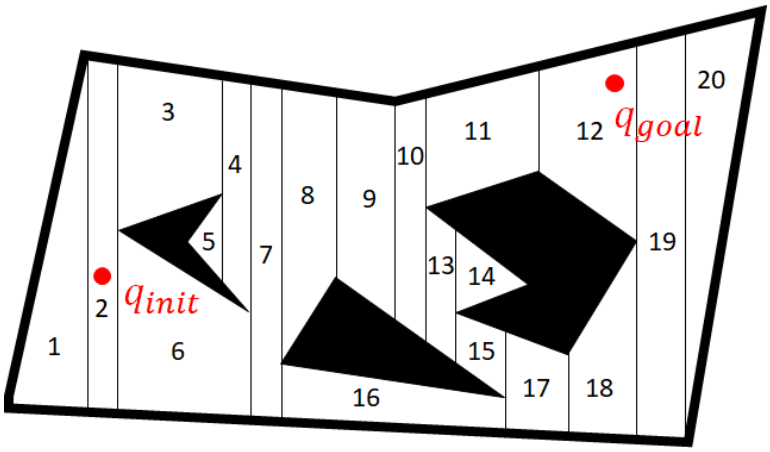


Figure 3.4 – Trapezoidal decomposition.

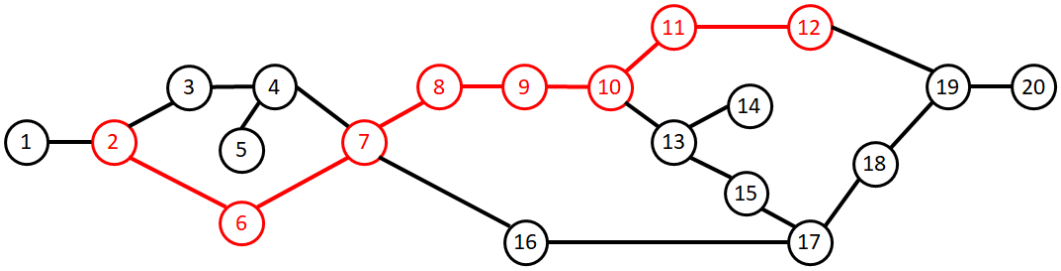


Figure 3.5 – Connectivity graph.

The performance of this algorithm is better than the grid-cell decomposition but the path obtained is not optimal.

Delaunay triangulation and Voronoï diagram

Given a set of points P, the Delaunay triangulation is such that the circumscribed circle of each triangle does not contain any other points (see Figure 3.6). This triangulation exists and is uniquely defined.

For the motion planning problem, we consider the set P of all the vertices of the obstacles[3]. A sweepline algorithm in $O(n \log n)$ is used to construct the triangulation where n is the number of vertices of the set of obstacles. A connectivity graph representing the adjacency between two

[3] the obstacles are supposed to have a polygonal shape

triangles is then obtained. The complexity of Delaunay triangulation is similar to the trapezoidal methods and the obtained path is also not optimal.

Note that the dual of the Delaunay triangulation is the Voronoï diagram, which is the set of points equidistant to the obstacles. Thus to find a path, it is possible to join the initial and final configurations to the closest point of the Voronoï diagram and then follow the diagram to reach the other configuration.

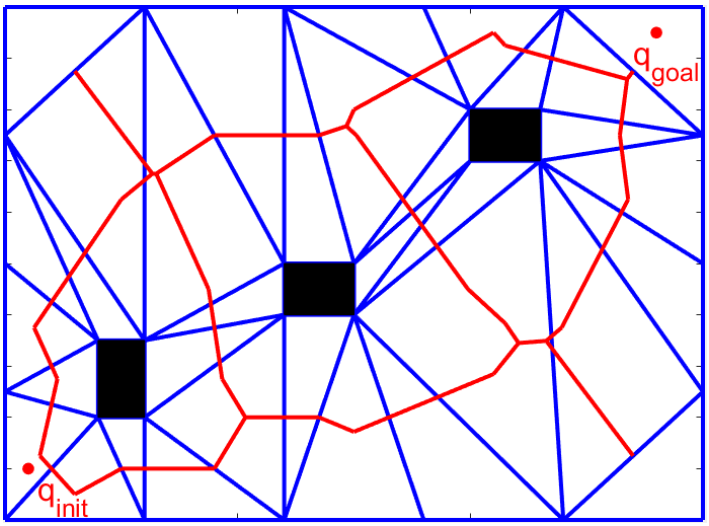


Figure 3.6 – Delaunay triangulation (blue), Voronoï diagram (red) and obstacles (black).

Visibility graph

Given a set of points P representing the vertices of the obstacles, we can construct a visibility graph (see Figure 3.7): two points are connected if they can be linked by a collision-free segment (or on the border of an obstacle). For a two-dimensional configuration space, the time complexity of the construction of the visibility graph is $O(n^2 \log n)$ using a binary search tree algorithm where n is the number of points in P. At the end, the initial and final configurations are also linked to the graph. If dim(CS) = 2, the visibility graph contains the shortest path. However, this is no longer the case for dim(CS) = 3.

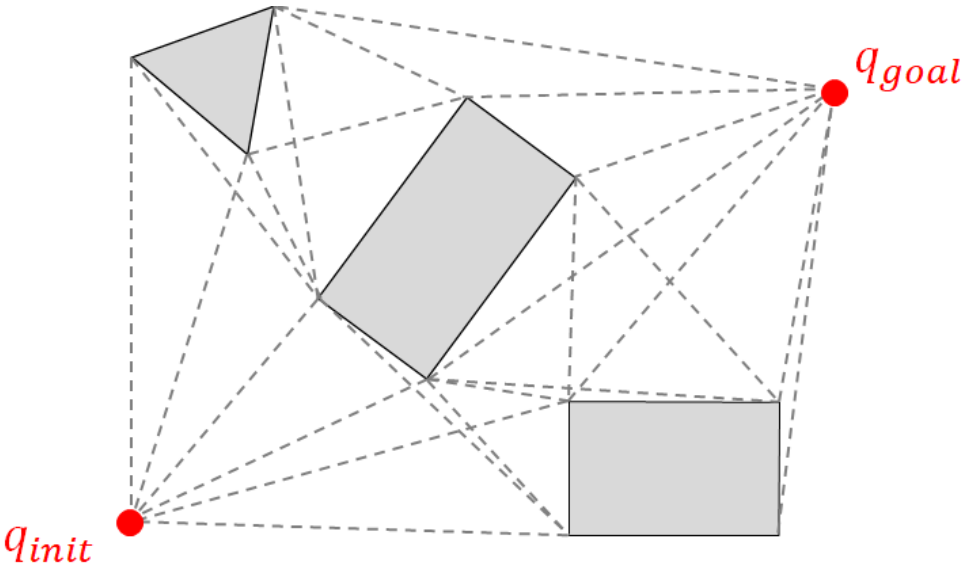


Figure 3.7 – Visibility graph (dotted lines) and obstacles (grey).

3.1.3 Probabilistic sampling-based methods

As the dimension of the configuration space or the number of obstacles increase, the computational time of deterministic sampling-based methods becomes prohibitive for real-time applications. Therefore, probabilistic sampling-based methods have been proposed in the literature: they consist in capturing the topology of the free configuration space CS_{free}, in particular the connected components, with a graph[4] by drawing randomly new points in the configuration space. Only the points in CS_{free} are kept. However, the improvement of the computational efficiency is achieved at the cost of relaxing the completeness property of the deterministic methods to a probabilistic completeness: the probability that the algorithm finds a safe path if one exists is converging to one as the computational time increases. Therefore, it is no longer possible to know if a safe path exists or not: only the mathematical expectation is considered. These methods work well for reasonable problems. Once a graph has been build, the *query phase* consists in obtaining the optimal path using graph-search algorithms such as A* or Dijkstra. Probabilistic sampling-based methods rely on boolean collision-test operators [Reggiani et al., 1990, Jiménez et al., 1998] in order to check whether two configurations can be connected without colliding with an obstacle.

Two main techniques exist in the literature and will be developed in this section: the Probabilistic Roadmap Method (PRM) and the Rapidly-exploring Random Tree (RRT). The only difference between them is the way the connectivity graph is constructed.

Probabilistic Roadmap Method (PRM)

The PRM has been introduced in [Kavraki et al., 1996]: a graph is grown by connecting new random configurations belonging to the free configuration space to it, until the graph is dense enough (see Figure 3.8). At the end, the initial and final configurations are also connected to the graph. In order to check the possibility to connect a new configuration to an existing one in the graph, the collision-test operator is used. PRM are not able to handle differential constraints such as nonholonomic constraints (see section 3.3).

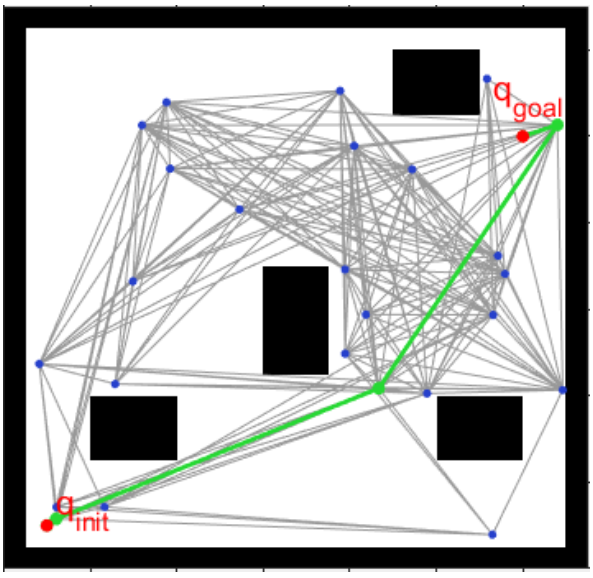


Figure 3.8 – Probabilistic Roadmap Method: the green path joins the initial and final configurations. The expended graph is shown in grey.

Rapidly-exploring Random Trees (RRT)

Random sampling has been introduced in [LaValle and Kuffner, 2001]. At the *exploring phase*, a tree rooted at the starting configuration is grown (see Figure 3.9): a random sample is drawn in the

[4] *i.e.* a binary relation between points

configuration space and kept only if it belongs to CS_{free}; then the collision-test operator checks whether this new configuration can be connected to the existing tree while avoiding obstacles and respecting the differential constraints (unlike PRM, RRT can handle these constraints). A connectivity graph is thus obtained. Once the final configuration can be connected to the graph, the *query phase* finds the shortest path between the initial and final configuration.

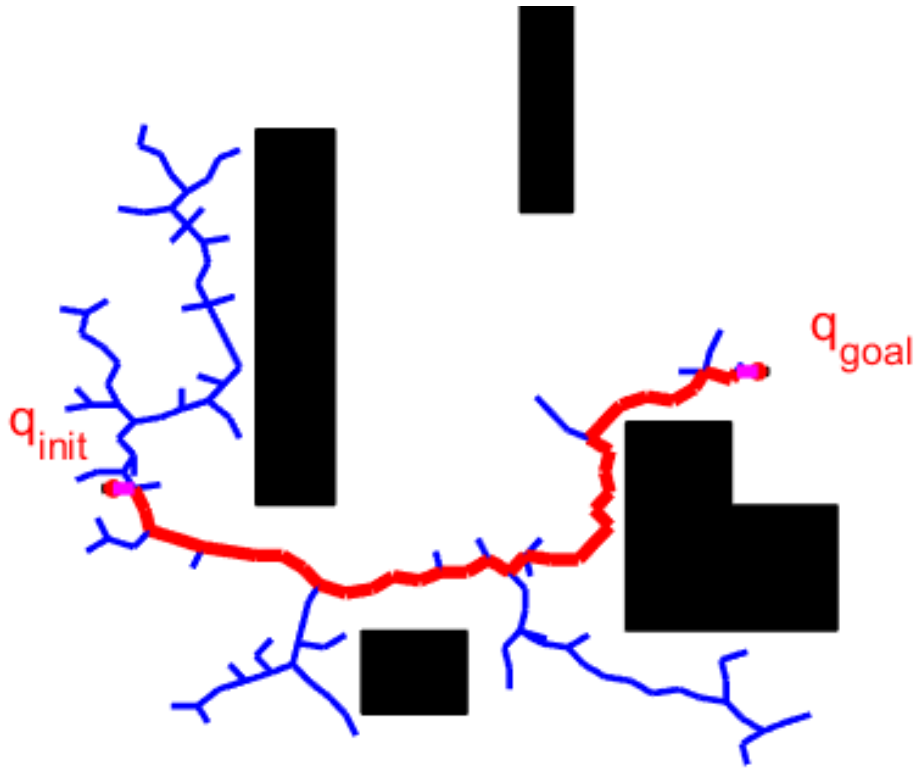


Figure 3.9 – Rapidly-exploring Random Tree: the red path joins the initial and final configuration. The expended rooted tree is shown in blue.

3.2 Optimal control problems

Applying optimization techniques for solving the motion planning problem have become quite popular in the last decade. This is due to the beginning of the "second-half of the chessboard"[5] of Moore's law [Brynjolfsson and McAfee, 2014], who predicted in 1965 that chip performances would double every 18 months. This prediction has turned out to be approximately true for more than 40 years. Therefore, the computational power currently available is such that optimal control problems can be solved in real-time.

Optimal control problems, compared to methods based on the configuration space, are able to easily handle constraints, in particular the non-holonomic and dynamic constraints (which will be discussed respectively in section 3.3 and 3.4) while relying on semi-analytical resolution which can make this method extremely fast to compute [Houska et al., 2011b]. Moreover, some aspects of the motion can be penalized such as the duration, the jerk or even the energy consumption: this enables to choose amoung the infinite number of paths/trajectories solution of the problem the most suited one. Finally, optimal control problems are better suited for coupling the longitudinal and lateral behavior of the vehicle, resulting in a reference trajectory instead of only a reference path. This is extremely important in the case of an environment with a lot of dynamic obstacles

[5]The chessboard is a good illustration of the power of exponential growth: if one places one rice grain on the first square of the chessboard, 2 on the second, 4 on the third, 8 on the fourth and so on, there will be $2^{32} \approx 4$ billions rice grains on the 33^{th} square, which is the first one of the second-half of the chessboard.

such as cars and pedestrians[6].

An optimal control problem formulation is composed of two parts: a cost function to be optimized and constraints on the system to be respected. Typically, the constraints comprises: (i) the initial state of the vehicle, (ii) the vehicle dynamics and kinematics, (iii) obstacle avoidance, (iv) the limits of the actuators, (v) the limits of the states (*e.g.* speed limit, road boundaries). However, defining a cost function is complex: minimizing the execution time for example leads to the saturations of the actuators which is nor dynamically feasible, nor comfortable. Therefore, a regularization term is usually added to penalize the control inputs and their variations. For better computational efficiency, constraints can also be relaxed using slack variables that are penalized in the cost function: in this case, a strong penalization of the slack variables should be considered.

3.2.1 Model Predictive Control (MPC)

Among optimal control techniques, Model Predictive Control (MPC) is one of the most popular one [Camacho and Bordons, 1999]. It computes in a receding horizon fashion an optimal control sequence that minimizes the cost function while respecting the constraints. Therefore, the problem is solved on a finite time-horizon T_y which enables to anticipate the behavior of the system.

We limit our scope to state-based MPC. In this case, a state-space *model* of the system is available and used to *predict* the outputs of the system given a sequence of *control* inputs as shown in Figure 3.10. An optimal control problem solver such as ACADO toolkit [Houska et al., 2011a] is then used to solve the problem and compute the optimal control sequence.

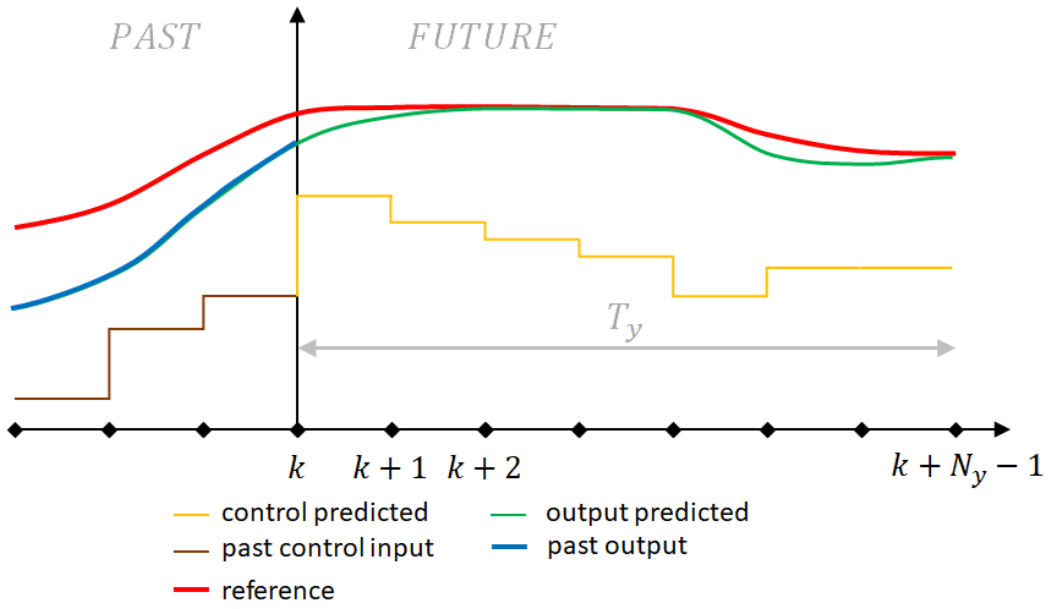


Figure 3.10 – Working principle of Model Predictive Control.

A typical MPC problem formulation looks like Equations (3.2). $\xi_t^k = \xi_t^{t+k\Delta t_y}$ denotes the state of the vehicle at time $t+k\Delta t_y$ predicted at time t ($k \in [\![0; N_y - 1]\!]$); $u_t^j = u_t^{t+j\Delta t_u}$ the control input to be applied at time j predicted at time t ($j \in [\![0; N_u - 1]\!]$) by the MPC. The vector $\bar{\xi}_t = [\xi_t^0, \xi_t^1, .., \xi_t^{N_y-1}]$ is the vector of all the state predictions between t and $t+T_y$ made at time t by the MPC; similarly, $\bar{U}_t = [u_t^0, u_t^1, .., u_t^{N_u-1}]$ is the vector of all the control input predictions between t and $t+T_u$ computed at time t. The cost function J depends both on the state and the control vectors: it is a quadratic function given by Equation (3.3), where Q and R are two definite positive matrices respectively of dimension $N_y$card(ξ) and $N_u$card(U). card(ξ) and card(U) denote respectively the number of states and control inputs of the state-space model. Usually, the ponderation of the cost

[6] A given portion of a path might be safe at a time t but not at time $t+\Delta t$.

function is the same throughout the prediction horizon leading to Q and R of the form given by Equation (3.4) where q and r are two diagonal square matrices of size respectively card(ξ) and card(U). The MPC solver computes the optimal control sequence $\bar{U}_t^*$ and its associated optimal state sequence $\bar{\xi}_t^*$. The main notations of the MPC formulation are summarized in Table 3.1.

$$J(\bar{\xi}_t^*, \bar{U}_t^*) = \min_{\bar{U}_t} J(\bar{\xi}_t, \bar{U}_t) \tag{3.2a}$$

subject to:

$$\xi_t^{k+1} = f(\xi_t^k, u_t^j) \qquad k = 0, 1, ..., N_y - 2; \tag{3.2b}$$
$$j \in \left[\max_l \left(l \leq k \frac{\Delta t_y}{\Delta t_u} \right); \max_l \left(l < (k+1) \frac{\Delta t_y}{\Delta t_u} \right) \right[$$

$$u_t^j \in [u_{lb}; u_{ub}] \qquad j = 0, 1, ..., N_u - 1 \tag{3.2c}$$

$$\xi_t^k \in [\xi_{lb}; \xi_{ub}] \qquad k = 0, 1, ..., N_y - 1 \tag{3.2d}$$

$$\Delta u_t^{j+1} = u_t^{j+1} - u_t^j \qquad j = 0, 1, ..., N_u - 2 \tag{3.2e}$$

$$\Delta u_t^{j+1} \in [\Delta u_{lb}; \Delta u_{ub}] \qquad j = 0, 1, ..., N_u - 2 \tag{3.2f}$$

$$\xi_t^0 = \xi(t) \tag{3.2g}$$

$$\begin{aligned} J(\bar{\xi}_t, \bar{u}_t) &= \bar{\xi}_t^T Q \bar{\xi}_t + \bar{U}_t^T R \bar{U}_t \\ &= \sum_{i=0}^{N_y - 1} {\xi_t^i}^T q \xi_t^i + \sum_{j=0}^{N_u - 1} {u_t^j}^T r u_t^j \end{aligned} \tag{3.3}$$

$$Q = \begin{pmatrix} q & 0 & ... & 0 \\ 0 & q & ... & 0 \\ 0 & 0 & ... & 0 \\ 0 & 0 & ... & q \end{pmatrix} \quad ; \quad R = \begin{pmatrix} r & 0 & ... & 0 \\ 0 & r & ... & 0 \\ 0 & 0 & ... & 0 \\ 0 & 0 & ... & r \end{pmatrix} \tag{3.4}$$

Variables	Signification
T_y	State prediction horizon of the MPC
T_u	Control horizon of the MPC
Δt_y	Prediction time step of the MPC
Δt_u	Control time step of the MPC
ξ_t^k	State vector of the vehicle at time $t + k\Delta t_y$ predicted at time t ($k \in [\![0; N_y - 1]\!]$)
u_t^j	Control input vector at time $+ j\Delta t_u$ predicted at time t ($j \in [\![0; N_u - 1]\!]$)
N_y	Number of state predictions
N_u	Number of control steps

Table 3.1 – Notations for MPC (general formulation).

Remark 6. *We have* $N_y = \lfloor T_y / \Delta t_y \rfloor + 1$ *and* $N_u = \lfloor T_u / \Delta t_u \rfloor + 1$. *Necessarily,* $T_u \leq T_y$ *(i.e.* $N_u \leq N_y$*) as the control has a direct impact on the state. In the rest of this thesis, we consider* $\Delta t_u = \Delta t_y$ *and* $T_u = T_y$*, thus* $N_u = N_y$*. However, let's notice that in this case, the last control predicted, namely* $U_t^{N_u - 1}$*, has no impact on the control problem due to Equation (3.2b); therefore, it is usual to consider* $N_u = N_y - 1$.

3.2.2 Model Predictive Path Integral (MPPI)

A variant to MPC is Model Predictive Path Integral. It consists in sampling the control inputs before applying them to a vehicle simulator in order to obtain many different paths. Then, the best path is chosen [Williams et al., 2016]. The computational power available nowadays on an autonomous vehicle allows to run quickly rather complex simulation models, which can be done moreover in parallel to reduce the computational time. However, the completeness of this method is not guaranteed.

3.3 The non-holonomic constraints

Given an admissible motion for a 3D mechanical system, we know from Proposition 1 that it can be characterized by a collision-free path for a point moving in the configuration space. However, the converse is only true for *free-flying* robots, *i.e.* those who are not restricted in their motions. Therefore, a collision-free path in the configuration space does not necessarily corresponds to an admissible motion in the physical space.

This is the case for an autonomous vehicle; the non-holonomic constraints due to the slip-free rolling conditions restrict the motion of the vehicle: the lateral motion of the vehicle can only be approached by successive longitudinal and angular motions. For example, to park a vehicle in a parking spot, the number of maneuvers to conduct varies as a function of the inverse of ϵ^2, where ϵ is the free-space, *i.e.* the difference between the length of the parking spot and the vehicle [Laumond et al., 1994].

What is a non-holonomic constraint?

A non-holonomic constraint for a system is a constraint that does not reduce the dimension of its conguration space: the robot can still reach any configuration. In other words, it is a non-integrable constraint. This is the case of the slip-free rolling constraint of a vehicle. In order to better understand this concept, a comparison with the holonomic constraint that links the minute and hour hands of a watch is made below:

- *Holonomic constraint*
  The angular speed of the hour hand $\dot{y}$ and the minute hand $\dot{x}$ of a watch are linked by the following constraint:

$$\dot{y} = \frac{1}{12}\dot{x} \tag{3.5}$$

  The configuration space, corresponding to the angular position of each hand $(x, y) \in [0; 2\pi[^2$, is a tore represented on Figure 3.11 in blue. However, after integration of Equation (3.5), the admissible (or feasible) configuration space is reduced to the red spiral. Therefore, the configuration $q = (x, y) = (\pi, 0)$ corresponding to the situation on the left of Figure 3.11 and represented by the black dot on the right, is not admissible as it is not on the red spiral.

- *Non-holonomic constraint*
  The velocities $\dot{X}$ and $\dot{Y}$ of the vehicle in the inertial frame and the heading ψ of a vehicle are linked by the slip-free rolling condition:

$$\dot{Y}\cos\psi - \dot{X}\sin\psi = 0 \tag{3.6}$$

  This constraint cannot be integrated. Therefore, the vehicle can still reach any configuration $q = (X, Y, \psi) \in \mathbf{R}^2 \times [0; 2\pi[$. With only two actuators, it is still possible to reach any configuration of the three-dimensional configuration space.

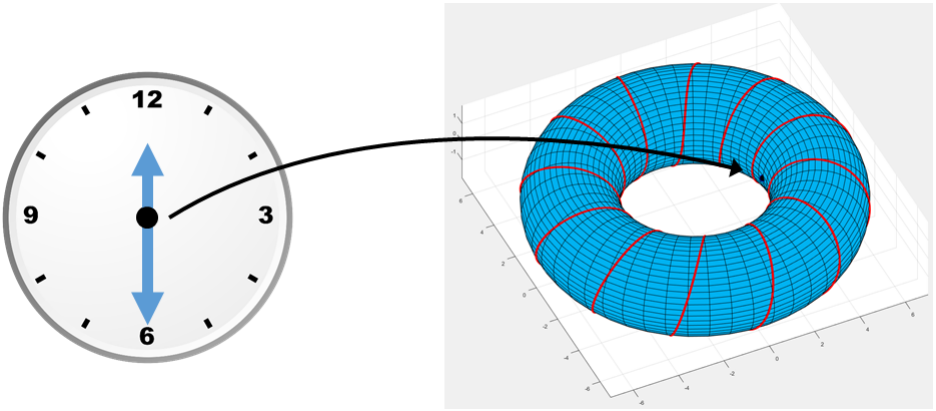


Figure 3.11 – Example of holonomic constraint: the constraint between the minute and hour hands of a watch. The configuration space (blue tore) is reduced by the constraint to the red spiral.

Impact of non-holonomic constraints on motion planning?

Although the non-holonomic slip-free rolling constraint does not reduce the dimension of the configuration space, it limits the motion of the vehicle: the vehicle cannot move solely in the lateral direction. This phenomenon has to be taken into account at the motion planning level.

Case 1: small-time controllability
In the case where the vehicle can move both forwards and backwards, at low speed, small-time controllability results proove that any reference path in the configuration space can be approached up to an ϵ. Therefore, the non-holonomic constraint does not cause any problems. The results are based on Lie Algebra: in order to check that the system is non-holonomic, Frobenius' theorem is used; then, the Lie Algebra Rank Condition (LARC) proves the small-time controllability of this non-holonomic system. Finally, the Heine–Borel theorem proves the existence of a finite covering (thus that can be obtained on a computer) of open sets joining two configurations. The main concepts of small-time controllability are given in Appendix B.1.

Case 2: curvature constraints
In the case where the vehicle can only move forwards, the non-holonomic constraints need to be considered at the motion planning level: there exists a maximal curvature γ_{max} (respectively a minimal curvature radius R_{min}) the vehicle can follow which depends on the geometry of the vehicle and is given by Equation (3.7) (δ_{max} corresponds to the maximum steering angle of the wheel). Resolution methods based on the control space often rely on a kinematic bicycle model to enforce the respect of the non-holonomic constraint, while those based on the configuration space use Dubins'paths [Dubins, 1957]: for an application on RRT, see for example [Kuwata et al., 2008].

$$R_{min} = \frac{1}{\gamma_{max}} = \frac{l_f + l_r}{\tan \delta_{max}} \tag{3.7}$$

Remark 7. *Similarly to Dubins'path, Reeds and Shepp's paths [Reeds and Shepp, 1990] can be considered for a vehicle moving both forwards and backwards (case 1) in order to obtain a smoother path for the control.*

3.4 Taking dynamic effects into account at the motion planning level

Most of the algorithms used in the motion planning community have been derived from earlier work on wheeled robots. However, a vehicle evolves in a more dynamic environment than a wheeled robot due to the number of other vehicles and pedestrians around. Therefore, it is usually no longer possible to decouple the longitudinal and the lateral behaviors as it has been done

previously: in particular, a given reference path might be safe for a given velocity profile but not for another one. Moreover, a vehicle only moves forwards except during some specific low-speed maneuvers such as parking or U-turns.

Therefore, the motion planning layer should provide directly a reference trajectory. In this case, the non-holonomic constraint have some impacts on the feasible trajectories and should be taken into account. However, compared to small wheeled robots, vehicles are heavier and faster. Thus, they have a bigger inertia, making safe operations more complex. In particular, at the wheel level, phenomena such as slip or skid might occur. Although it is complex to consider the wheel dynamics at the motion planning level, it is necessary in order to ensure the feasibility of the reference trajectory either to take such phenomena into account or to prevent them from occuring.

This problem has obtained only little attention in the motion planning literature. Wheel dynamics is usually only considered at the low-level control, assuming that a feasible reference trajectory is given (see for example [Falcone et al., 2007a]). This is due not only to the strong non-linearities of the wheel dynamics such as the computation of the longitudinal slip ratio, but also to the different time-scales of the carbody and wheel dynamics which would require a lot of computational resources (wheel dynamics requires a smaller discretization time-step). However, mass deployment of autonomous vehicles in the future requires to be able to guarantee their safety at all time. For that purpose, two main techniques can be applied: model reduction and validity domain delimitation.

Model reduction

To account for vehicle dynamics, it is possible to simplify the complexity of the full vehicle model presented in chapter 2 using simpler relations with less variables, reducing thus the complexity of the model. Some loss of accuracy would be ineluctable but the computational time would be reduced. In particular, motion planning deals with the motion of the vehicle in the real world. Therefore, it is more intuitive to control accelerations to study the evolution of the carbody, even though forces can also be controlled. Three examples of how to simplify the complex vehicle model in order to be used at the motion planning level are presented:

- **GG-diagram and friction circles** - The friction limits induce an envelope for the possible accelerations of a vehicle. This envelope can be represented by a so-called “friction circle". In order to account for the assymettry between the maximal acceleration and deceleration, a “gg-diagram" (see Figure 3.12) has been introduced in [Rice, 1973]. It has been applied to design aggressive-driving controllers [Kritayakirana and Gerdes, 2012, Funke et al., 2012]. A few works such as [Zhan et al., 2016] include it at the motion planning level too. However, it neither captures the non-holonomic constraint nor the load transfer between tires (which modifies the shape, or at least the radius, of the gg-diagram).

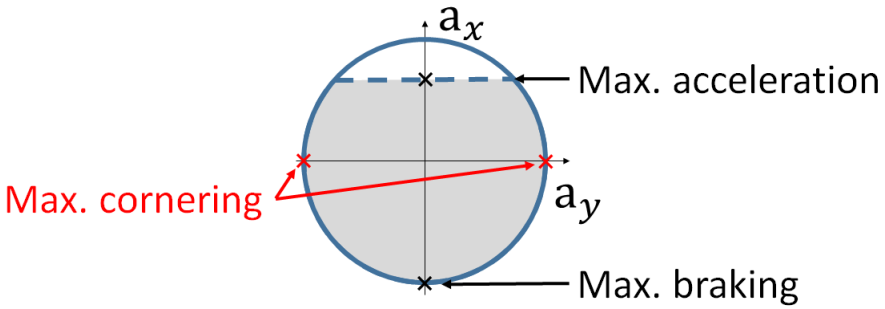


Figure 3.12 – A “gg-diagram"

- **Load transfer** - In order to simplify the four-wheeled vehicle model, a bicycle model is often considered. In that case, the load transfer between the left and the right wheels are not

considered anymore, which has a direct impact on the maximal tire forces that can be generated on each tire due to the friction circle. However, it is still possible to compute the load transfer without computing the roll and pitch angles of the carbody (see for example [Funke et al., 2017]): at steady state, assuming a small roll angle, Equation (3.8) can be used to ajust the radius of the friction circle; similarly, at steady-state, the load transfer between the front and rear tires assuming a small pitch angle is given by Equation (3.9):

$$F_{zl} = M_T \frac{l_w g + h_{CoG} a_y}{2 l_w} \quad ; \quad F_{zr} = M_T \frac{l_w g - h_{CoG} a_y}{2 l_w} \tag{3.8}$$

$$F_{zf} = M_T \frac{l_r g - h_{CoG} a_x}{l_f + l_r} \quad ; \quad F_{zr} = M_T \frac{l_f g + h_{CoG} a_x}{l_f + l_r} \tag{3.9}$$

- **Offline computation** - As an accurate vehicle model is usually computationally expensive, it is possible to do some offline precomputation to speed up the online computation. This is the case for example of explicit MPC [Besselmann and Morari, 2009]. However, to the best of our knowledge, this technique has never been apply at the motion planning level. Instead, a variant has been proposed in [Altché et al., 2017]: the achievable acceleration enlevope of a vehicle depending on its initial state, such as its speed and steering angle is precomputed offline. The trajectory planning is then done using a simple point-mass model with linear constraints on the accelerations obtained using the precomputed acceleration envelope. The problem is then very efficient from a computational point-of-view (QP-problem with linear constaints) but its application on an actual vehicle might be challenging as one needs to obtain the acceleration envelopes.

  Finally, some works precompute motion primitives: for example, in [Gray et al., 2012], a feasible trajectory is generated offline from predefined motion primitives and a MPC using a simplified four-wheel vehicle model performs the path tracking. The use of motion primitives transform the continuous MPC optimization problem into a combinatorial one by choosing the best sequence of actions to perform.

Validity domain delimitation

When using simplified models for motion planning, it is important to ensure that these models operate inside their validity domain in order to plan feasible trajectories. These validity domains depend both on the inner state of the vehicle and on external factors such as the road surface.

Several works have been dealing with such problems in the literature and have derived some simple criteria to check the consistency of a model. For example in [Park et al., 2009], the dynamic bicycle model was proven to remain valid as long as $a_y \leq 0.5g$ and $|\beta| \leq 4°$ where a_y is the lateral acceleration and β is the slip angle at center of gravity. In chapter 4.2, we will prove that the kinematic bicycle model can only be used if $a_y \leq 0.5\mu g$ on curves with constant curvature traveled at constant speeds. If these criteria are not satisfied, the modeling errors become huge and these models would be completely unable to capture the vehicle's actual behavior. At last, let us mention [Schürmann et al., 2017] who uses a point-mass model at the motion planning level but checks afterwards if the planned maneuver is realizable using a non-deterministic system modeled as a differential inclusion.

3.5 Conclusion

Due to its ability to handle complex vehicle models and the speed of todays solvers for optimal control problems, Model Predictive Control is probably the most promising method for generating feasible reference trajectories. In particular, it is able to coupled the longitudinal and lateral motion of the vehicle simultaneously as path planning and velocity control cannot be decoupled anymore in a highly dynamic environement.

Although the computational complexity of MPC can be suited for real-time applications, it depends strongly on the vehicle model used: hence, it is necessary to make a good compromise between its accuracy and its complexity. Therefore, the choice of the model will be discussed in the next section. Moreover, this can have an impact on the type of motion planning and control architecture that will be chosen, namely an integrated approach where they are both unified in a single framework, or a hierarchical approach where they are split into two separate layers.

Chapter 4

Guaranteeing the “feasibility" of the kinematic bicycle model

“ In any moment of decision the best thing you can do is the right thing.
The worst thing you can do is nothing. ”

Theodore Roosevelt

Contents

As mentioned already in the previous chapter, many planning and control algorithms used for autonomous vehicles have been derived from earlier work on wheeled robots. For example at the planning level, algorithms such as RRT or potential fields have been employed (see chapter 3) while at the control level, the pure-pursuit [Coulter, 1992] is still quite popular. The technological transfer from wheeled robots to autonomous vehicles has shown good results so far as the latter are moving in confined environments and at rather low speeds. However, as we move towards their mass deployment, autonomous vehicles will encounter more and more dynamic obstacles. Moreover, they will only be allowed to move backwards in some very particular cases such as parking or U-turn maneuvers.

Therefore, the decoupling between the longitudinal and lateral motions of the vehicle cannot be considered anymore at the planning level. The latter should provide not only a reference path but also a reference velocity, *i.e.* a reference trajectory: as already mentioned, a portion of the path can be safe at a given moment but not a few seconds latter.

The Model Predictive Control (MPC) framework presented in chapter 3.2.1 appears as a promising method in order to plan reference trajectories for autonomous vehicles: not only is it able to handle easily constraints such as vehicle dynamics, but also it allows semi-analytical resolution which can make this method extremely fast to compute [Houska et al., 2011a]. Therefore, it presents a natural framework to deal with the coupling of longitudinal and lateral dynamics of a vehicle. Usually, MPC is used for control purpose where the optimal control input sequence to apply to the system is obtained. However, it can also provide the optimal state sequence, such as a reference trajectory. One question that arises then and still need to be addressed is the choice of a model for representing the evolution of the system, given by Equation (3.2b).

A *model* is a mathematical description of the evolution of a system, *i.e.* a representation of the "real world". It plays a fundamental role in a large variety of applications such as simulation, control theory, motion planning, prediction intention. It enables not only to explain a system, but also to predict its evolution and thus anticipate its behavior. It can usually be written in the continuous form given by Equation (4.1), named state-space model, where ξ is the state vector of the system and u its input vector.

$$\dot{\xi} = f(\xi, u) \tag{4.1}$$

There exists a wide range of models for representing a system. The choice is a trade-off between two conflicting objectives: the level of abstraction and accuracy on the one hand, and the computational time on the other hand. The more detailled the model is, the better the predictions are but the longer it takes to obtain them. The choice of a model depends also on the information available: fine models require a good knowledge of the parameters of the system as they might not be robust to large errors on these values. Also, in order to derive more general results, it can be useful not to detail to much the modeling: typically, making abstraction of the engine for an autonomous vehicle enables to consider indifferently both electric and gas engine vehicles.

In chapter 2.4, we introduced different vehicle models, namely the point-mass, the unicycle, the kinematic bicycle and dynamic bicycle models. This chapter discusses which models are the most suited for motion planning applications and how to integrate them in a motion planning and control architecture. Therefore, section 4.1 presents the two main categories of architecture, namely the hierarchical and the integrated approaches, and which models they can be associated with. The rest of this chapter focuses then on the kinematic bicycle model, which is the one we chose: in particular, section 4.2 compares this model with the simulation model developed in chapter 2. Then, a motion planning and control architecture for planning trajectories that are not only safe but also efficient from a traffic point of view is proposed in section 4.3. Finally, some enhancements of the architecture are proposed in section 4.4 to handle roads with a low friction coefficient.

4.1 Hierarchical versus integrated motion planning and control architecture: a choice of model

In the literature, many works can be found on motion planning and control architectures. They can be classified into two categories [Berntorp, 2017]: hierarchical and integrated approaches.

4.1.1 Integrated motion planning and control architecture

In an integrated approach, motion planning and low-level control are unified into one unique MPC formulation such as in [Jiang et al., 2016, Liniger et al., 2015, Gao et al., 2010]. Different vehicle models can be used as long as the control inputs correspond to the ones of the low-level controllers, for example the steering angle and the wheel torques. Therefore, the dynamic bicycle model presented in chapter 2.4.4 is often chosen. This model is more time consuming than the one used for planning in a hierarchical approach. One of the main assets of such a method is that the reference trajectory obtained is feasible (if the model is well chosen). However, the long computational time limits the prediction horizon and thus the speed of the vehicle. Moreover, integrated approaches are not robust to modeling errors, disturbances and parameter uncertainties as the system evolves in open-loop during its recomputational time, which has a low frequency compared to conventional low-level controllers (around 200ms).

Except from using simple lateral tire models such as the linear tire model or the Fiala one, the wheel dynamics is normally not considered in the modeling for two reasons: first, the wheel dynamics is much faster than the carbody dynamics. Therefore, it would require a very small control time step in the MPC formulation which is unsuited for real-time application. Secondly, the wheel dynamics is highly non-linear, a good example beeing the computation of the longitudinal slip ratio τ_x (see chapter 2.2). Therefore, some simplifications are still necessary such as in [Falcone et al., 2007b] where the slip ratio is assumed to remain constant on the whole prediction horizon.

4.1.2 Hierarchical motion planning and control architecture

The hierarchical approach separates the motion planning from the low-level control to overcome the problem of long prediction horizon and disturbances. Two independent layers are thus obtained: first, the motion planning computes a reference trajectory; then, the latter is fed to the controllers in order to be tracked. Examples comprise: [Rosolia et al., 2017] which uses a unicycle model for the planning layer and a dynamic bicycle model for the control; [Li et al., 2017] which uses a unicycle model with a constant constraint on the maximal curvature and a dynamic bicycle model for the control; [Abbas et al., 2014] which uses a kinematic bicycle model with a constant speed assumption for planning a reference trajectory; [Ziegler et al., 2014b] uses a point mass model with a constraint on the curvature.

In the motion planning layer, simple models such as a point-mass[1] or a unicycle model are usually used in order to generate a safe reference trajectory with a long prediction horizon. The low-level controllers are assumed to be able to track the reference trajectory obtained by choosing a very detailled vehicle model able to drive at the vehicle's handling limits, typically a dynamic bicycle model at least. In a sense, the controller is assumed to be “perfect", able to follow any given reference trajectory [Zhan et al., 2016, Qian et al., 2016, Nilsson et al., 2015a].

However, using simple models at the motion planning level might lead to plan unfeasible reference trajectories as these models are unable to capture the kinematic and dynamic effects presented in chapter 3.3 and 3.4. This problem, critical for the safety of the vehicle, will arise as the number of traveled kilometers increases, the driving strategy becomes less conservative and the road condition changes.

[1] Using a quadratic cost function and a point-mass model, the optimization problem of Equations (3.2) becomes a Quadratic Problem with linear constraints which can be solved in polynomial time using an ellipsoid method [Kozlov et al., 1979].

In particular, a hierarchical approach using a simple model at the motion planning layer might lead to the situation depicted in Figure 4.1: the motion planner of the red vehicle outputs a reference trajectory (in blue) which avoids the yellow vehicles and is smooth. However, due the high initial velocity V_0 of the vehicle and the modeling error at the motion planning level, it turns out that the vehicle cannot do better than the red dotted lines, which is the handling limits of the vehicle. Moreover, taking into account the imperfection of the controllers (ex: measurement noise, delays, uncertainties, stability-accuracy dilemma, actuator limitation), the vehicle ends up following the yellow line which is no longer safe and hits the back of the yellow vehicle at high speed.

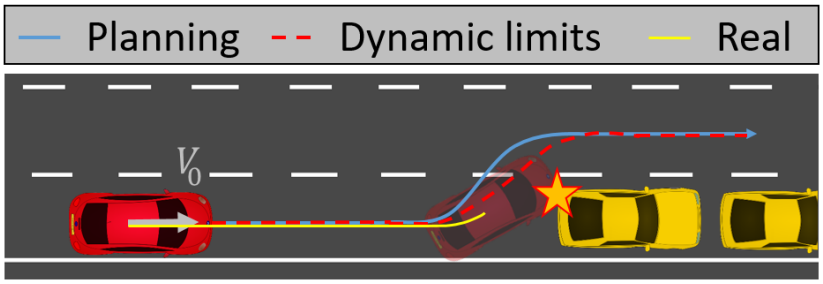


Figure 4.1 – Limits of the “perfect" controller assumption.

Therefore, there is a need to improve the modeling at the motion planning level. For example, for a point-mass model, it is possible to account for some dynamic phenomena such as the friction circle by limiting the Euclidean norm of the total acceleration of the vehicle $\sqrt{a_X^2 + a_Y^2} \leq a_{max}$ [Ziegler et al., 2014b]. A “gg-diagram" (cf. chapter 3.4) which takes into account the asymetry between traction and braking can also be used [Zhan et al., 2016]. The non-holonomic constraint can be approached by constraining the lateral speed to be lower than the longitudinal speed (see Equation (2.41)). However, such an approximation is limited to almost straight roads [Qian et al., 2016]. For the unicycle model, the non-holonomic constraint is respected by the model but dynamic effects are not considered. Therefore, the results obtained do not guarantee the feasibility of the reference trajectory at all time.

Some alternative solutions have been presented to compute feasible reference trajectories using a point-mass model. [Altché et al., 2017] computes offline the achievable acceleration envelope depending on the initial state of the vehicle, such as the speed and the steering angle. A linear point-mass model with linear constraints on the admissible acceleration envelope is then used, which is very efficient from a computational point-of-view. However, it seems difficult to get such acceleration envelopes on an actual vehicle. [Schürmann et al., 2017] uses a point-mass model as motion planner but checks afterwards if the planned maneuver is realizable using a non-deterministic system modeled as a differential inclusion.

The problem of feasibility of the reference trajectory becomes even more important for hierarchical approaches in the case of roads with a low friction coefficient as slips and skids are more likely to occur. Although some low-level controllers that can handle such situations have been proposed in the literature, see for example [Falcone et al., 2007a, Gao et al., 2010, Gray et al., 2012], obtaining a feasible reference trajectory remains a challenging task: in [Gao et al., 2010], the authors used a point-mass model for planning a reference trajectory on an icy road which gives no guarantee on the trajectory obtained and in [Gray et al., 2012], a feasible trajectory is generated offline from predefined motion primitives. The use of motion primitives transform the continuous MPC optimization problem into a combinatorial one by choosing the best sequence of actions to perform.

4.1.3 The kinematic bicycle model: a consistent model for planning feasible and safe trajectories?

In order to ensure that the planned trajectory is feasible, it is important to choose a consistent model, also referred to as “proper modeling" in the literature [Ersal et al., 2008]. In particular, it

is important to know the limits of validity of a given model. For that purpose, models have been compared to more complex ones in the literature. For example, [Liu et al., 2013] compares two integrated motion planning and control MPC: one using a 2 Degrees of Freedom (2 DoF) dynamic bicycle model with a linear tire model (Version 2 in chapter 2.4.4) and another one based on a 14 DoF vehicle model. The validity of this dynamic bicycle model is guaranteed by constraining the lateral acceleration to $0.5g$ on normal road conditions, condition which has been derived in [Park et al., 2009]. However, the practical implementation of this system on a real vehicle is limited as it assumes a constant velocity. [Kong et al., 2015] compares the performances of MPC using a kinematic bicycle model and one using a dynamic bicycle model with a linear tire model (Version 2 in chapter 2.4.4). The authors conclude that the kinematic bicycle model works better in most cases, except at high-speed, and suggests a further investigation on the impact of different lateral accelerations on the validity of the kinematic bicycle model.

The kinematic bicycle model takes into account the non-holonomic constraint of a vehicle, *i.e.* the slip-free rolling condition defined in section 3.3, but not the dynamic effects as it neglects the slip angle at each wheel. Therefore, it is usually considered as a bad approximation for high-speed driving or roads with low-friction coefficients: in [Rajamani, 2012] for example, it is not recommended to use this model for velocities higher than $5m/s$. However, a careful human driver tries to limit the slip and skid even in the case of slippery roads. He simply ajusts the longitudinal speed to the situation. Therefore, the kinematic bicycle model seems a good representation of the behavior of a vehicle in normal driving situations. Its relative simplicity and the low number of vehicle parameters necessary (only the distance between the center of gravity to the front and the rear axles, respectively l_f and l_r) make it a good choice for modeling the behavior of a vehicle. However, if used for motion planning, it is important to ensure that the no slip assumption holds, especially as we move towards faster autonomous vehicles and all kind of weather conditions, where slip and skip are more likely to occur. Therefore, in the next section, we will try to derive some criteria that can be used at the motion planning level in order to ensure the validity of the kinematic bicycle model.

4.2 Validity of the kinematic bicycle model

One of the major assumption behind the kinematic bicycle model is that the slip angle at each wheel is almost zero. *i.e.* that the dynamic effects such as slip and skid can be neglected. However, this assumption does not hold for example for race driving or emergency maneuvers.

This almost-zero slip angle assumption is complex to check in practice on a vehicle as it corresponds to the wheel dynamics, which not only is much faster than the carbody dynamics (see section 2.2) but also is limited in sensors (only wheel encoders are available on cheap autonomous vehicle). Hence, we compare the kinematic bicycle model presented in section 2.4.3 with the high-fidelity 9 DoF dynamics model[2] presented in section 2.3 in order to derive some high-level practical bounds where both models coincide. These criteria could then be used at the planning level in order to ensure the validity of the kinematic bicycle model more easily.

4.2.1 Experimental set-up

In order to compare the kinematic bicycle model and the 9 DoF vehicle model, circles with different curvature radius R traveled at constant speed V were chosen as reference trajectories. Such trajectories correspond to constant control inputs, $u_1 = 0$ and $u_2 = 0$ (Equations (2.43e) and (2.43f)), for the kinematic bicycle model. Equations (2.43) become then Equations (4.2) where β remains

[2] The vertical motion of the vehicle is neglected and $Z(t) = h_{CoG}$.

constant.

$$X(t) = R\sin(\psi(t)+\beta)+X(0) \tag{4.2a}$$
$$Y(t) = -R\cos(\psi(t)+\beta)+Y(0) \tag{4.2b}$$
$$\psi(t) = \frac{V}{R}t+\psi(0) \tag{4.2c}$$

As the yaw rate $\dot{\psi} = V/R$ is constant, Equation (2.43d) becomes Equation (4.3) where δ_{th}, the theoretical steering angle obtained using a kinematic bicycle model, is a function only of the curvature radius R, not of the speed V. More precisely, there is a one-to-one correspondence between the curvature radius and the steering angle, when using the kinematic bicycle model with constant speed.

$$\delta_{th}(R) = \operatorname{atan}\left(\left(\frac{l_f}{l_r}+1\right)\tan\left(\operatorname{asin}\left(\frac{l_r}{R}\right)\right)\right) \tag{4.3}$$

Note that Equation (4.3) is valid only for $l_r \leq |R|$, meaning that the minimal curvature radius of the path has to be larger than the distance between the rear axle and the center of gravity of the vehicle.

In the rest of this section, we are going to check the validity of Equation (4.3) and thus of the kinematic bicycle model. For that purpose, we will compare the steering angle δ_a applied in the 9 DoF vehicle model simulator with the theoretical steering angle δ_{th} obtained by Equation (4.3). The values of the different main parameters are the following: $l_f = 1.17$m, $l_r = 1.77$m, $l_w = 0.81$m, $M_T = 1820$kg and $\mu = 1$.

4.2.2 Limits of Equation (4.3)

The kinematic bicycle model assumes for constant speed that the curvature radius of the path described by the vehicle depends only on the steering angle, not on the speed. In order to test the validity of this assumption, we compare the results obtained by a kinematic bicycle model with those of the 9 DoF vehicle model. For that purpose, we run several 9 DoF vehicle simulations. Each simulation is parameterized by a constant steering angle δ_a and a constant wheel torque command T_ω applied at the front wheels. The initial speed of the vehicle is $V_x = 10$m/s and $V_y = 0$m/s and the friction coefficient μ of the road is assumed to be equal to 1. More precisely, for each steering angle δ_a chosen, we run several simulations with different values of wheel torque command T_ω. The system ends up describing a circle with constant speed (see Figures 4.2 and 4.3), except if the wheel torques are negative or too high. These solutions were discarded.

Figure 4.4 shows the curvature radius R of the circle obtained with the 9 DoF vehicle model versus the speed V for different values of the applied steering angle δ_a. In dashed lines, we added the constant curvature R_{th} obtained with the kinematic bicycle model from Equation (4.3). While at low speed, $R = R_{th}$, this is not the case for higher speed. In particular, as the speed or the steering angle increase, the error becomes huge. For example, at $V = 25$m/s, the curvature radius error is 4.0% for $\delta_a = 1°$ but 86.0% for $\delta_a = 4°$. Thus the limits of the kinematic bicycle model are highlighted.

4.2.3 Adding a constraint on the lateral acceleration

We have obtained in Figure 4.4 a curve $R = f(V)$ for each steering angle δ_a applied in the vehicle simulator. From Equation (4.3), we know that for each radius R corresponds a unique theoretical steering angle δ_{th} obtained from the kinematic bicycle model. Therefore, in Figure 4.5, the theoretical steering angle δ_{th} corresponding to the radius R obtained above is displayed as a function of V. We can now compare the steering angle theoretically obtained using a kinematic bicycle model, δ_{th}, with the one really applied, δ_a, in order to obtain the same curvature radius at different constant speeds. At low-speed, we observe that δ_a and δ_{th} are equal, meaning that the kinematic bicycle model and the 9 DoF dynamic model coincide. However, at higher speeds, δ_{th}

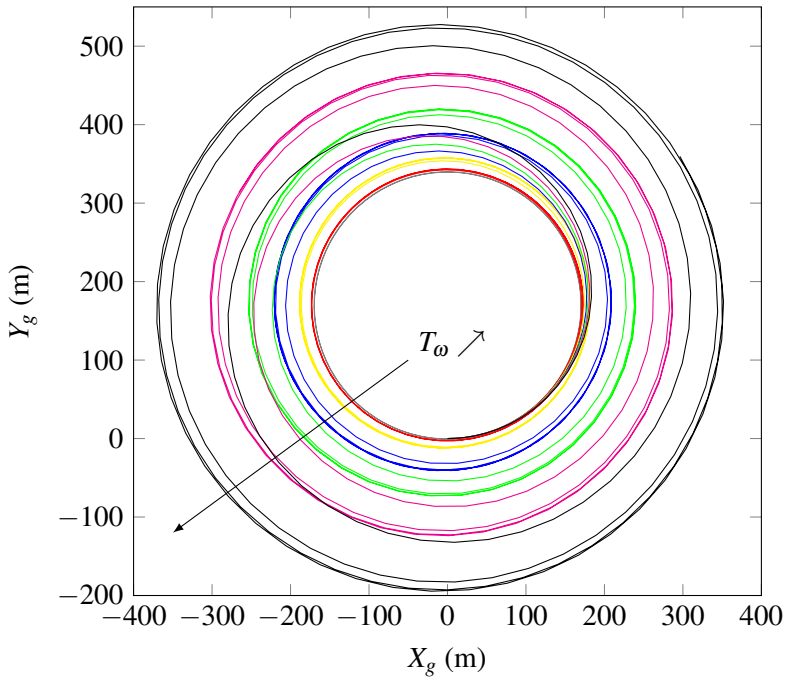
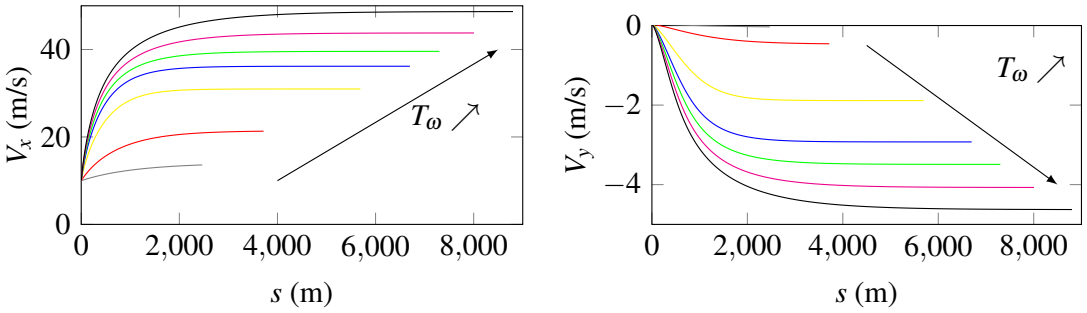


Figure 4.2 – Example of trajectories obtained with $\delta_a = 1°$ for different wheel torques T_ω applied.



(a) Longitudinal speed profiles.

(b) Lateral speed profiles.

Figure 4.3 – Example of speed profiles obtained with $\delta_a = 1°$ for different wheel torques T_ω applied. The colors matches with the trajectories displayed in Figure 4.2.

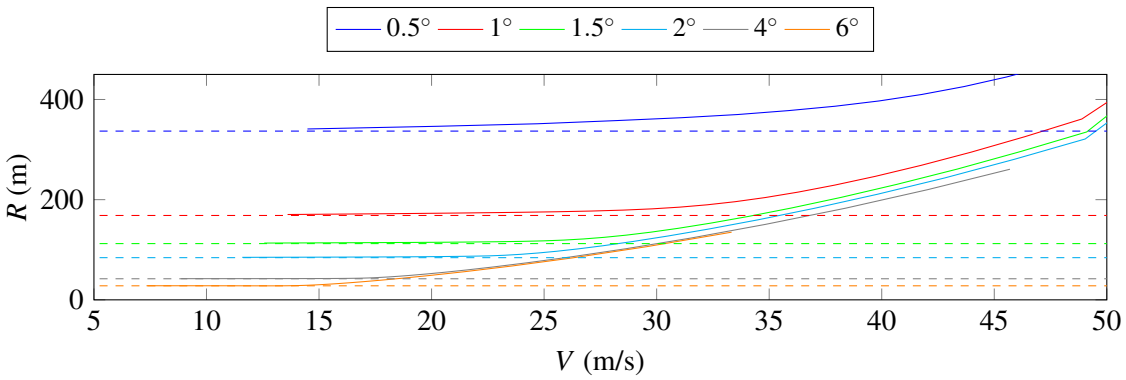


Figure 4.4 – Curvature radius versus speed for different steering angles obtained with the 9 DoF vehicle model (full lines) and the kinematic bicycle model (dashed lines).

decreases rapidly, except at low steering angle δ_a. Thus, the vehicle is understeering in the simulation model compared to the kinematic bicycle model at higher speeds and higher steering angles δ_a. This is due to side slip. Therefore, while the kinematic bicycle model seems a good modeling for low-speed vehicles or low steering angles, it is completely unable to capture the motion of the vehicle at higher speeds or steering angles. Hence, using it for planning purpose might lead to dynamically unfeasible trajectories for the vehicle to follow.

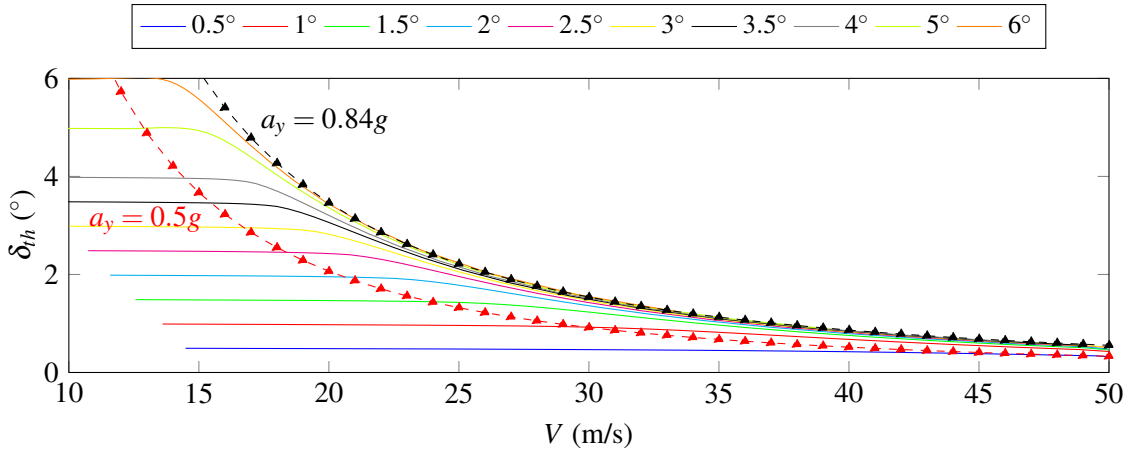


Figure 4.5 – Theoretical kinematic steering angle δ_{th} obtained by applying a steering angle δ_a on the 9DoF model (full line), and maximum kinematic steering angle δ_{max} allowed for different maximum lateral accelerations $a_{y_{max}}$ (triangles).

The lateral acceleration a_y of the vehicle can be expressed as a function of its speed V and the curvature radius R:

$$a_y = \frac{V^2}{R} \tag{4.4}$$

Therefore, we computed a_y for the different values of V and R obtained on Figure 4.4. The results are shown in Figure 4.6. We observe that the maximal lateral acceleration that can be obtained is $a_{y_{max}} = 8.22\text{m/s}^2 = 0.84g$.

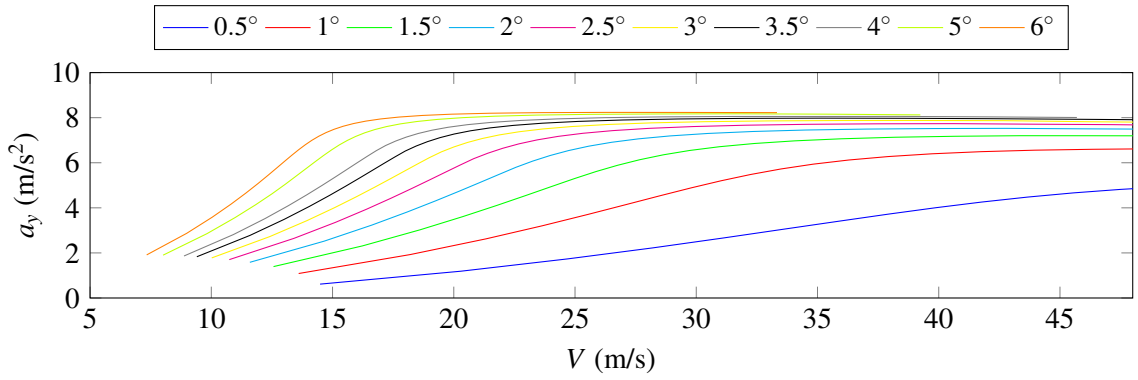


Figure 4.6 – Lateral acceleration versus speed for different steering angles obtained with the simulation model.

Moreover, let's suppose that we constraint the lateral acceleration of the vehicle to remain below a certain value $a_{y_{max}}$. Thus, Equation (4.3) leads to a maximum steering angle δ_{max} allowed for the kinematic bicycle model which is given by Equation (4.5).

$$\delta_{max} = \text{atan}\left(\left(\frac{l_f}{l_r}+1\right)\tan\left(\text{asin}\left(\frac{a_{y_{max}} l_r}{V^2}\right)\right)\right) \tag{4.5}$$

In Figure 4.5, we added the curves δ_{max} as a function of V for a maximum lateral acceleration corresponding to $a_{y_{max}} = 0.5g$ and $a_{y_{max}} = 0.84g$. Let's notice that the curve δ_{max} for $a_{y_{max}} = 8.22\text{m/s}^2 = 0.84g$ corresponds to the maximum steering angle enabled at a given speed. More interestingly, we observe that for a lateral acceleration below $0.5g$, the steering angle δ_a applied to the 9 DoF vehicle simulator matches with the theoretical steering angle δ_{th} obtained using a kinematic bicycle model: $\delta_{th} \approx \delta_a$. Thus, the kinematic bicycle model and the 9 DoF dynamics vehicle model coincide in this case. Moreover, above this limit, δ_{th} becomes much smaller than δ_a: the kinematic bicycle model is no more able to capture correctly the vehicle's behavior as dynamic effects become non negligible. In particular, prediction errors for trajectory planning would become large, leading to unfeasible trajectories from a dynamic point of view. These cases should be avoided. Therefore, this criteria on lateral acceleration seems a good one for ensuring the validity of the kinematic bicycle model.

Thus, on dry road ($\mu = 1$), the kinematic bicycle model remains valid as long as the lateral acceleration of the vehicle respects the constraint given by Equation (4.6). In cases where higher lateral accelerations are necessary, other models should be used to plan the future trajectory of the vehicle.

$$a_y \leq 0.5g \tag{4.6}$$

4.2.4 Relaxation of the dry road assumption

We run the same simulations as before but this time with $\mu = 0.7$ in the 9 DoF vehicle simulator which corresponds to a wet road [Persson et al., 2005]. The results are shown on Figure 4.7 and 4.8. We observe again the importance of limiting the lateral acceleration of the vehicle in order for the kinematic controller to be valid. In particular, limiting the lateral acceleration to $a_{y_{max}} = 0.5\mu g$ in Figure 4.8 ensures the validity of the kinematic bicycle model as $\delta_{th} \approx \delta_a$.

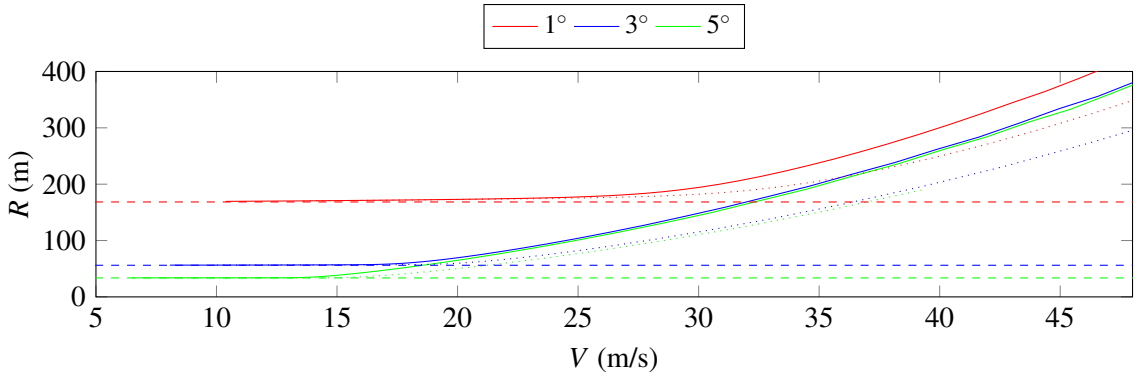


Figure 4.7 – Radius versus speed for different steering angles obtained with the 9DoF model for $\mu = 0.7$ (full lines), $\mu = 1$ (dotted lines) and with the kinematic bicycle model (dashed lines).

4.2.5 Conclusion

The kinematic bicycle model appears to be a good model for representing the behavior of a vehicle when the lateral acceleration remains below the criteria defined by Equation (4.7), where g is the gravity constant and μ the road friction coefficient:

$$a_y \leq 0.5\mu g \tag{4.7}$$

Therefore, it can be used to generate feasible reference trajectories, corresponding to the green region of Figure 4.9. If the lateral acceleration is not constrained, the model errors can become very large and might lead to plan trajectories in the black region of Figure 4.9 which are dynamically unfeasible for a vehicle. Therefore, no controllers will be able to track the reference trajectory,

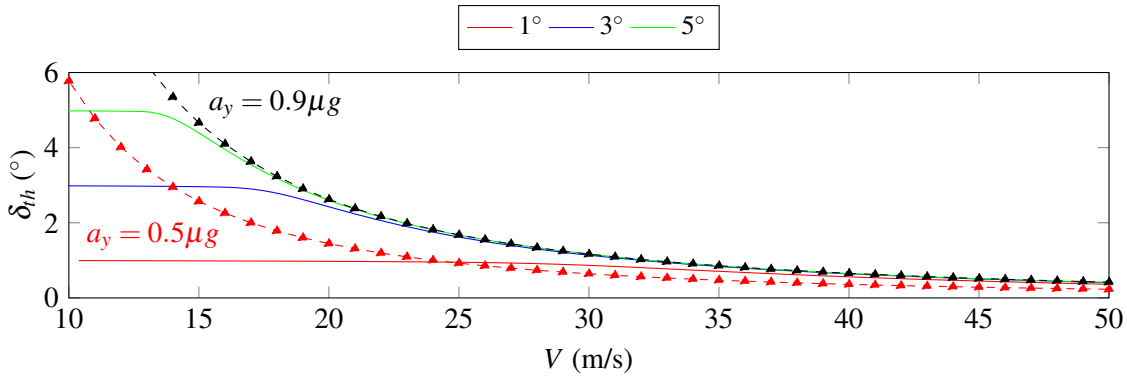


Figure 4.8 – Theoretical kinematic steering angle δ_{th} obtained by applying a steering angle δ_a on the 9DoF model with $\mu = 0.7$ (full line), and maximum kinematic steering angle δ_{max} allowed for different maximum lateral accelerations $a_{y_{max}}$ (triangles).

putting the whole system in jeopardy... Note that the red region in Figure 4.9 is still feasible from a dynamic point of view but requires low-level controllers able to handle the vehicle at its limits.

Moreover, the kinematic bicycle model gives some interesting results such as a simple relation between the speed of the vehicle V and the curvature radius R that can be achieved as shown in Equation (4.8). This result is obtained by combining Equations (4.4) and (4.7).

$$V \leq \sqrt{0.5\mu g R} \tag{4.8}$$

Such a results is very useful for trajectory planning as it enables to set a maximal speed V_{max} that depends on the shape of the road and the road conditions. Similarly, as there is a relation between the curvature radius R and the steering angle δ_{th} for the kinematic bicycle model, the maximum steering angle δ_{max} allowed in order to respect the lateral acceleration constraint can be expressed as a function of V and μ, given by Equation (4.9). This result is obtained by combining Equations (4.3), (4.4) and (4.7).

$$\delta_{max}(V,\mu) = \operatorname{atan}\left(\left(\frac{l_f}{l_r}+1\right)\tan\left(\operatorname{asin}\left(\frac{0.5\mu g l_r}{V^2}\right)\right)\right) \tag{4.9}$$

Thus, the kinematic bicycle model with the constraint on lateral acceleration represents well the behavior of a normal vehicle driver. In cases, say an emergency maneuver, where higher lateral accelerations are expected to push the vehicle near its handling limits, other models or methods should be considered. Moreover, the comparison between the 9 DoF vehicle model and the kinematic bicycle model presented here do not consider the coupling effects between the longitudinal and lateral dynamics due to the constant speed assumption.

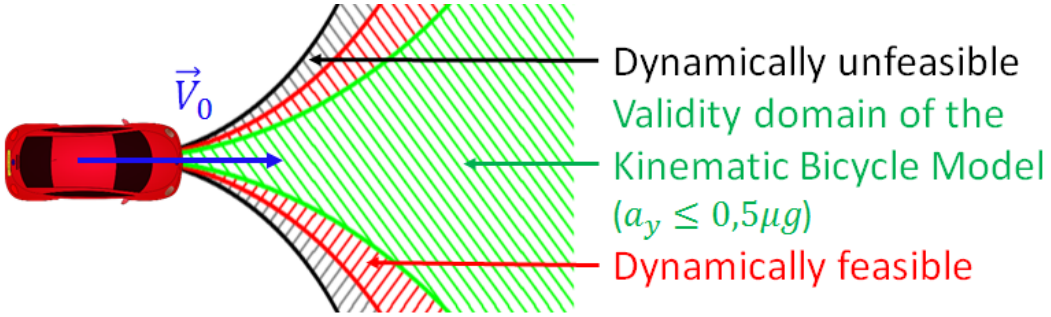


Figure 4.9 – Validity of the kinematic bicycle model.

4.3 A consistent planning and control architecture for normal driving situations

In the previous section, we have presented the different models that can be used in a MPC framework in order to obtain a reference trajectory for autonomous vehicles. In particular, we have shown the assets of using a kinematic bicycle model in order to generate feasible trajectories as we have a criteria on the maximum lateral acceleration allowed to ensure the model validity. Therefore, in this section, we are going to propose a consistent planning and control architecture in order to cope with most of the driving situations.

The goal of a planning and control architecture is to compute the control inputs, *i.e.* the steering angle and wheel torques, that should be applied on the actuators of the vehicle in order to avoid obstacles located on the path. It also defines a speed profile for the vehicle to follow. We propose an architecture composed of four main components to deal with most driving situations, namely a local velocity planner, an obstacle manager, a local MPC planner and low-level controllers. These modules, shown in Figure 4.10, are detailed respectively in sections 4.3.1 to 4.3.4. The main characteristics of our architecture are:

- A smart computation by the local velocity planner of a heuristic speed V_{heur} based on the lateral constraint for the kinematic bicycle model, to guide the MPC towards high-speed maneuvers.
- A combination of a 10Hz MPC planner for anticipating new situations, with a high-frequency (100Hz) closed-loop low-level controller for a better tracking of the reference trajectory.
- A validity guarantee of the kinematic bicycle model used in the MPC formulation by constraining the lateral acceleration, generating thus only feasible trajectories to follow for the low-level controllers.

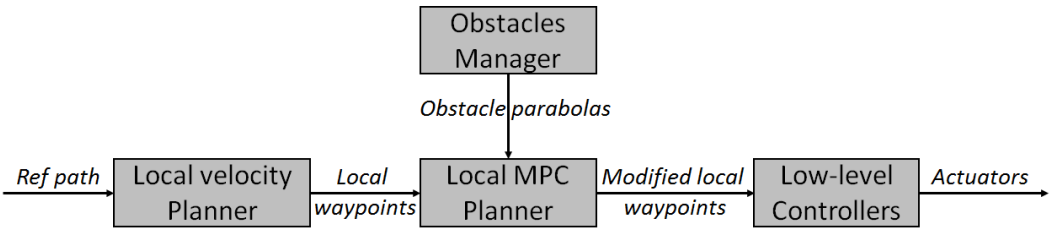


Figure 4.10 – The planning and control architecture proposed.

4.3.1 Local velocity planner

The reference input of the planning and control architecture only comprises the future coordinates $(X_i, Y_i)_{i\in\mathbb{N}}$ of the center line of the road. However, in order not to be too conservative while ensuring the safety of the vehicle at all time, the system should foster higher but safe speeds. Therefore, a velocity term should be added to the cost function of the local MPC planner (see section 4.3.3). Choosing a cost such as $||V - V_{max}||^2$ with V_{max} constant would cause the velocity cost to change too much according to the situation, thus making it difficult to tune its weight. A small weight would lead the MPC to be very conservative when it could drive at higher speeds, while a bigger weight would cause safety problems when the velocity should be low, as the optimal solution might be to drive fast rather than to respect some safety constraints.

Therefore, we introduced a local velocity planner that provides a heuristic velocity V_{heur} defined by Equation (4.10). It takes into account the current speed of the vehicle V, a predefined maximum speed allowed V_{max} and the future coordinates $(X_i, Y_i)_{i\in\mathbb{N}}$ of the reference path. The strategy consists in increasing the actual speed of the vehicle V by ΔV while V_{max} is not reached,

except if the curvature $\gamma_{max} = 1/R_{min}$ of the path in the next T_{prev} seconds leads to a lateral acceleration $a_y = \gamma_{max} V^2$ over the authorized limit for a kinematic bicycle model of $0.5\mu g$. V_{max} is defined specifically for straight lines, to ensure that the vehicle is able to come to a full stop if necessary at the end of the prediction horizon T_H of the MPC.

$$V_{heur} = \min\left(\sqrt{0.5\mu g R_{min}}, V_{max}, V + \Delta V\right) \tag{4.10}$$

Remark 8. *The local velocity planner only gives a heuristic velocity for the local MPC trajectory planner in order to obtain a velocity cost that does not depend to much on the velocity. The goal of the MPC is not to track it perfectly. Therefore, it does not need to be dynamically feasible.*

4.3.2 Obstacle manager

The obstacle manager module defines a parabola around each surrounding obstacle. More precisely, each obstacle o is represented by a parabola p_0 with directrix parallel to the reference path at its point closest to o as shown in Figure 4.11; the parameters of p_0 are chosen such that it is the minimal parabola containing all vertices of o. The interior points of the parabola defines then a region that is forbidden for the vehicle, and will be used as a constraint in the local MPC planner. As mentioned in chapter 1.3.1, the problem of obstacle detection is beyond the scope of this thesis.

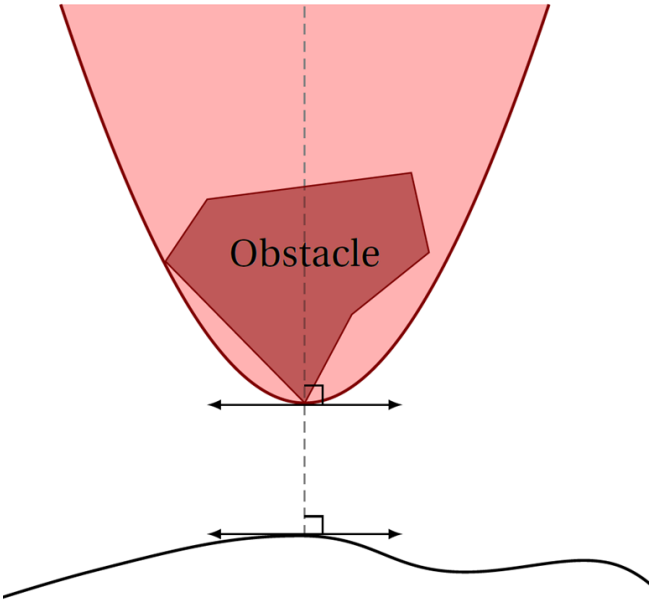


Figure 4.11 – Parabola defined by the obstacle manager.

4.3.3 Local MPC trajectory planner

The local MPC planner computes a reference trajectory for the vehicle that will then be tracked by the low-level controllers. This trajectory must be:

- *Safe*: the trajectory is collision-free, *i.e.* it avoids all the other obstacles;
- *Feasible*: the trajectory respects the vehicle dynamics;
- *Non-conservative*: the trajectory should be efficient in terms of traffic flow.

The computation is refreshed every 100ms (10Hz) to take into account new information about the environment, in particular obstacles. The local MPC trajectory planner is based on a kinematic bicycle model of the vehicle where the lateral acceleration is constrained in order to ensure its validity at all time. We add Equation (4.11) to the model presented in section 2.4.3, where s is the

curvilinear abscissa traveled by the vehicle. The control inputs are the total acceleration u_1 of the vehicle and the steering rate u_2. These control inputs differ from the one used at the low-level control, namely the wheel torques and the steering angle, presented in section 4.3.4. In addition, we add some slack variables $obs_{tol}, x_{tol}, y_{tol}, \delta_{tol}$ to the control variables that will be defined below.

$$\dot{s} = \mathrm{V} \tag{4.11}$$

Finally, let $\mathrm{U}_t^k = [u_1, u_2, obs_{tol}, x_{tol}, y_{tol}, \delta_{tol}]$ and $\xi_t^k = [s, \mathrm{X}, \mathrm{Y}, \mathrm{V}, \psi, \delta]$ denote respectively the control input and the state of our MPC at time $t + k\Delta t_u$, predicted at time t, with $k = 0..\mathrm{N}_u - 1$.

Cost function

We set $e_v = (\mathrm{V}_k - \mathrm{V}_{heur_k})_{k\in[0:\mathrm{N}_y-1]}$ to be the speed cost; $e_\delta = (\delta_k)_{k\in[0:\mathrm{N}_y-1]}$ and $\dot{e}_\delta = (\dot{\delta}_j)_{j\in[0:\mathrm{N}_u-1]}$ respectively the cost on the steering angle and the steering angle variation; $e_{\mathrm{X}_{tol}} = (x_{tol_j})_{j\in[0:\mathrm{N}_u-1]}$ and $e_{\mathrm{Y}_{tol}} = (y_{tol_j})_{j\in[0:\mathrm{N}_u-1]}$ the cost of the slack variables associated respectively with the longitudinal and lateral offset to the reference trajectory; $e_{Obs_{tol}} = (obs_{tol_j})_{j\in[0:\mathrm{N}_u-1]}$ the cost of the slack variables associated with the obstacle constraints; $e_{\delta_{tol}} = (\delta_{tol_j})_{j\in[0:\mathrm{N}_u-1]}$ the cost of the slack variable associated with the validity of the kinematic bicycle model constraint.

We define the following cost function for a control sequence $\bar{\mathrm{U}}_t = [\mathrm{U}_t^0, \mathrm{U}_t^1, ..., \mathrm{U}_t^{\mathrm{N}_u-1}]$ and a state sequence $\bar{\xi}_t = [\xi_t^0, \xi_t^1, ..., \xi_t^{\mathrm{N}_y-1}]$:

$$\mathrm{J}(\bar{\mathrm{U}}_t, \bar{\xi}_t) = ||e_v||^2_{\mathrm{Q}_v} + ||e_\delta||^2_{\mathrm{Q}_\Delta} + ||\dot{e}_\delta||^2_{\mathrm{Q}_{\dot{\delta}}} + ||e_{\mathrm{X}_{tol}}||^2_{\mathrm{Q}_x} + ||e_{\mathrm{Y}_{tol}}||^2_{\mathrm{Q}_y} + ||e_{Obs_{tol}}||^2_{\mathrm{Q}_{obs}} + ||e_{\delta_{tol}}||^2_{\mathrm{Q}_{\delta_{tol}}} \tag{4.12}$$

where $||x||^2_{\mathrm{Q}} = x^{\mathrm{T}}\mathrm{Q}x$ and $\mathrm{Q}_v = q_v\mathrm{I}_{\mathrm{N}_y}$, $\mathrm{Q}_\Delta = q_\Delta\mathrm{I}_{\mathrm{N}_y}$, $\mathrm{Q}_{\dot{\delta}} = q_{\dot{\delta}}\mathrm{I}_{\mathrm{N}_u}$, $\mathrm{Q}_x = q_x\mathrm{I}_{\mathrm{N}_u}$, $\mathrm{Q}_y = q_y\mathrm{I}_{\mathrm{N}_u}$, $\mathrm{Q}_{obs} = q_{obs}\mathrm{I}_{\mathrm{N}_u}$, $\mathrm{Q}_{\delta_{tol}} = q_{\delta_{tol}}\mathrm{I}_{\mathrm{N}_u}$. For the simulations presented in section 4.3.5, the parameters displayed in Table 4.1 were chosen. In the rest of this chapter, $\mathrm{N}_y = \mathrm{N}_u$ and T_H denotes the prediction horizon of the MPC.

q_v	q_Δ	$q_{\dot{\delta}}$	q_x	q_y	q_{obs}	$q_{\delta_{tol}}$
4	10	0.2	5	5	100	100

Table 4.1 – Parameters of the cost function of the local MPC planner.

Constraints

First, we constraint the state ξ_t^0 to be equal to the initial state $\xi_0(t)$ of the vehicle at time t (Equation (4.15a)). Then, Equation (4.15b) describes that the trajectories should respect the kinematic bicycle model described in Equations (2.43) and (4.11). The actuators are limited respectively to $u_1 \in [a_{min}; a_{max}]$ and $u_2 \in [u_{2min}; u_{2max}]$ (Equations (4.15c) and (4.15d)). The road limits are encoded through a soft constraint: the longitudinal and lateral deviations from the reference path, respectively x_{err} and y_{err}, have to remain below a certain tolerance margin x_{tol} and y_{tol} (Equations (4.15e) and (4.15f)). The longitudinal and lateral deviations x_{err} and y_{err} are given by Equations (4.13a) and (4.13b) where $(\mathrm{X}_i, \mathrm{Y}_i)$ is the closest point on the reference path to the vehicle. Then, for each obstacle, we have an inequality constraint given by Equation (4.15g) which forbids the vehicle to enter the parabola p_o defined in section 4.3.2; h is a function of both the position (X, Y) of the vehicle and the parabola p_o defined around obstacle o. The vehicle is not located inside p_o if and only if $h(\mathrm{X}, \mathrm{Y}, p_o) \leq 0$.

$$x_{err} = (\mathrm{X} - \mathrm{X}_i)\cos(\psi_0) + (\mathrm{Y} - \mathrm{Y}_i)\sin(\psi_0) \tag{4.13a}$$
$$y_{err} = -(\mathrm{X} - \mathrm{X}_i)\sin(\psi_0) + (\mathrm{Y} - \mathrm{Y}_i)\cos(\psi_0) \tag{4.13b}$$

At last, to ensure the validity of the kinematic bicycle model, we constraint the lateral acceleration a_y to remain below $0.5\mu g$. More precisely, using Equation (4.14), we obtain the maximum

authorized steering angle for a given speed V (δ_{ub} corresponds to the maximal steering angle physically possible). Instead of constraining directly the steering angle $|\delta| \leq \delta_{max}$, a slack variable δ_{tol} is used (see Equation (4.15h)).

Remark 9. *For better computational efficiency, Equation (4.14) is computed for* $V = V_{init}$*, where* V_{init} *is the speed of the vehicle at the time of replanning: thus constraint (4.15h) becomes simple to implement (linear constraint) and also more robust. Indeed, choosing* $V = V_{init}$ *corresponds to the most critical case as if there is an obstacle, the vehicle will normally reduce its speed.*

$$\delta_{max}(V) = \min\left(\operatorname{atan}\left(\left(\frac{l_f}{l_r}+1\right)\tan\left(\operatorname{asin}\left(\frac{0.5\mu g l_r}{V^2}\right)\right)\right), \delta_{ub}\right) \tag{4.14}$$

$$\begin{aligned}
\xi_t^0 &= \xi_0(t) && && (4.15a)\\
\dot{\xi}_t^k &= f(\xi_t^k, U_t^k) && \text{for each } k && (4.15b)\\
a_{min} &\leq u_1^k \leq a_{max} && \text{for each } k && (4.15c)\\
u_{2min} &\leq u_2^k \leq u_{2max} && \text{for each } k && (4.15d)\\
|x_{err}^k| &\leq x_{tol}^k && \text{for each } k && (4.15e)\\
|y_{err}^k| &\leq y_{tol}^k && \text{for each } k && (4.15f)\\
h(X^k, Y^k, p_o) &\leq obs_{tol}^k && \text{for each } o && (4.15g)\\
-\delta_{max} + |\delta^k| &\leq \delta_{tol}^k && \text{for each } k && (4.15h)
\end{aligned}$$

All the values used in simulation are presented in Table 4.2.

a_{min}	a_{max}	u_{2min}	u_{2max}	δ_{lb}	δ_{ub}
$-8m/s$	$6m/s$	-0.5rad/s	0.5rad/s	$-0.6rad$	$0.6rad$

Table 4.2 – Parameters of the actuator constraints of the local MPC planner.

Numerical resolution

The numerical optimization was done using ACADO Toolkit. The working principle of the resolution algorithm can be found in [Houska et al., 2011a]. The general parameters of the MPC formulation are given in Table 4.3.

f	$10Hz$	frequency
T_H	$3s$	prediction horizon
Δt_u	$0.2s$	control discretization time step
N_u	$T_H/\Delta t_u + 1$	control prediction horizon
N_y	$T_H/\Delta t_u = 1$	state prediction horizon

Table 4.3 – Parameters of MPC.

4.3.4 Longitudinal and lateral low-level controllers

The low-level controllers are responsible for translating the trajectory computed by the local planner into actuator inputs. They will be studied in more details in chapters 5 and 6 of this thesis.

In the planning and control architecture presented in this section, we want to prove that the trajectory obtained with the local MPC planner produces feasible trajectories that is easy to track for low-level controllers. Therefore, we used very simple ones. In order to be robust to modeling errors, they are computed at high-frequency (100Hz). Longitudinal and lateral control problems are treated separately.

For the longitudinal controller, a simple PID controller [Astrom and Hagglund, 1995] which takes into account the error between the reference speed V^r given by the local MPC planner and the actual speed V of the vehicle, was implemented. More details will be given in chapter 5. V^r is more precisely a linear interpolation of the reference speed output given by the MPC planner. If $e = V - V^r$, then the control law is given by Equation (4.16), where K_{P_v}, K_{D_v} and K_{I_v} are the gains of the controller.

$$u(t) \quad = \quad -K_{P_v} e(t) - K_{D_v} \dot{e}(t) - K_{I_v} \int_0^t e(\tau) d\tau \tag{4.16}$$

More precisely, the braking gains are different from the acceleration ones (multiplication by a different factor λ_f). This control value is then dispatched equally on each of the four wheels of the vehicle in case of braking ($u < 0$), and each of the two front wheels in case of acceleration ($u \geq 0$).

For the lateral controller, the steering angle applied δ is composed of an open-loop part δ_{ff} (for feedforward) and a closed-loop part δ_{fb} (for feedback). The open-loop steering angle δ_{ff} is computed by integrating the first value of the control obtained by the MPC planner, which corresponds to the steering rate. This value is only refreshed once every 100ms. The closed-loop steering angle δ_{fb} is a simple PID controller applied on the yaw angle error, projected one MPC time step ahead. It is computed every 10ms and uses as reference value for the yaw angle a linear interpolation of the one computed by the MPC planner. The gains are denoted K_{P_ψ}, K_{I_ψ} and K_{D_ψ}.

All the gains used for the longitudinal and lateral PID controllers are given in Table 4.4

K_{P_v}	K_{I_v}	K_{D_v}	$\lambda_f\,(u \geq 0)$	$\lambda_f\,(u < 0)$	K_{P_ψ}	K_{I_ψ}	K_{D_ψ}
$0.5\lambda_f$	$0.01\lambda_f$	$0.005\lambda_f$	1500	5000	2	0.005	0.01

Table 4.4 – Parameters of the longitudinal and lateral PID controllers.

4.3.5 Simulation results

In order to test the proposed planning and control architecture, simulations were performed in PreScan [TASS International,] using the 10 DoF vehicle model presented in chapter 2.3. The test track was designed to alternate between long straight lines and small radius curvatures (up to 10m only) as shown in Figure 4.12. This is particularly challenging as the constraint on lateral acceleration is rapidly reached in the curves if the vehicle arrives too fast. Therefore, some precautions must be taken in the choice of the parameters of the planning and control architecture. For example, the prediction horizon of the MPC depends on the maximum speed V_{max} as the vehicle needs to be able to decelerate sufficiently in advance to reach the target speed in the curves. The road friction coefficient μ was chosen equal to 1. The initial speed is $V_{x_0} = 15$m/s.

Simulation without obstacles

First, we tested our planning and control architecture on the reference track without obstacles in order to show its capacity to adapt the speed accordingly to the situation. The results are shown in Figure 4.13 where the green lines represent the trajectory planned by the local MPC planner and the blue dots represent the actual trajectory followed by the vehicle. Figure 4.14 compares the heuristic speed V_{heur} computed by the local velocity planner (section 4.3.1), the target speed V^r computed by the MPC planner (section 4.3.3) and the actual speed V of the vehicle (sesction 4.3.4). We observe that before entering a curve, the heuristic speed computed by the local velocity planner reduces severely in order not to exceed $0.5g$ for the lateral acceleration. As there are no obstacles, the speed V^r of the MPC planner is guided by this heuristic, while the longitudinal low-level

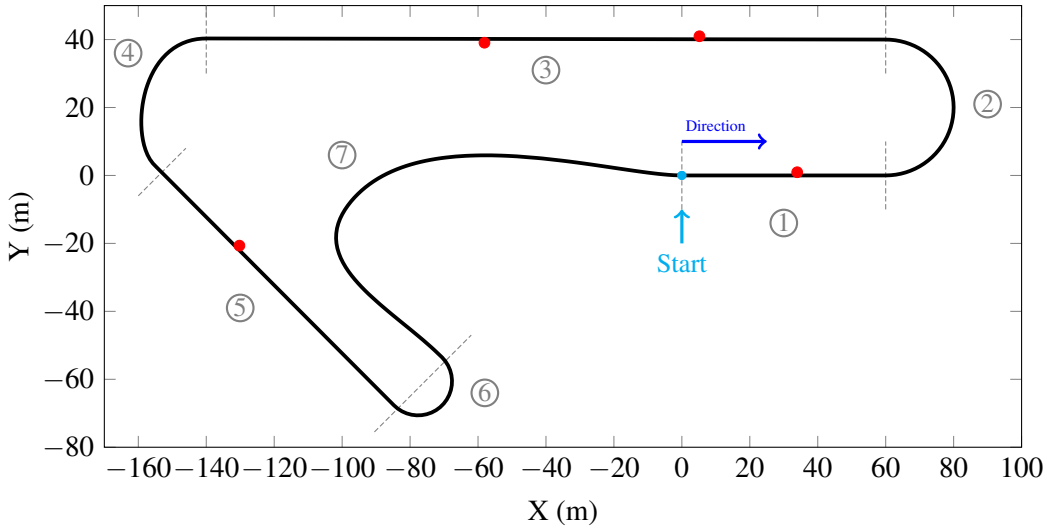


Figure 4.12 – Reference track. The obstacles are in red. The numbers indicate different road sections to facilitate the matching with Figures 4.14 and 4.17.

controller tracks sufficiently well this reference speed in curves to avoid skipping and skidding as shown in Figures 4.15 and 4.16. The low-level control inputs are shown in Figure 4.17a and 4.18 (in blue). The absolute value of the lateral error to the reference path given at entry of our system does not exceed 0.4m as shown in Figure 4.19; the computational time of the MPC remains always below 100ms as shown in Figure 4.20. Moreover, we observe in Figure 4.17a that δ_{fb} is important due to the slow refresh time of the MPC: using only δ_{ff}, the architecture does not work as the controller does not follow the reference trajectory planned by the local MPC planner.

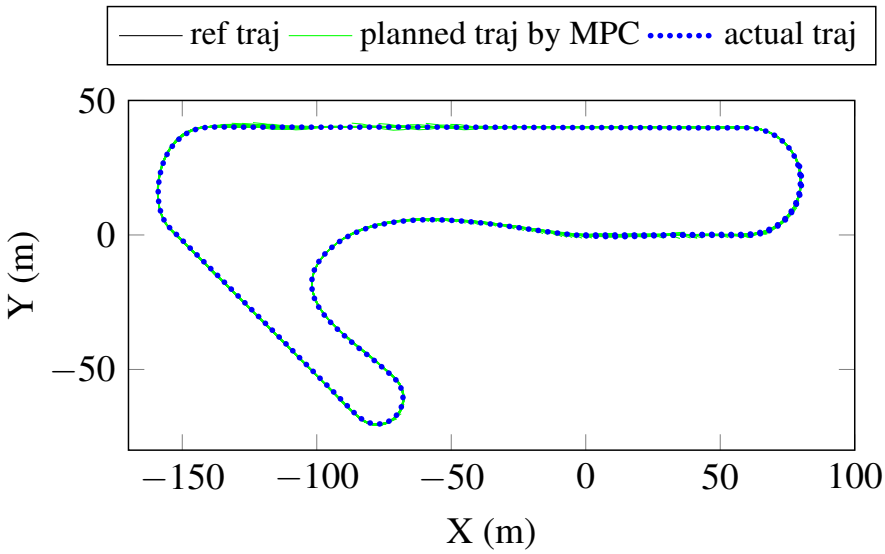


Figure 4.13 – Trajectories planned by MPC (green lines) and actual trajectory followed by the controller (blue dots), with obstacles (red dots).

Simulation with obstacles

In a second step, we tested our planning and control architecture on the reference track with static obstacles in order to show the capacity of the motion planner to still plan safe and feasible reference trajectories. The trajectory obtained is shown in Figure 4.21. The low-level control inputs are shown in Figure 4.17b and 4.18 (in red). The different speeds are shown in Figure 4.22: this time V^r does not follow V_{heur} when there is an obstacke. As expected, the vehicle does almost not slip or

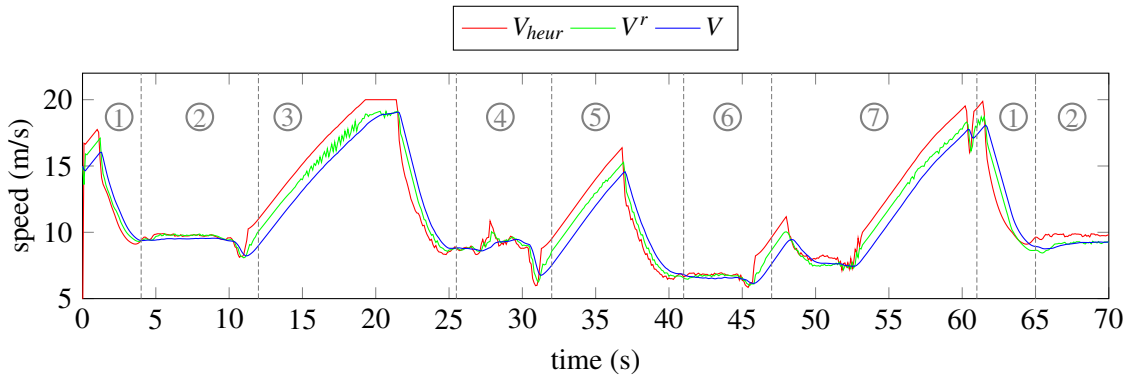


Figure 4.14 – Comparison between V_{heur} (red), V^r (green) and V (blue), in the case of no obstacles.

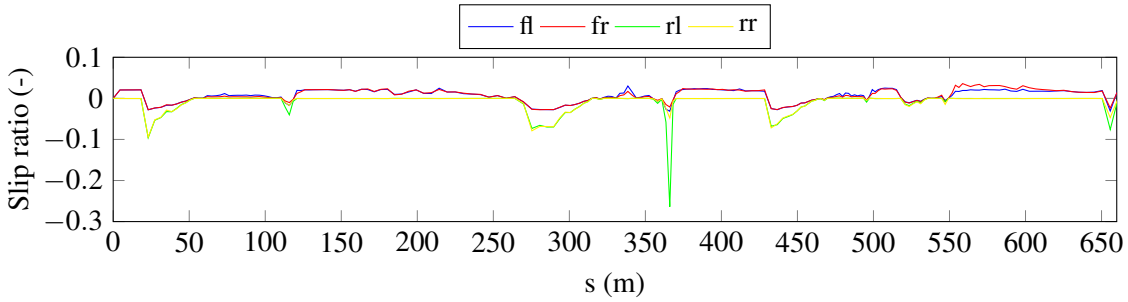


Figure 4.15 – Longitudinal slip ratio at each wheel in the case of no obstacles.

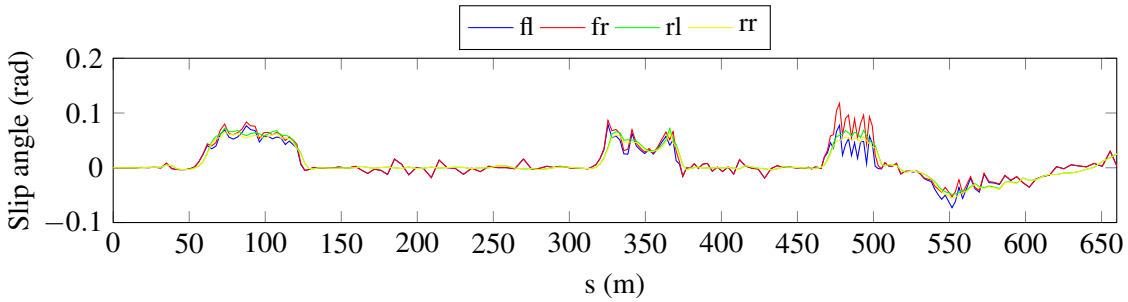


Figure 4.16 – Lateral slip angle at each wheel in the case of no obstacles.

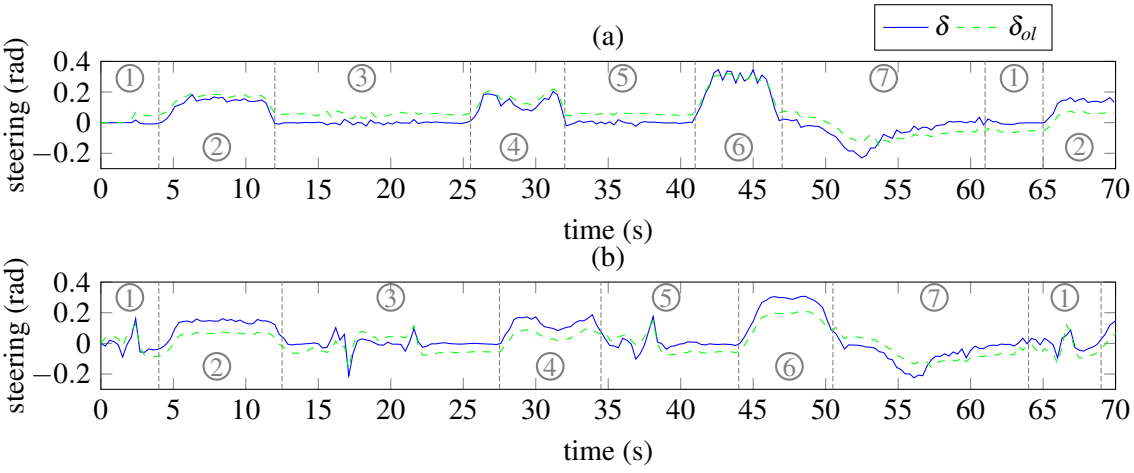


Figure 4.17 – Total steering angle δ (blue) and closed-loop steering angle δ_{fb} (green): (a) without obstacles; (b) with obstacles.

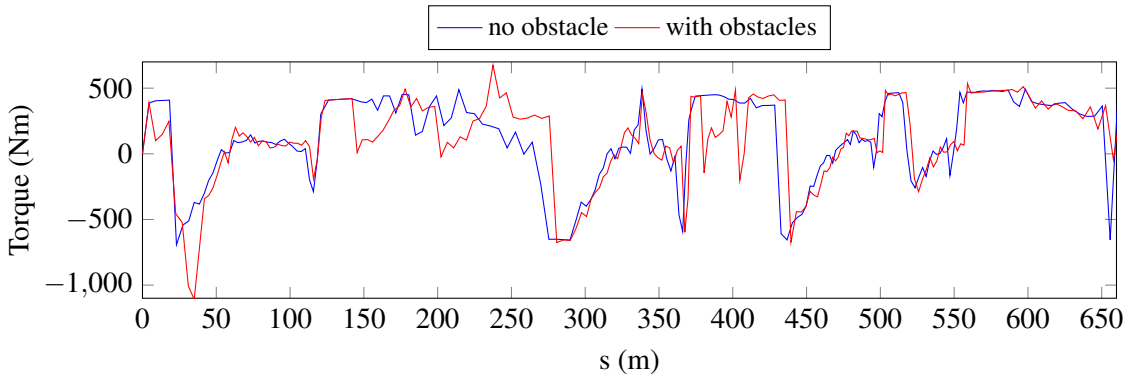


Figure 4.18 – Wheel torque applied on the front left wheel: without obstacles (blue); with obstacles (red).

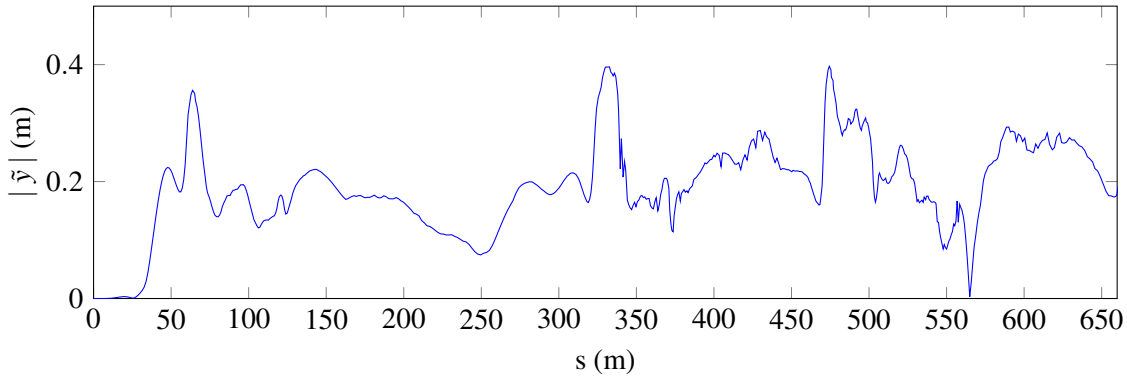


Figure 4.19 – Absolute value of the lateral error in the case of no obstacles.

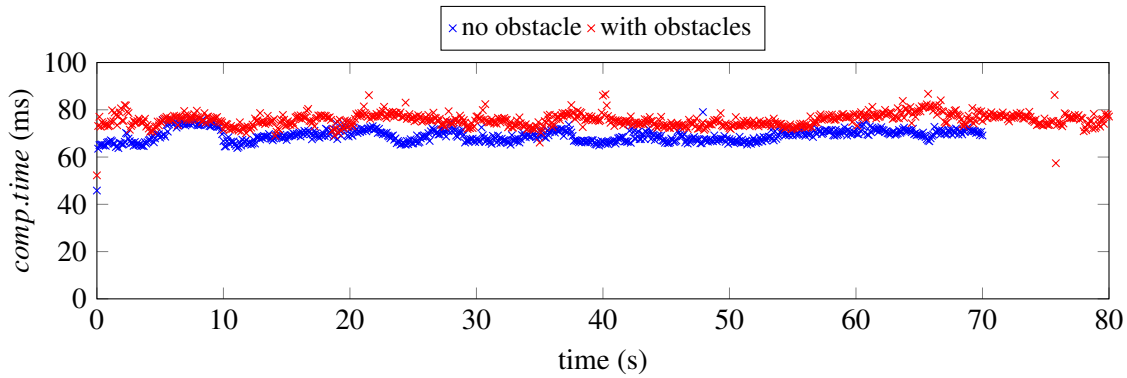


Figure 4.20 – Computational time in the case of no obstacle (blue) and with obstacles (red).

skid thanks to the constraint on lateral acceleration (see Figures 4.23 and 4.24) except at the first obstacle ($s \approx 40$m) due to the initial speed V_{x_0} which was too high. The computational time of the MPC remains below 100ms (see Figure 4.20).

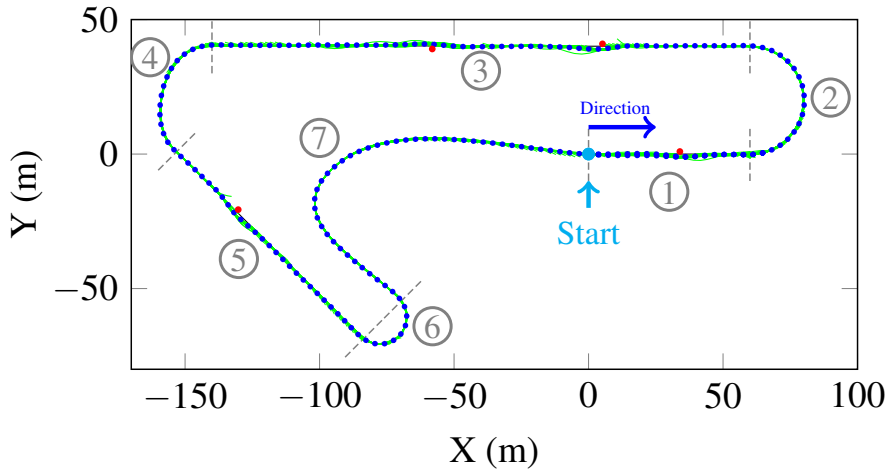


Figure 4.21 – Trajectories planned by MPC (green lines) and actual trajectory followed by the controller (blue dots), with obstacles (red dots). The numbers indicate different road sections to facilitate the matching with Figures 4.17 and 4.22.

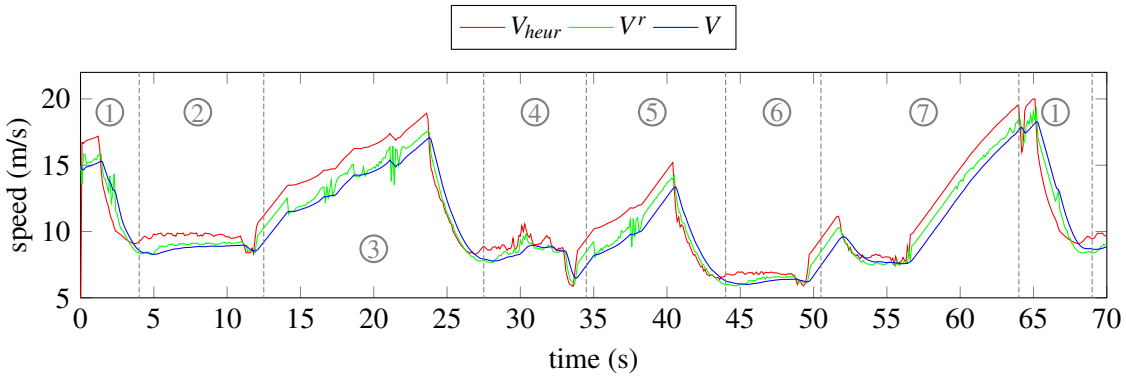


Figure 4.22 – Comparison between V_{heur} (red), V^r (green) and V (blue), in the case of obstacles.

4.3.6 Conclusion

Our planning and control architecture for autonomous vehicle enables under normal driving conditions to guarantee the consistency between the motion planning and the control layers by ensuring the validity of the kinematic bicycle model used for planning at any time. This guarantee is obtained using a dynamic constraint on the maximal authorized steering angle. Therefore, only feasible trajectories will be generated as the lateral acceleration remains below $0.5g$. This is done at almost no additional computational cost compared to motion planners already using the kinematic bicycle model. The low-level controller enables the architecture to be robust to disturbances and modeling errors. Moreover, this architecture is also less conservative as it is guided toward the highest speed that keeps the architecture safe thanks to the computation of a heuristic speed based only on the road geometry.

The proposed architecture is able to cope with most normal driving situations and also guarantees that lateral controllers based on the kinematic bicycle model are able to follow any of the output reference trajectories. However, in some situations such as emergency maneuvers, the constraint on lateral acceleration might be too harsh. This require to switch to a more complex model at the planning and control level where dynamic effects are taken into account.

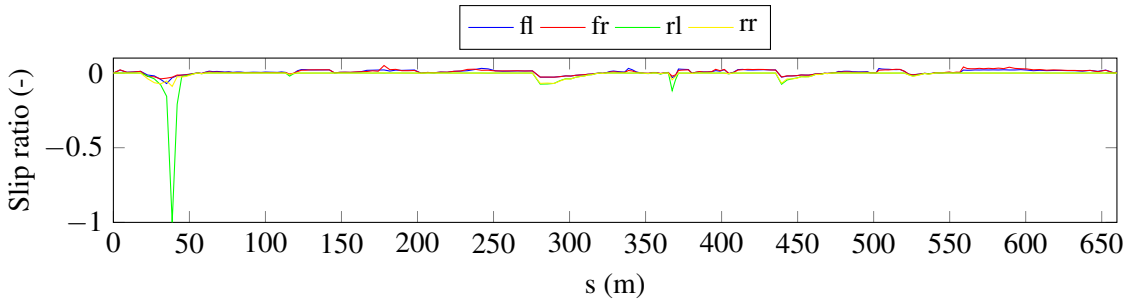


Figure 4.23 – Longitudinal slip ratio at each wheel in the case of obstacles.

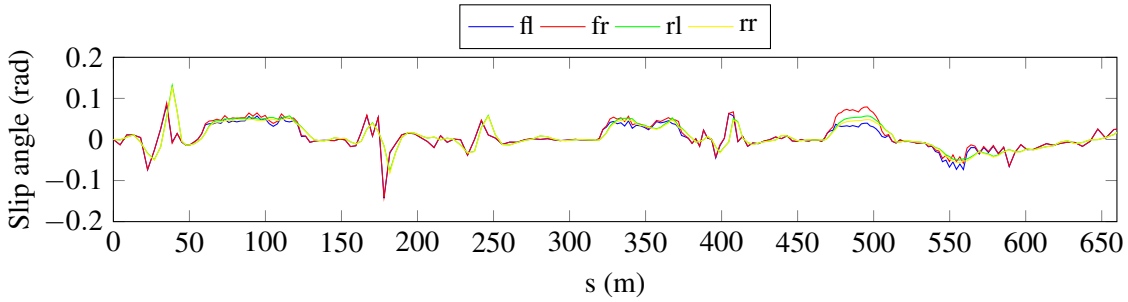


Figure 4.24 – Lateral slip angle at each wheel in the case of obstacles.

In the next section, we are going to adapt this architecture to low friction coefficient roads such as wet and snow-covered roads.

4.4 Adaptation to low friction coefficient roads

4.4.1 Adaptation of the planning and control architecture

Several constraints have to be added at the motion planning level to ensure that the low-slip and low-skid assumptions remain valid, in particular on slippery roads.

Dynamic maximum steering angle

In section 4.2, we showed that the kinematic bicycle model remained valid on a road with friction coefficient μ if the lateral acceleration of the vehicle remained below $0.5\mu g$ where g is the gravitational constant. Moreover the modeling errors become large if this constraint is not respected. The constraint on the lateral acceleration can be translated to a constraint on the maximum steering angle δ_{max} depending on the speed and given by Equation (4.14), with $|\delta| \leq \delta_{max}$. This constraint is linear by taking into account Remark 9.

Limitation of the total acceleration

The acceleration u_1 that can be reached by the vehicle depends strongly on the road friction coefficient μ. Therefore, the maximum acceleration a_{max} and deceleration a_{min} authorized have to be modeled as functions of μ.

The longitudinal and lateral tire forces F_{xp} and F_{yp} expressed in the pneumatic frame can be expressed as following, using an approximation of the Pacejka tire model [Pacejka, 2002]:

$$F_{xp} \approx \mu F_{xp_0} \quad ; \quad F_{yp} \approx \mu F_{yp_0} \tag{4.17}$$

where

$$\begin{aligned} F_{xp_0} &= F_z \sin\left(C_x \operatorname{atan}(B_x\tau_x - E_x(B_x\tau_x - \operatorname{atan}(B_x\tau_x)))\right) \\ F_{yp_0} &= F_z \sin\left(C_y \operatorname{atan}(B_y\alpha - E_x(B_y\alpha - \operatorname{atan}(B_y\alpha)))\right) \end{aligned}$$

τ_x and α are respectively the longitudinal slip ratio and the lateral slip angle of the wheel, F_z the normal reaction force, and B_x, C_x, E_x, B_y, C_y and E_y tire parameters of the Pacejka model.

The frame change between the pneumatic frame and the vehicle frame is given Equation (4.18) for the front wheels.

$$F_x = F_{xp}\cos\delta_f - F_{yp}\sin\delta_f = \mu F_{xi,0} \tag{4.18a}$$
$$F_y = F_{yp}\cos\delta_f + F_{xi}\sin\delta_f = \mu F_{yi,0} \tag{4.18b}$$

where

$$F_{xi,0} = F_{xp_0}\cos\delta_f - F_{yp_0}\sin\delta_f$$
$$F_{yi,0} = F_{yp_0}\cos\delta_f + F_{xp_0}\sin\delta_f$$

The dynamic equations for the vehicle when neglecting the aerodynamic drag forces are:

$$M_T a_x = \sum_{i=1}^{4} F_{x_i} \quad ; \quad M_T a_y = \sum_{i=1}^{4} F_{y_i} \tag{4.19}$$

Therefore, if we set $a_{max}(\mu = 1)$ and $a_{min}(\mu = 1)$ to be respectively the maximal acceleration and deceleration for $\mu = 1$, we obtain:

$$a_{max}(\mu) = \mu a_{max}(\mu = 1) \quad ; \quad a_{min}(\mu) = \mu a_{min}(\mu = 1) \tag{4.20}$$

In our MPC, we set $a_{max}(\mu = 1) = +6\text{m/s}^2$ and $a_{min}(\mu = 1) = -8\text{m/s}^2$. Thus, the total acceleration u_1 of the vehicle has to respect the following linear constraint:

$$\mu a_{min}(\mu = 1) \leq u_1 \leq \mu a_{max}(\mu = 1) \tag{4.21}$$

Moreover, in order to avoid unnecessary variations of the speed profile, a small cost penalty on the acceleration, $u_1^T q_{ac} u_1$ with $q_{ac} = 1$, was added to the MPC cost function (see section 4.3.3).

Adaptation of the heuristic speed

Finally, the local velocity planner presented in section 4.3.1 also needs to be adapted. First, it is important to take into account the road friction coefficient μ in Equation (4.10). Secondly, as the time horizon T_H of the local MPC planner is constant, it is necessary for safety reasons that $T_H \geq T_{stop}$, where T_{stop} is the time necessary for the vehicle to come to a full stop; as T_{stop} depends on the actual speed of the vehicle V and on the friction coefficient μ, the speed limit V_{max} should be chosen such that if the vehicle runs at that speed, $T_H = T_{stop}$. Thus, V_{max} depends both on T_H and on μ. Letting d_{br} be the braking distance, we have in that case:

$$d_{br} = -\frac{V_{max}^2}{2\mu a_{min}} \tag{4.22a}$$
$$d_{br} = V_{max} T_H - \frac{\mu a_{min} T_H^2}{2} \tag{4.22b}$$

Thus, the expression of the maximal speed is:

$$V_{max} = -\mu T_H a_{min} \tag{4.23}$$

Therefore, the choice of T_H is a trade-off between the computational time and the maximum speed V_{max} allowed; however, in practice, T_H is limited by the perception range. Constraint (4.23) is very important in the case of low μ as the braking distance is much larger than for situations with a road coefficient close to 1.

Remark 10. *As the vehicle does not need to come to a full stop in our simulations, we were able to relax a little Equation (4.23). We set* $V_{max} = 1.2 - \mu T_H a_{min}$.

Moreover, it is important that $T_{prev} \geq T_H$ where T_{prev} is the preview horizon of the local velocity planner (see section 4.3.1), in order to take into account the coming maximum curvature γ_{max} of the reference path during the MPC planning horizon. Choosing $T_{prev} > T_H$ enables to better anticipate the shape of the reference path; in particular, unlike T_H, T_{prev} is not limited by the perception range of the sensors, as it only depends on the road profile of the ego-lane which can be stored in High-Definition maps.

4.4.2 Simulation results

We used the same reference track as previously, alternating between small radius curves and long straight lines. Again, the goal of the motion planning and control architecture is to remain on the center lane except if there is an obstacle on the path. Two different road conditions were tested: wet roads ($\mu = 0.7$) and snow-covered roads ($\mu = 0.2$). The value of the road friction coefficient is assumed to be known and remain constant during the whole simulation.

Rain conditions ($\mu = 0.7$)

First, we tested our proposed planning and control architecture on wet roads[3], which corresponds to $\mu = 0.7$ [Persson et al., 2005]. We observe that the architecture is able to avoid the obstacles while following the reference path. The different speeds mentioned earlier are shown in Figure 4.25a: we observe that on the one hand, the heuristic speed V_{heur} (red) computed by the local velocity planner guides the reference speed V^r set by the MPC (green), and that on the other hand, the use of a longitudinal PID low-level controller is satisfying as the actual speed of the vehicle V (blue) is close to its reference (green). As expected, the vehicle is accelerating on the straight portions and decelerates as soon as the local velocity planner detects a coming curve. The speed in the curve depends on its curvature radius. Simulations were performed with $V_{max} = 18$m/s ($T_H = 3$s). Note that the minimum curvature radius R_{min} in Equation (4.10) is chosen among a discrete number of computed curvature radius R, which explains the small rapid increase in V_{heur} in Figure 4.25. The steering angle input is shown on Figure 4.26a. It is rather smooth, showing that our controller does not compensate for modeling errors: this shows the consistency of our architecture. The slip ratios and slip angles remain low as shown in Figure 4.27, even though the reference track is challenging: the slip ratio remains bounded between $[-6\%; +5\%]$, except the rear-left wheel (yellow) which reaches -10%. However, these values still corresponds to the linear part of the tire dynamics. Similarly, the slip angle remains in the bounds $[-0.1; +0.1]$rad which also corresponds to the linear part of the tire dynamics, except for the front-right wheel (red) when avoiding the second obstacle in road section n°3 and in the curve corresponding to road section n°1.

Snow conditions ($\mu = 0.2$)

The architecture was then tested on a snow-covered road[4], which corresponds to a friction coefficient of $\mu = 0.2$ [Waluś and Olszewski, 2011]. Again, we observe that the vehicle is able to follow the reference path while avoiding the red obstacles. On Figure 4.25b, we observe that the heuristic speed (red) computed by the local velocity planner limits the speed of the vehicle to ensure its safety. Simulations were performed choosing $V_{max} = 6$m/s ($T_H = 3$s). As expected, the vehicle is much slower on snow: it takes about 138s to perform a full round of the track on icy road while it only took 75s on wet roads. In particular, the speed in the curves is lower than in the previous case: this is due to the lateral acceleration condition in the local velocity planner which depends on μ. The slip ratio is displayed in Figure 4.28: it remains bounded for all wheels between $[-1.5\%; +1.5\%]$, which is small, especially for driving in the snow. The slip angles remain also very low (less than 0.06rad for all wheels), except for the two front wheels when avoiding obstacles were

[3] video available at https://youtu.be/6LFNhpcmssY
[4] video available at https://youtu.be/qUT5sFY_RE4

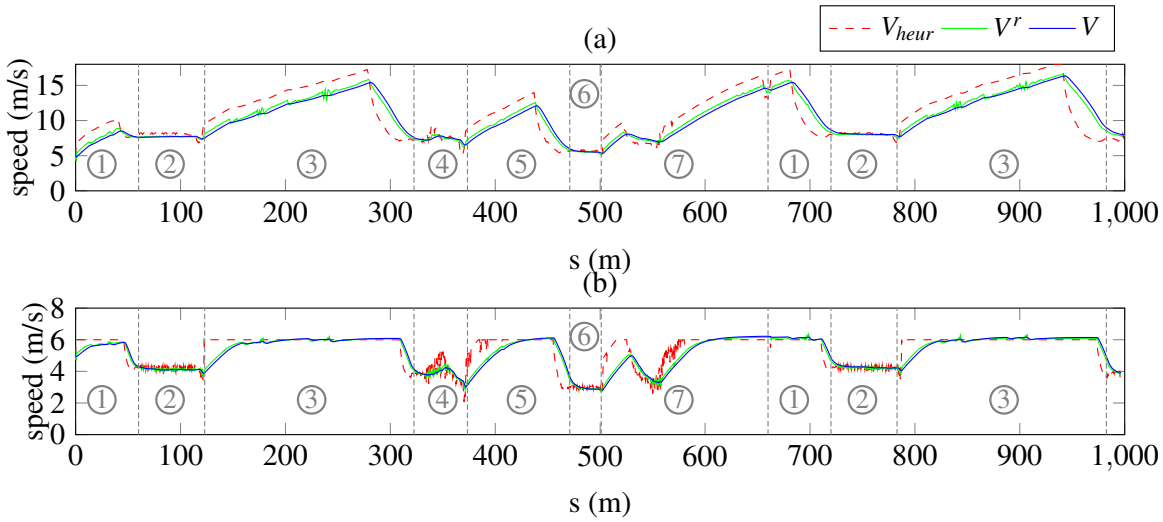


Figure 4.25 – Heuristic speed (red), reference speed computed by the MPC (green) and actual speed of the vehicle (blue) for (a) wet roads ($\mu = 0.7$) and (b) icy roads ($\mu = 0.2$).

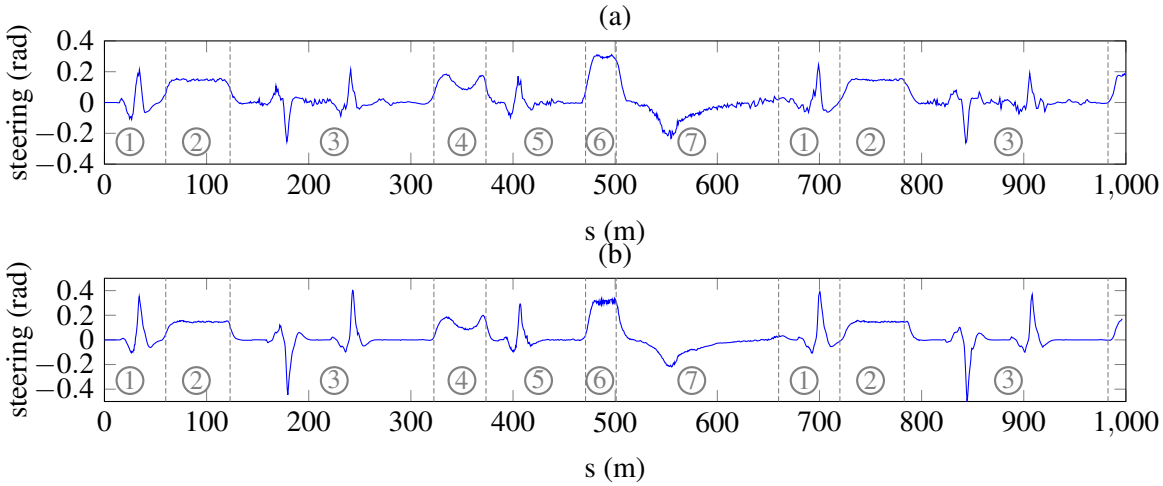


Figure 4.26 – Steering angle inputs: (a) on wet road ($\mu = 0.7$) and (b) on icy road ($\mu = 0.2$).

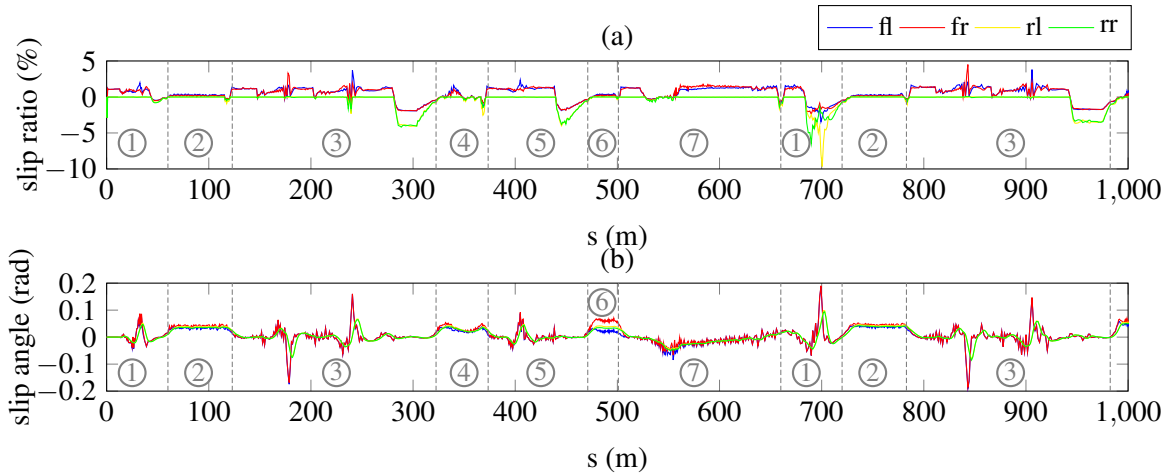


Figure 4.27 – Slip ratios (a) and slip angles (b) at each wheel on wet road (μ = 0.7): front-left (blue), front-right (red), rear-left (yellow) and rear-right (green).

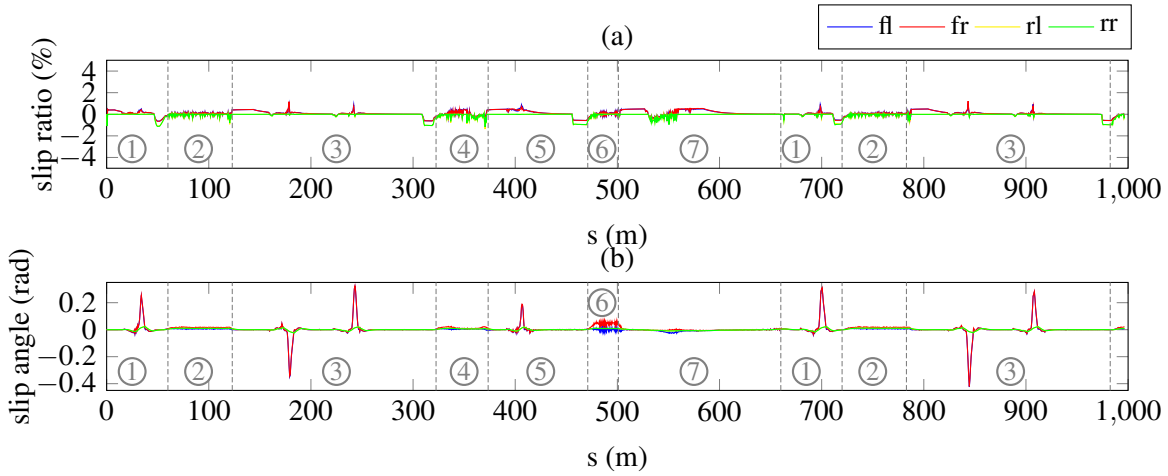


Figure 4.28 – Slip ratios (a) and slip angles (b) at each wheel on icy road (μ = 0.2): front-left (blue), front-right (red), rear-left (yellow) and rear-right (green).

they can reach up to ±0.3rad. The steering angle input is displayed in Figure 4.26b: again, we observe that the control is smooth. We notice that the steering angle in the curves is almost the same on the wet and snow-covered roads: this illustrates the validity of the kinematic bicycle model as we mentioned in Equation (4.3) that the steering angle depends only on the curvature radius of the path, not the speed or the friction coefficient.

Note that removing one of the three conditions described in section 4.4.1 leads to a failure of the architecture, usually during the sharp curves. Thus, these conditions that were derived to obtain a consistent motion planning and control architecture are actually necessary in order to work on slippery roads.

The tuning of the PID parameters and the weights of the MPC cost function remained the same as in the dry road simulations (μ = 1) showing the consistency of our planning and control architecture: on the one hand, by ensuring the validity of the model used for motion planning in all situations, there is no need to adapt the controller anymore; on the other hand, the introduction of the local velocity planner depending on the road characteristics enables to keep the same weight for the speed cost in the local MPC planner. Thus, implementation on a vehicle is easen.

4.5 Conclusion

The kinematic bicycle model appears thus to be a good compromise between model accuracy and computational efficiency for planning safe trajectories. However, it is important to guarantee its validity at all time by limiting the lateral acceleration of the vehicle, or equivalently, by limiting the steering angle by a function of speed. The more knowledge we set at the motion planning level, the easier it is to ensure the safety of the vehicle, in particular as the controllers are blind to obstacles. Such a motion planner based on the kinematic bicycle model should be used in a hierarchical motion planning and control architecture as bigger time steps can be chosen in the MPC formulation, thus improving the efficiency and increasing the prediction horizon. Moreover, separating the motion planning from the controllers improves the robustness to disturbances as the low-level controllers can act at high-frequency. However, the control sequence computed by the motion planner can still be used as a feedforward term for the low-level controller.

One of the main assets of the architecture proposed in this chapter is that there is no need to adapt the gains or weight to the situation, neither for the motion planner, nor for the controllers. Moreover, this architecture turns out to be not too conservative while ensuring the safety of the vehicle at all time.

However, several limits to our approach can be found. First, we assumed that the friction coefficient μ of the road is known in advance and does not vary. This is a very restrictive assumption and some future works should focus on how to adapt this architecture in real-time for varying μ. Moreover, it has not been tested on an actual vehicle so far. Finally, in the motion planner MPC formulation, the kinematic bicycle model is a non-linear constraint and the search space is not convex. Therefore, the resolution might end up in some local minima which could violate some safety constraints. One way to reduce this problem is to solve the MPC formulation on subproblems corresponding to each *homotopy class* associated to a driving decision (ex: overtake right, overtake left, stay behind,...) in order not to hit the obstacles [Bender et al., 2015]. The homotopy class was implicitly given here by the parabola as the vehicle can only overtake on one side of the obstacle. This method is well suited for a single lane country road as it is only possible to overtake a vehicle on one side. Reinforcement learning or supervised deep learning approaches could also be used to choose the best class of homotopy and provide an initial guess to the MPC solution.

Ensuring model validity can be extended to other models and motion planning techniques: for example for RRT using Dubins'path, the maximal curvature should be chosen as a function of the speed to guarantee that the lateral acceleration remains below $0.5\mu g$. Similarly, it is possible to use a unicycle model, which is a kinematic bicycle model expressed at the center of the rear axle, and ensure its validity. However, as the control input is no longer the steering angle δ but the yaw rate, the steering angle constraint should be adapted as done in Equation (4.24) to ensure the dynamic feasibility of the reference trajectory obtained. Moreover, this provides a reference trajectory for the rear-wheel which is often used for low-level controllers as we will see in chapter 5.

$$|\dot{\psi}| \leq \frac{0.5\mu g}{V} \tag{4.24}$$

Moreover, it would be interesting to investigate the application of machine learning techniques for learning the vehicle dynamics in order to generate feasible and safe trajectories. Some preliminary work have been done in [Devineau et al., 2018] for coupled longitudinal and lateral controllers but can be extended to motion planners.

Part III

Low-level Controllers

Chapter 5

Controllers for autonomous vehicles

" Si hortum in bibliotheca habes, nihil deerit[a] "

[a] If you have a garden and a library, you have everything you need
Cicero

Contents

In the previous chapter, we have presented how to compute a dynamically feasible reference trajectory for the vehicle. Such a trajectory corresponds to the intention of the vehicle, *i.e.* the plan it intends to follow in the next few seconds. However, in order to ensure the safety of the vehicle, this trajectory must be executed as accurately as possible. This is the role of low-level controllers which translate the reference trajectory into actions to perform on the actuators, namely the gas pedal, the brake pedal and the steering wheel. The more robust and more accurate the controllers are, the better the action realized by the vehicle matches its intention. In fact, it is highly critical regarding the safety of the vehicle as at the low-level control, obstacles are usually not considered anymore due to the level of abstraction.

In the literature, controllers are often split between *longitudinal controllers* which track a given reference speed or acceleration profile, and *lateral controllers* which track a given reference path $(X_i, Y_i)_{i\in[\![1,n]\!]}$. This decoupling of the longitudinal and lateral dynamics is in most situations a fair assumption: usually, drivers either accelerate/decelerate the vehicle or turn the steering wheel, but rarely do both at the same time (or at least, one of the actions is performed with parcimony). However, in some cases such as aggressive driving or emergency scenarios for obstacle avoidance, the coupling between longitudinal and lateral dynamics can not be neglected anymore. One of the main reason is that the tire forces reach the limits of the friction circle described in section 2.2. Therefore, some *coupled controllers* have been introduced in the literature that deals with the longitudinal and lateral dynamics at the same time.

Thus, the aim of this chapter is to make a (non-exhaustive) review of commonly used longitudinal and lateral controllers, as well as coupled ones. A special focus is made on lateral control, which is a harder problem than longitudinal control. In particular, we want to be able to predict if a given lateral controller will be able to track a given reference trajectory. Therefore, a comparison between controllers is made on an actual vehicle both in terms of accuracy, smoothness and requirements on the reference trajectory.

The rest of this chapter is organized as following: section 5.1 presents shortly the principle of longitudinal control and provides two examples. Section 5.2 deals with commonly used lateral controllers and their operational limits. In section 5.3, some of the lateral controllers are implemented on an actual vehicle and their performances compared. Finally, section 5.4 presents some controllers dealing simultaneously with the longitudinal and lateral dynamics. Note that more references on longitudinal and lateral controllers can be found in the literature, see for example [Khodayari et al., 2010] which presents a brief historical review of those developed from 1990 to 2010.

5.1 Longitudinal control

Two different levels of control can be considered for the longitudinal dynamics: a high-level controller that defines a desired reference speed or acceleration depending on the other traffic participants and/or the road geometry, and a low-level controller that tracks a given reference speed by computing the torques required at the wheels. Examples of high-level controllers comprise Adaptive Cruise Control (ACC) and Stop-and-Go algorithms which can be found in [Swaroop et al., 2001, Martinez and Canudas-de Wit, 2007, Liang and Peng, 1999]. As we assume that the reference speed will be set by the motion planner, only the low-level controllers are considered in this thesis.

Several techniques exist in the literature for low-level longitudinal control. We will give here two examples: a controller based on a dynamic model of the vehicle and a conventional PI controller.

5.1.1 Example 1: longitudinal controller based on a dynamic model

Let us recall from section 2.1 and 2.2 the equations of the longitudinal vehicle dynamics and of the wheel dynamics, where x is the longitudinal displacement of the vehicle:

$$\mathrm{M_T}\ddot{x} = \sum_{i=1}^{4} \mathrm{F}_{xi} - \mathrm{F}_{aero} \tag{5.1a}$$

$$\mathrm{I}_r\dot{\omega}_i = \mathrm{T}_{mi} - \mathrm{T}_{bi} - r_{eff}\mathrm{F}_{xpi} \tag{5.1b}$$

Let's define $\mathrm{T}_{\omega i}$ as the total torque applied on wheel i, *i.e.* the difference between the motor and the brake torques: $\mathrm{T}_{\omega i} = \mathrm{T}_{mi} - \mathrm{T}_{bi}$. Set $\mathrm{T}_{tot} = \sum_{i=1}^{4} \mathrm{T}_{\omega i}$ to be the total wheel torque applied on the vehicle and $\ddot{x}_{ref}$ the reference acceleration. Assuming that the front steering angle δ_f is small, we get $\mathrm{F}_{xpi} \approx \mathrm{F}_{xi}$. Thus:

$$\mathrm{M_T}\ddot{x}_{ref} + \mathrm{F}_{aero} = \frac{1}{r_{eff}} \sum_{i=1}^{4} (\mathrm{T}_{\omega i} - \mathrm{I}_r\dot{\omega}_i) \tag{5.1c}$$

Therefore, the open-loop total torque to apply on the wheels becomes:

$$\mathrm{T}_{tot} = \mathrm{M_T} r_{eff}\ddot{x}_{ref} + \mathrm{I}_r \sum_{i=1}^{4} \dot{\omega}_i + r_{eff}\mathrm{F}_{aero} \tag{5.1d}$$

Non-linear Lyapunov-based controllers can then be derived from there as in [Attia et al., 2012b] in other to ensure the stability. More precisely, let's define $\mathrm{V} : e \rightarrow \frac{1}{2}e^2$ where $e = \dot{x}_{ref} - \dot{x}$ is the speed tracking error.

If $\dot{\mathrm{V}} \leq -k\mathrm{V}$, V is a Lyapunov function:

- $\forall e, \mathrm{V}(e) \geq 0$ and $\mathrm{V}(e) = 0 \Rightarrow e = 0$
- $\dot{\mathrm{V}} \leq -k\mathrm{V} \leq 0$, thus $\forall e \in \mathbf{R} \setminus \{0\}$, $\dot{\mathrm{V}}(e) < 0$
- $\mathrm{V}(e) \underset{e \rightarrow +\infty}{\longrightarrow} +\infty$

Thus, the system is asymptotically stable and converges to zero.

In order to get $\dot{\mathrm{V}} \leq -k\mathrm{V}$, set:

$$\mathrm{T}_{tot} = k\mathrm{M_T} r_{eff} e + \mathrm{M_T} r_{eff}\ddot{x}_{ref} + \mathrm{I}_r \sum_{i=1}^{4} \omega_i + r_{eff}\mathrm{F}_{aero} \tag{5.2a}$$

as

$$\dot{\mathrm{V}} = e\left(\ddot{x}_{ref} - \frac{1}{\mathrm{M_T}}\left(\frac{1}{r_{eff}}(\mathrm{T}_{tot} - \mathrm{I}_r \sum_{i=1}^{4} \dot{\omega}_i) - \mathrm{F}_{aero}\right)\right) \tag{5.2b}$$

$$= -ke^2 \tag{5.2c}$$

$$\leq \frac{-ke^2}{2} = -k\mathrm{V} \tag{5.2d}$$

5.1.2 Example 2: longitudinal PI controller

A longitudinal PI controller consists in applying a total wheel torque T_{tot} that has one term proportional to the speed tracking error $e_v = v - v_r$ and another term proportional to the integral of the speed tracking error $\int_0^t e_v(\tau)\mathrm{d}\tau$ as illustrated by Equation (5.3). While the proportional term corrects the speed tracking error, the integral term enables to reject external perturbations.

$$\mathrm{T}_{tot}(t) = -\mathrm{K_P} e_v(t) - \mathrm{K_I} \int_{t_0}^{t} e_v(\tau)\mathrm{d}\tau \tag{5.3}$$

A PI controller was implemented for example on Stanley, the vehicle that won the Grand Darpa Challenge in 2005 [Thrun et al., 2006].

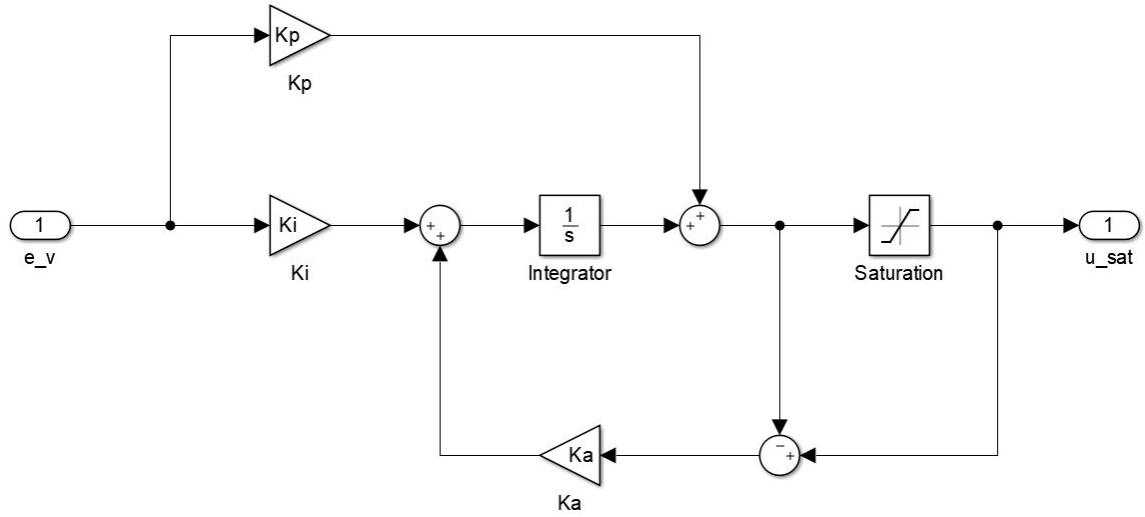


Figure 5.1 – PI controller with anti-windup.

Remark 11. *It is important to implement an anti-windup as shown on Figure 5.1 on the integral term to avoid that the integral error blows up during saturation of the actuator.*

The effectiveness of a PI controller can be illustrated as following. Suppose that the angular velocity of each wheel is approximately constant, *i.e.* $\dot{\omega}_i \approx 0$ for $i = 1..4$. Then the equation of the wheel dynamics given by Equation (5.1b) becomes:

$$\mathrm{T}_{tot} \approx r_{eff} \sum_{i=1}^{4} \mathrm{F}_{xpi} \tag{5.4a}$$

Thus the longitudinal forces are proportional to the control T_{tot}. Neglecting the aerodynamic drag forces, and defining $\mathrm{K}'_\mathrm{P} = \frac{\mathrm{K_P}}{r_{eff}}$ and $\mathrm{K}'_\mathrm{I} = \frac{\mathrm{K_I}}{r_{eff}}$, Equation (5.1a) becomes:

$$\mathrm{M_T}\,\dot{v} = \frac{\mathrm{T}_{tot}}{r_{eff}} \tag{5.4b}$$

$$\mathrm{M_T}\,(\dot{e}_v + \dot{v}_r) = -\mathrm{K}'_\mathrm{P} e_v(t) - \mathrm{K}'_\mathrm{I} \int_{t_0}^{t} e_v(\tau)\mathrm{d}\tau \tag{5.4c}$$

Therefore, in regulation *i.e.* when $\dot{v}_r = 0$, the velocity error e_v converges to 0. However, performances in tracking of a PI controller is limited: thus, if the reference speed profile is highly dynamic and varies constantly, the performance might be unsatisfactory. The integral terms enables to cancel the perturbations caused by the aerodynamic drag forces.

5.2 Review of lateral controllers

Lateral controllers for autonomous vehicles have been widely studied in the literature and comprise a wide range of control techniques. Among others, let's mention:

- Fuzzy logic [Hessburg and Tomizuka, 1994, Zalila et al., 1998, Wang et al., 2015];
- H_∞ controllers [Hingwe et al., 2002];
- Proportional-Integral-Derivative (PID) controllers [Hessburg et al., 1991, Sharp et al., 2000, Menhour et al., 2011b];
- Model Predictive Control [Falcone et al., 2007b, Shinohara et al., 2016];
- Sliding mode control [Hingwe and Tomizuka, 1997, Zhao et al., 2007, Talj et al., 2013];

- Adaptive control [Netto et al., 2004].

Some works have compared different approaches such as [Chaib et al., 2004] which compares H_∞, adaptive, PID and fuzzy logic lateral controllers. However, making a full review of all the contributions to lateral control would be tedious, time-consuming and moreover not very relevant. Instead, we chose to make a comparative study limited to five types of lateral controllers that have been applied successfully on actual vehicles and are implemented in practice. These controllers are : (i) the Proportional-Integral-Derivative (PID), (ii) the Stanley controller (SC), (iii) the pure-pursuit (PP), (iv) a non-linear controller based on the kinematic bicycle model using chained form transform (KBCF), and (v) a controller based on a linearized dynamic bicycle model (LDBM).

While the PID and Stanley controllers can be considered as model-free controllers as their control laws are not directly derived from a vehicle model, both the PP, the KBCF and the LDBM are model-based. More precisely, the two first are based on the kinematic bicycle model presented in section 2.4.3, and the latter has been derived from the dynamic bicycle model (see section 2.1.1) using a linear tire model (see section 2.2.2). Model-based controllers require some basic knowledge about the vehicle such as for example the distance between the center of gravity and the center of the front axle l_f (resp. rear axle l_r).

These different controllers have already been compared in simulation on three different tracks, namely a lane change, a eight-shaped and a road course [Snider, 2009]. However, no dynamic limitations were derived in order to ensure their ability to track a given reference trajectory. Such a result is important for ensuring the consistency between the motion planning and the control layer. Imagine that the reference trajectory generated by the motion planner is dynamically challenging: it might be that a given lateral controller would not be able to perform well in such a situation because it has been derived assuming low slip angles for example. Therefore, we will present in the rest of this section the five controllers aforementioned, understand what are the underlying assumptions behind each one and define their operational range.

Remark 12. *All the control techniques presented in the rest of this section are defined by assimilating the vehicle to a bicycle model: the two front wheels (resp. the two rear wheels) of the vehicle are lumped into a unique wheel located at the center of the front axle (resp. of the rear axle) such as illustrated in Figure 5.2. Moreover, we assume that only the front wheel can be steered.*

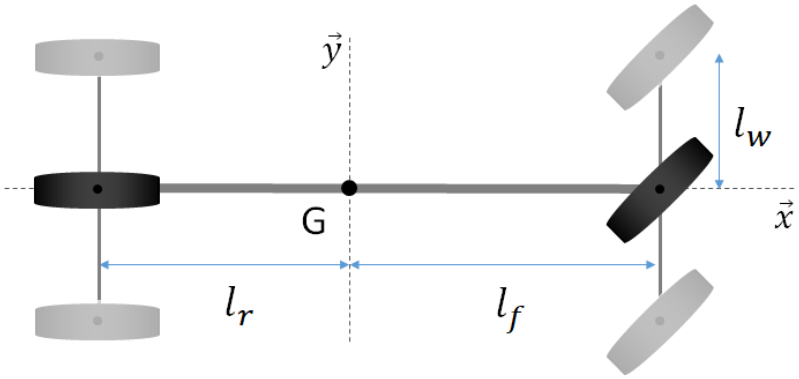


Figure 5.2 – The bicycle model: the two front (resp. rear) wheels (in grey) are lumped into a unique wheel (in black).

5.2.1 Proportional-Integral-Derivative (PID)

Proportional-Integral-Derivative controllers are the most popular controllers in industrial application[1]. They have been applied for the first time in [Minorsky, 1922] and widely studied since

[1] https://taskforce.ifac-control.org/industry-committee/legacy-materials-pilot-industry-committee-2014-2017/meeting-materials/meeting-presentation-17-june-2015/view, visited on March 2018.

then in the literature (see for example [Astrom and Hagglund, 1995]). The principle of a PID controller is the following; the tracking error $e = y - y^r$, defined as the difference between the system output y and the reference output y^r, is corrected by applying a control u that is composed of three terms as shown in Equation (5.5): one proportional to the error e, one proportional to the derivative of the error $\dot{e}$, and one proportional to the cumulative sum of the error which corresponds to its integral $\int e$.

$$u(t) = -K_P e(t) - K_D \dot{e}(t) - K_I \int_0^t e(\tau) d\tau \tag{5.5}$$

For lateral control application, the error e can either be the lateral error $\tilde{y} = y - y^r$, *i.e.* the distance between the vehicle and the reference trajectory, or the heading angle error $\tilde{\psi} = \psi - \psi^r$. The lateral and heading angle errors are usually computed either at the center of the rear axle, at the center of gravity G of the vehicle or at a point P located at a preview distance L_P ahead of the vehicle as shown on Figure 5.3. In the third case, the stability of the system is improved as the vehicle anticipates the curvature of the reference trajectory, especially during cornering. Lateral PID controllers often comprise a PID on the lateral error $\tilde{y} = y - y^r$ given by Equation (5.6a) and on the heading angle error $\tilde{\psi} = \psi - \psi^r$ given by Equation (5.6b), which lead to Equation (5.6c). In [Sharp et al., 2000] for example, the final control law is a ponderation of PID controllers on the lateral errors located at different preview distances.

$$\delta_y(t) = -K_{P_y} \tilde{y}(t) - K_{D_y} \dot{\tilde{y}}(t) - K_{I_y} \int_0^t \tilde{y}(\tau) d\tau \tag{5.6a}$$

$$\delta_\psi(t) = -K_{P_\psi} \tilde{\psi}(t) - K_{D_\psi} \dot{\tilde{\psi}}(t) - K_{I_\psi} \int_0^t \tilde{\psi}(\tau) d\tau \tag{5.6b}$$

$$\delta_{PID}(t) = \delta_y(t) + \delta_\psi(t) \tag{5.6c}$$

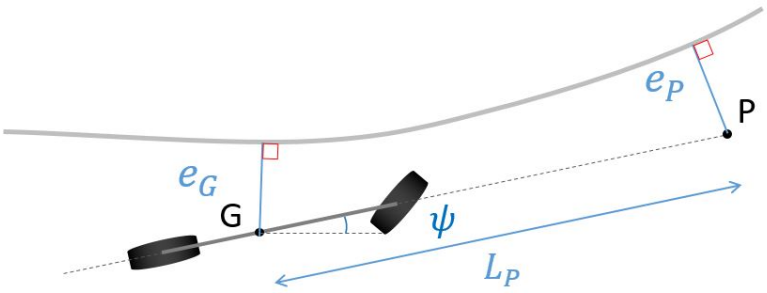


Figure 5.3 – PID controller.

	Accuracy	Stability	Reactivity
K_P	+	-	+
K_I	+	-	-
K_D	-	+	+

Table 5.1 – Influence of the gains of a PID controller on the behavior of the closed-loop system.

The gains of a PID controller, namely the proportional K_P, the integral K_I and the derivative K_D, all play a different role in terms of stability, accuracy and time response of the closed-loop system as illustrated in Table 5.1. Therefore, tuning correctly a lateral PID controller is complex and time consuming. Due to the coupling with longitudinal control, conventional techniques such as the Ziegler-Nichols method [Ziegler and Nichols, 1993] cannot be used. In particular, the tuning of the gains depends strongly on the speed of the vehicle: the faster the vehicle is, the

smaller the gains should be in order to avoid oscillations and even instability (see for example [Menhour et al., 2011b]). Moreover for the same reason mentioned in section 5.1.2, the PID controller is efficient for regulation but not for tracking.

5.2.2 Pure-pursuit (PP)

The pure-pursuit algorithm has been introduced in [Coulter, 1992] and applied on Carnegie Mellon University's NavLAB vehicles. It consists in drawing a circular arc joining the center of the rear axle to a point P on the reference path located at a preview distance L_P from the rear axle as shown in Figure 5.4. The control law given by Equation (5.7) is then derived from the kinematic bicycle model in order to follow this circular arc:

$$\delta_{PP}(t) = \operatorname{atan}\left(\frac{2(l_f + l_r)\sin\alpha}{L_p}\right) \tag{5.7}$$

where α is the angle between the orientation of the vehicle and the vector going from the center of the rear axle to the preview point P.

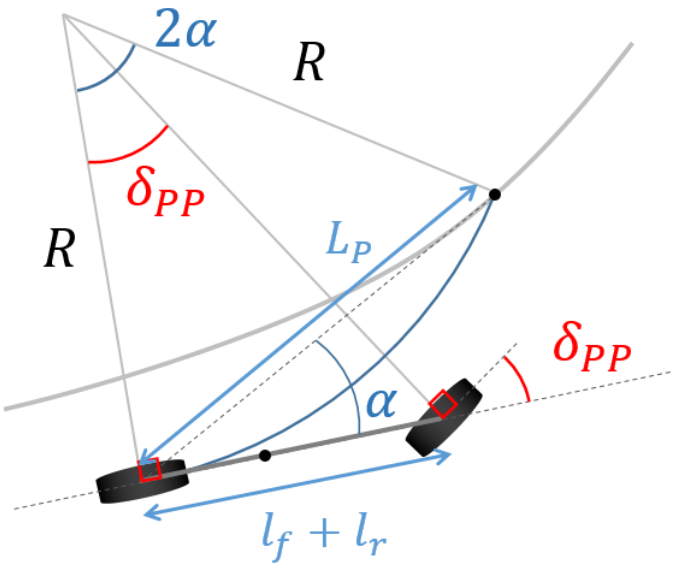


Figure 5.4 – Pure-pursuit controller.

More precisely, Equation (5.7) was obtained as following:

1. Assuming the slip angle is zero (*i.e.* using a kinematic bicycle model), the steering angle δ_{PP} necessary for the center of the rear axle to follow the circular arc of radius R can be expressed as:

$$\tan\delta_{PP} = \frac{l_f + l_r}{R} \tag{5.8}$$

2. Then, noticing that $\sin(\alpha) = \frac{L_P}{2R}$ leads to the expected result.

One of the main difficulties for implementing a pure-pursuit algorithm is the setting of the preview distance L_P. [Snider, 2009] showed that the shorter the preview distance is, the more accurate but the less smooth the tracking is. Moreover, as the pure-pursuit controller ignores the curvature of the path, the preview distance depends strongly on the shape of the path and thus on the scenario: for example, a large preview distance is better suited to follow a long straight path while a short distance is necessary in curves to avoid cutting corners. This is a major drawback for pure-pursuit controller as the preview distance will necessarily vary. One solution is to increase

the preview distance with the speed, and set the reference speed according to the curvature of the path.

However, as pure-pursuit controllers do not use explicitly information about the reference path, the latter does not need to have any smoothness properties. It can even be discontinuous (as long as the preview point is defined). In particular, this makes them very robust to fast transient sections [Snider, 2009]. Finally, as the pure-pursuit controller relies on a kinematic bicycle model, it should only be used for trajectories where the lateral acceleration a_y remains below $0.5\mu g$ (see chapter 4.2), where μ is the friction coefficient of the road and g the gravitational constant.

5.2.3 Stanley controller (SC)

The Stanley controller was designed for the eponymous vehicle from Stanford University that won the DARPA Grand Challenge in 2005 [Thrun et al., 2006]. It is a control law based on the lateral error e_f and the heading error $\tilde{\psi}_f = \psi_f - \psi^r$ both taken at the center of the front axle of the vehicle (see Figure 5.5). Its expression is given by Equation (5.9), where v_{xf} is the velocity of the front wheel and k is a control gain:

$$\delta_{SC}(t) = \tilde{\psi}(t) + \arctan\left(\frac{k e_f(t)}{v_{xf}(t)}\right) \tag{5.9}$$

Exponential convergence has been proved for small front lateral errors using a linear bicycle model with infinite tire stiffness and tight steering angle limitations [Thrun et al., 2006].

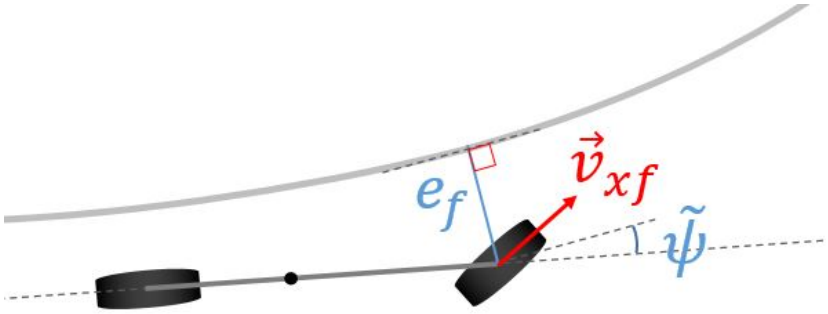


Figure 5.5 – Stanley controller.

[Snider, 2009] showed that higher values of k increase the tracking performance but there is an upper limit for stability. Moreover, it is neither robust to lane change maneuvers as only low values of k work, nor to discontinuities of the reference path. However, it is better suited for higher speeds than the pure-pursuit algorithm.

5.2.4 Non-linear controller using chained form transform on the kinematic bicycle model (KBCF)

The kinematic bicycle model, given by Equations (2.43) in section 2.43, can be expressed in a Frenet frame at the projection of the rear axle on the reference path [Weiskircher and Ayalew, 2015] as shown on Figure 5.6. The system obtained is given by Equations (5.10a)-(5.10c).

$$\dot{s} = \frac{v_r \cos\tilde{\psi}}{1 - \kappa\tilde{y}} \tag{5.10a}$$

$$\dot{\tilde{y}} = v_r \sin\tilde{\psi} \tag{5.10b}$$

$$\dot{\tilde{\psi}} = v_r\left(\frac{\tan\delta}{l_f + l_r} - \frac{\kappa\cos\tilde{\psi}}{1 - \kappa\tilde{y}}\right) \tag{5.10c}$$

where κ is the curvature of the reference path, s the curvilinear abscissa, $\tilde{y}$ the lateral error at the center of the rear axle, $\tilde{\psi} = \psi - \psi^r$ the heading error and v_r the velocity at the center of the rear axle.

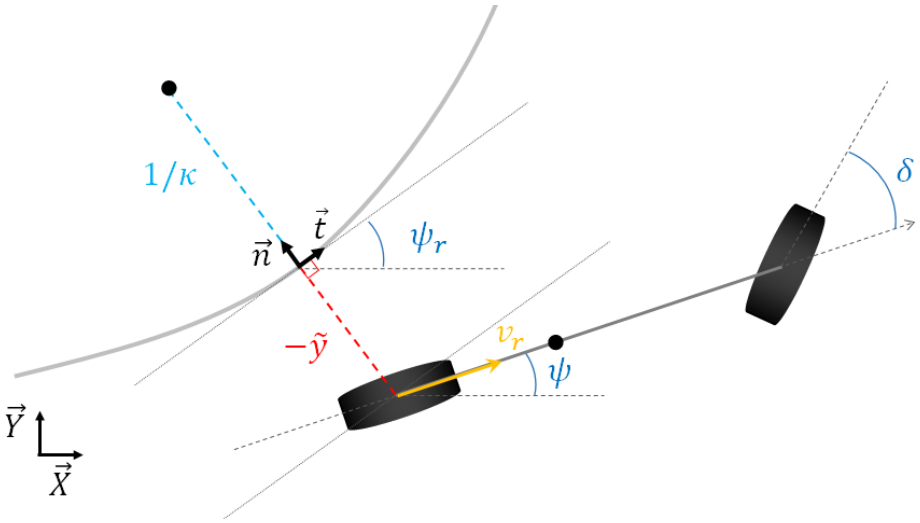


Figure 5.6 – Frenet frame for the kinematic bicycle model.

In [Samson, 1995], the author proposed to transform the system composed of Equations (5.10a)-(5.10c) into a chained form system. The latter is a system that can be written:

$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_1 x_3 \\ \dot{x}_3 &= u_1 x_4 \\ &\vdots \\ \dot{x}_{n-1} &= u_1 x_n \\ \dot{x}_n &= u_2 \end{aligned} \tag{5.11}$$

or equivalently, in matrix form:

$$\dot{\mathrm{X}} = h_1(\mathrm{X})\,u_1 + h_2(\mathrm{X}) \tag{5.12a}$$

with

$$\mathrm{X} = \begin{pmatrix} x_1 & x_2 & \cdots & x_{n-1} & x_n \end{pmatrix}^{\mathrm{T}} \tag{5.12b}$$

$$h_1(\mathrm{X}) = \begin{pmatrix} 1 & x_3 & x_4 & \cdots & x_n & 0 \end{pmatrix}^{\mathrm{T}} \tag{5.12c}$$

$$h_2(\mathrm{X}) = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \end{pmatrix}^{\mathrm{T}} \tag{5.12d}$$

Applying the state mapping $a_1 = s$, $a_2 = \tilde{y}$, $a_3 = (1-\kappa\tilde{y})\tan\tilde{\psi}$ and the control mapping $m_1 = \dot{a}_1$, $m_2 = \dot{a}_3$ to Equations (5.10), the following chained system is obtained where the derivative is performed with respect to time:

$$\dot{a}_1 = m_1 \tag{5.13a}$$

$$\dot{a}_2 = a_3 m_1 \tag{5.13b}$$

$$\dot{a}_3 = m_2 \tag{5.13c}$$

This system can then be transformed into a linear system where the derivative is performed with respect to the curvilinear abscissa s, by defining $m_3 = \frac{m_2}{m_1}$:

$$a_1' = 1 \tag{5.14a}$$

$$a_2' = a_3 \tag{5.14b}$$

$$a_3' = m_3 \tag{5.14c}$$

Choosing $m_3 = -K_D a_3 - K_P a_2$ with $(K_P, K_D) \in (\mathbf{R}^+)^2$ leads to Equation (5.15). Thus $a_2 = \tilde{y}$ and $a_2' = a_3 = (1-\kappa\tilde{y})\tan\tilde{\psi}$ converge to zero. Therefore, $\tilde{y}$ and $\tilde{\psi}$ both converge to zero. K_P and K_D are tuned such that the system converges to the reference path in a given distance that does not depend on the velocity v_r of the vehicle.

$$a_2'' + K_D a_2' + K_P a_2 = 0 \tag{5.15}$$

The non-linear control law obtained after doing all the inverse mapping is given by Equations (5.16) and ensures the convergence to the reference path.

$$\delta_{KBM} = \operatorname{atan}\left((l_f + l_r)\left[\frac{\cos^3\tilde{\psi}}{(1-\kappa\tilde{y})^2}A + \frac{\kappa\cos\tilde{\psi}}{1-\kappa\tilde{y}}\right]\right) \tag{5.16a}$$

where

$$A = \frac{d\kappa}{dt}\tilde{y}\tan\tilde{\psi} - K_D(1-\kappa\tilde{y})\tan\tilde{\psi} - K_P\tilde{y} + \kappa(1-\kappa\tilde{y})\tan^2\tilde{\psi} \tag{5.16b}$$

This non-linear controller requires the reference path to be at least C^2 in order to compute the curvature κ and even C^3 due to the term $\frac{d\kappa}{dt}$. It has been implemented on actual vehicles such as in [Sotelo, 2003] and [Thuilot et al., 2004]. Like the pure-pursuit controller, it is based on a kinematic bicycle model: therefore, it should only be used for trajectories where the lateral acceleration a_y remains below $0.5\mu g$ (see chapter 4.2). Extension to situations where the front or rear slip angles (denoted respectively α_f and α_r) cannot be neglected have been proposed in [Lenain, 2005] for off-road vehicles and are given by Equations (5.17). In that case, estimators of the slip angles are necessary (see for example [Lenain et al., 2010, Baffet et al., 2009]).

$$\delta_{KBM2} = \operatorname{atan}\left(\frac{l_f + l_r}{\cos\alpha_r}\left[\frac{\cos^3\tilde{\psi}_2}{(1-\kappa\tilde{y})^2}A_2 + \frac{\kappa\cos\tilde{\psi}_2}{1-\kappa\tilde{y}}\right] + \tan\alpha_r\right) - \alpha_f \tag{5.17a}$$

with

$$A_2 = -K_D(1-\kappa\tilde{y})\tan\tilde{\psi}_2 - K_P\tilde{y} + \kappa(1-\kappa\tilde{y})\tan^2\tilde{\psi}_2 \tag{5.17b}$$

$$\tilde{\psi}_2 = \tilde{\psi} + \alpha_r \tag{5.17c}$$

5.2.5 Linearized dynamic bicycle model controller (LDBM)

Let's recall the equations of the dynamic bicycle model that have been introduced in section 2.1.1:

$$M_T(\dot{V}_x - \dot{\psi}V_y) = F_{xf} + F_{xr} \tag{5.18a}$$

$$M_T(\dot{V}_y + \dot{\psi}V_x) = F_{yf} + F_{yr} \tag{5.18b}$$

$$I_z\ddot{\psi} = l_f F_{yf} - l_r F_{yr} \tag{5.18c}$$

F_{xf} and F_{xr} correspond to the longitudinal forces generated by the road on respectively the front and rear tire, in the vehicle frame. Similarly, F_{yf} and F_{yr} correspond to the lateral forces in the vehicle frame.

Assuming small slip angles, the linear tire model given by the following equations can be used:

$$F_{ypf} = C_f\alpha_f \tag{5.19a}$$

$$F_{ypr} = C_r\alpha_r \tag{5.19b}$$

where F_{ypf} and F_{ypr} are the lateral forces generated by the road respectively on the front and rear wheel, expressed in the pneumatic frame (subscript "p"), C_f and C_r are the front and rear cornering stiffness, and α_f and α_r are the front and rear slip angles, given by the following expression:

$$\alpha_f = \delta_f - \operatorname{atan}\left(\frac{V_y + l_f\dot{\psi}}{V_x}\right) \tag{5.20a}$$

$$\alpha_r = -\operatorname{atan}\left(\frac{V_y - l_r\dot{\psi}}{V_x}\right) \tag{5.20b}$$

The relation between the forces F_{yf} and F_{yr} expressed in the vehicle frame and F_{xpf}, F_{ypf} and F_{ypr} expressed in the pneumatic frame is given by Equations (5.21):

$$F_{yf} = F_{ypf}\cos\delta_f + F_{xpf}\sin\delta_f \tag{5.21a}$$
$$F_{yr} = F_{ypr} \tag{5.21b}$$

Since we consider a decoupled lateral controller, we neglect the longitudinal dynamics, *i.e.* we assume a constant longitudinal velocity V_x (or at least that it varies slowly) and that the longitudinal tire forces on the front wheel in the tire frame $F_{xpf} = 0$. Therefore, assuming also that all the angles are small, Equations (5.18b)-(5.18c) lead to the state-space model:

$$\begin{pmatrix}\dot{V}_y\\ \ddot{\psi}\end{pmatrix} = \begin{pmatrix}\frac{-(C_f+C_r)}{M_T V_x} & \frac{l_r C_r - l_f C_f}{M_T V_x} - V_x\\ \frac{l_r C_r - l_f C_f}{I_z V_x} & \frac{-(l_f^2 C_f + l_r^2 C_r)}{I_z V_x}\end{pmatrix}\begin{pmatrix}V_y\\ \dot{\psi}\end{pmatrix} + \begin{pmatrix}\frac{C_f}{m}\\ \frac{l_f C_f}{I_z}\end{pmatrix}\delta \tag{5.22}$$

This state-space model can be expressed in terms of error with respect to the reference path. Defining $\tilde{y}$ as the lateral error at the center of gravity of the vehicle and $\tilde{\psi} = \psi - \psi^r$ the heading error, Equation (5.22) becomes:

$$\begin{pmatrix}\dot{\tilde{y}}\\ \ddot{\tilde{y}}\\ \dot{\tilde{\psi}}\\ \ddot{\tilde{\psi}}\end{pmatrix} = \begin{pmatrix}0 & 1 & 0 & 0\\ 0 & \frac{-(C_f+C_r)}{M_T V_x} & \frac{C_f+C_r}{M_T} & \frac{l_r C_r - l_f C_f}{M_T V_x}\\ 0 & 0 & 0 & 1\\ 0 & \frac{l_r C_r - l_f C_f}{I_z V_x} & \frac{l_r C_r - l_f C_f}{I_z} & \frac{-(l_f^2 C_f + l_r^2 C_r)}{I_z V_x}\end{pmatrix}\begin{pmatrix}\tilde{y}\\ \dot{\tilde{y}}\\ \tilde{\psi}\\ \dot{\tilde{\psi}}\end{pmatrix} + \begin{pmatrix}0\\ \frac{C_f}{M_T}\\ 0\\ \frac{l_f C_f}{I_z}\end{pmatrix}\delta + \begin{pmatrix}0\\ -\frac{l_f C_f - l_r C_r}{M_T V_x} - V_x\\ 0\\ -\frac{l_f^2 C_f + l_r^2 C_r}{I_z V_x}\end{pmatrix}\dot{\psi}_r \tag{5.23}$$

where $\dot{\psi}^r = \kappa V_x$ and κ the curvature of the reference path.

The control law given by Equation (5.24) can then be derived in order for the error vector $\xi = (\tilde{y}; \dot{\tilde{y}}; \tilde{\psi}; \dot{\tilde{\psi}})^T$ to converge to zero. All the details can be found in [Rajamani, 2012].

$$\delta_{\text{LDBM}} = \delta_{ff} + \delta_{fb} \tag{5.24a}$$

where δ_{ff} (resp. δ_{fb}) are the feedforward (resp. feedback) control laws:

$$\delta_{ff} = \kappa(l_f + l_r) + \left(\frac{l_r M_T}{(l_f + l_r)C_f} - \frac{l_f M_T}{(l_f + l_r)C_r}\right) a_y - k_3 \tilde{\psi}_{ss} \tag{5.24b}$$
$$\delta_{fb} = -K\xi \tag{5.24c}$$
$$= -k_1 \tilde{y} - k_2 \dot{\tilde{y}} - k_3 \tilde{\psi} - k_4 \dot{\tilde{\psi}}$$

$\tilde{\psi}_{ss}$ corresponds to the steady-state yaw angle error:

$$\tilde{\psi}_{ss} = -\kappa l_r + \frac{l_f}{C_r(l_f + l_r)} M_T V_x^2 \kappa \tag{5.24d}$$

Varying longitudinal velocity

In the case where the longitduinal velocity varies, Equation (5.23) becomes a linear parameter varying (LPV) system as shown in Equation (5.25):

$$\dot{\xi} = A(V_x)\xi + B_1 u + B_2(V_x)\dot{\psi}_r \tag{5.25}$$

Let's assume that $V_x \in [V_{x_{min}}, V_{x_{max}}]$. Set $A_{min} = A(V_{x_{min}}) - B_1 K$ and $A_{max} = A(V_{x_{max}}) - B_1 K$.

Theorem 1. *If* K *is chosen such that for some* $P > 0$, $A_{min}^T P + P A_{min} < 0$ *and* $A_{max}^T P + P A_{max} < 0$, *then the closed-loop system is stable for all* $V_x \in [V_{x_{min}}, V_{x_{max}}]$.

Proof. First, let us notice that $A(V_x) = \lambda A_{min} + (1-\lambda) A_{max}$ with $\lambda \in [0,1]$ is a function of V_x.

Assuming that $\exists P > 0, (A_{min}^T P + P A_{min} < 0) \wedge (A_{max}^T P + P A_{max} < 0)$, define $\mathcal{V} : x \to x^T P x$. Let's proove that $\mathcal{V}$ is Lyapunov function:

- $\mathcal{V}(x) = 0 \Rightarrow x = 0$ since $P > 0$.
- $\dot{\mathcal{V}} = \lambda x^T (A_{min}^T P + P A_{min}) x + (1-\lambda) x^T (A_{max}^T P + P A_{max}) x < 0$ by definition of P and λ
- $\lim_{||x|| \to +\infty} \mathcal{V}(x) = +\infty$ by definition of P.

□

5.3 Implementation and comparison of the lateral controllers

After presenting different lateral controllers, we implemented the pure-pursuit, the KBCF and the Stanley controllers on an actual vehicle in collaboration with the CyberC3 laboratory of Shanghai Jiao-Tong University. Section 5.3.1 introduces the different methods for computing the lateral and heading angle errors given a reference path. Then section 5.3.2 presents the experimental vehicle while section 5.3.3 explains how to implement the pure-pursuit, the KBCF and the Stanley controllers in practice. The performances of these controllers are compared in section 5.3.4 before concluding with section 5.3.5.

5.3.1 Computation of the lateral and heading angle errors

All lateral controllers require to compute the lateral error $\tilde{y}$ and/or the heading angle error $\tilde{\psi}$. While their implementation details are usually not mentioned in publications, they are not straightforward in practice. This is mainly due to a discrete representation of the continuous reference path which is necessary in order to be stored on a computer. This representation, named *waypoints*, contains a list of reference positions (X_i^r, Y_i^r), $i = 1..n$ for the vehicle. Therefore, this section will present cubic spline and polynomial interpolation techniques, as well as different methods for computing the lateral error and the heading angle error.

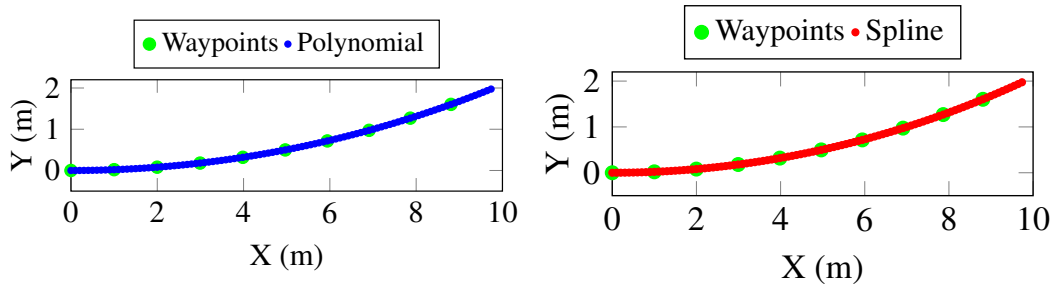
Interpolation of the list of waypoints

In order to compute the lateral or the heading angle errors, it is usually necessary to get the closest point on the reference path to the vehicle. Therefore, the closer the waypoints are to each other, the more accurate the lateral errors and the heading angle errors are. However, computational costs are increased both in time and in space: first, the motion planner needs a smaller time step discretization to provide a denser list of waypoints which increases the computational time drastically; then, the list of waypoints needs to be stored on the memory of the computer which consumes $\mathcal{O}(n)$ in space; finally, the look-up time for the closest point is $\mathcal{O}(n)$ in time. One way to limit these computational burdens is to increase the number of waypoints only on a short time horizon, corresponding to the path the controller will be able to track before replanning the reference trajectory. This can be done using polynomial or spline interpolations as shown on Figure 5.7a and 5.7b. Note that the linear interpolation should be avoided as it provides a non-smooth interpolation in curves.

Interpolations of the waypoints should be done with respect to its curvilinear abscissa s, which is strictly increasing[2], in order to be properly defined. Otherwise, in the case where there are two values $Y_{j_1}^r$ and $Y_{j_2}^r$ corresponding to one X^r on the reference path, the interpolation of the Y-coordinates with respect to the X-coordinates is not defined. The curvilinear abscissa s_i^r of each waypoint i can be computed from its coordinates (X_i^r, Y_i^r) and the previous waypoint (X_{i-1}^r, Y_{i-1}^r) using to Equation (5.26a).

$$s_i^r = \begin{cases} 0, & \text{if } i = 1 \\ s_{i-1}^r + \sqrt{(X_i^r - X_{i-1}^r)^2 + (Y_i^r - Y_{i-1}^r)^2}, & \text{otherwise} \end{cases} \tag{5.26a}$$

[2] Except in the cases where the vehicle is at a full stop



(a) Polynomial interpolation

(b) Spline interpolation

Figure 5.7 – Interpolation of waypoints.

Set $X^r = (X_1^r, .., X_n^r)$, $Y^r = (Y_1^r, .., Y_n^r)$ and $s^r = (s_1^r, .., s_n^r)$. Define a new list of curvilinear abscissa s_{interp} where the distance between two consecutive curvilinear abscissa is smaller than for s^r. Applying Equations (5.26b) and (5.26c) where *interpolation* is either a polynomial or spline interpolation function, a new denser list of waypoints (X_{interp}, Y_{interp}) is obtained. Similarly, an interpolation of the heading angle ψ to compute the heading angle error can be obtained using Equation (5.26d) as illustrated by Equation (5.26d).

$$X_{interp} = \text{interpolation}(s^r, X^r, s_{interp}) \tag{5.26b}$$
$$Y_{interp} = \text{interpolation}(s^r, Y^r, s_{interp}) \tag{5.26c}$$
$$\psi_{interp} = \text{interpolation}(s^r, \psi^r, s_{interp}) \tag{5.26d}$$

In the rest of this section, we present three methods to compute the lateral and heading angle errors:

Method 1: the closest point

The simplest method to compute the lateral error and the heading angle error is to consider the closest waypoint (X_i^r, Y_i^r) from the vehicle and its tangent unit vector $\vec{t}_i^r$:

1. Find the closest waypoint (X_i^r, Y_i^r).

2. Compute its tangent unit vector $\vec{t}_i^r$ based on the waypoints at index $i-1$ and $i+1$:

$$\vec{t}_i^r = \frac{1}{\sqrt{(X_{i+1}^r - X_{i-1}^r)^2 + (Y_{i+1}^r - Y_{i-1}^r)^2}} \begin{pmatrix} X_{i+1}^r - X_{i-1}^r \\ Y_{i+1}^r - Y_{i-1}^r \end{pmatrix} \tag{5.27a}$$

3. Compute the normal unit vector $\vec{n}_i^r$ by rotating $\vec{t}_i^r$ counter-clockwise:

$$\vec{n}_i^r = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \vec{t}_i^r \tag{5.27b}$$

4. Define $\vec{r} = \begin{pmatrix} X - X_i^r \\ Y - Y_i^r \end{pmatrix}$ as the vector joining the closest point and the vehicle.

5. The lateral error $\tilde{y}$ is obtained from Equation (5.27c) where $\langle ., . \rangle$ corresponds to the scalar product.

$$\tilde{y} = \langle \vec{n}_i^r, \vec{r} \rangle \tag{5.27c}$$

6. The reference heading angle ψ_i^r is defined as follows:

$$\psi_i^r = \text{acos}\left(\langle \vec{t}_i^r, \begin{pmatrix}1\\0\end{pmatrix}\rangle\right) \tag{5.27d}$$

7. The heading angle error $\tilde{\psi}$ corresponds to:

$$\tilde{\psi} = \psi - \psi_i^r \tag{5.27e}$$

Although this method is simple to implement in practice, it is not very accurate if the density of the waypoints is not sufficient. Therefore, interpolation techniques presented previously should be used beforehand.

Method 2: the lateral displacement

The lateral displacement method to compute $\tilde{y}$ considers the distance between the vehicle (X, Y) and the tangent to the path at (X_i^r, Y_i^r), where (X_i^r, Y_i^r) is the closest waypoint to the vehicle (see Figure 5.8). Its expression is given by Equation (5.28). Of course, the denser the list of waypoints is, the better the results are. Therefore it can be useful to apply the aforementioned interpolation techniques first. The reference heading angle ψ^r and heading angle error $\tilde{\psi}$ are computed as in the previous method.

$$\tilde{y} = (Y - Y_i^r)\cos\psi_i^r - (X - X_i^r)\sin\psi_i^r \tag{5.28}$$

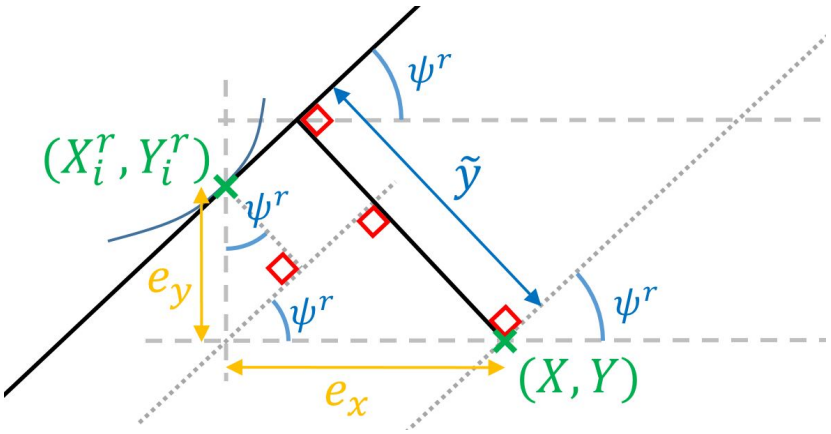


Figure 5.8 – The lateral displacement.

Method 3: the pseudo-distance

This method was introduced in [Ziegler et al., 2014a]. Unlike the previous methods, it does not require any interpolation beforehand. Instead, a linear interpolation of the tangent vectors at the two closest waypoints is made:

1. Find the closest waypoint (X_i^r, Y_i^r) to the vehicle.
2. Compute its tangent vector $\vec{t}_i^r$ based on the waypoints at index $i-1$ and $i+1$.
3. Define $\vec{r} = \begin{pmatrix} X - X_i^r \\ Y - Y_i^r \end{pmatrix}$ which corresponds to the vector joining the closest point and the vehicle.
4. Compute ϵ given by Equation (5.29a). If $\epsilon \geq 0$, define $P_{prev} = (X_i^r, Y_i^r)$ and $P_{next} = (X_{i+1}^r, Y_{i+1}^r)$; else if $\epsilon < 0$, define $P_{prev} = (X_{i-1}^r, Y_{i-1}^r)$ and $P_{next} = (X_i^r, Y_i^r)$. P_{prev} and P_{next} are the two nearest waypoints located respectively before and after the vehicle.

$$\epsilon = \text{sign}\left(\langle \vec{r}, \vec{t}_i^r \rangle\right) \tag{5.29a}$$

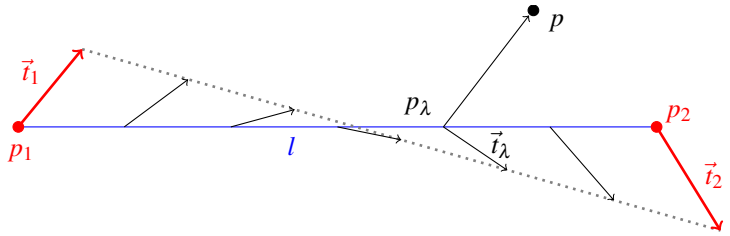


Figure 5.9 – The pseudo distance.

5. Compute the tangents $\vec{\mathrm{T}}_{prev}$ and $\vec{\mathrm{T}}_{next}$ of the path at P_{prev} and P_{next}.

6. Make a frame change centered in P_{prev} and where the x-abscissa coincides with the axis $(\mathrm{P}_{prev}, \mathrm{P}_{next})$. Define $p_1 = (0,0)$ and $p_2 = (l,0)$ as the coordinates of respectively P_{prev} and P_{next} in the new frame as shown in Figure 5.9. Define $t_1 = (1, m_1)$ and $t_2 = (1, m_2)$ as the tangents at p_1 and p_2 in the new frame.

7. A pseudo tangent t_λ given by Equation (5.29b) is created by linear interpolation along the segment $[p_1, p_2]$ at the point p_λ expressed by Equation (5.29c), with $\lambda \in [0,1]$.

$$\vec{t}_\lambda = \lambda \vec{t}_2 + (1-\lambda)\vec{t}_1 \tag{5.29b}$$
$$p_\lambda = \lambda p_2 + (1-\lambda) p_1 \tag{5.29c}$$

8. The position of the vehicle, denoted by $p = (x_v, y_v)$ in the new frame, is projected on the segment $[p_1, p_2]$ on p_λ such that the pseudo normal vector $\vec{n}_\lambda = p - p_\lambda$ verifies Equation (5.29d).

$$\langle \vec{n}_\lambda, \vec{t}_\lambda \rangle = 0 \tag{5.29d}$$

As $p_1 = (0,0)^{\mathrm{T}}$ and $p_2 = (l,0)^{\mathrm{T}}$, the solution is:

$$\lambda = \frac{m_1 y_v + x_v}{(m_1 - m_2) y_v + l} \tag{5.29e}$$

9. The pseudo-distance is then:

$$\tilde{y} = \mathrm{sign}(y_v) ||n_\lambda|| \tag{5.29f}$$

Remark 13. *The frame change enables to compute* λ *easily. Otherwise,* λ *would be the solution of a second order polynomial equation* $\mathrm{A}\lambda^2 + \mathrm{B}\lambda + \mathrm{C} = 0$ *which not very robust to variations of the coefficient due to small computation errors.*

5.3.2 Experimental set-up

The pure-pursuit (PP), the non-linear controller based on the kinematic bicycle model (KBCF) and the Stanley controllers (SC) have been tested on the CyberGL8 vehicle belonging to the CyberC3 laboratory of Shanghai Jiao-Tong University. The vehicle is shown in Figure 5.10. It is equipped with a 10Hz Hemisphere GPS un237c located at the center of the rear axle of the vehicle and a 100Hz Bosch SMI130 IMU. The odometers are integrated in the original GL8 vehicle and provide measurements at 100Hz. The embedded computer is an Intel NUC 6I7KYK. The rest of its sensors are not used in these experiments.

In order to localize the vehicle, an Extended Kalman Filter is used based on the GPS, the IMU and the odometer measurements. Thus, the localization is not only more accurate but also provided at a higher frequency, 100Hz, even though the frequency of the GPS is only 10Hz. The position of the center of the rear axle of the vehicle is given in the inertial coordinate system. The control of the steering wheel is ensured by a steer-by-wire system working at 100Hz.

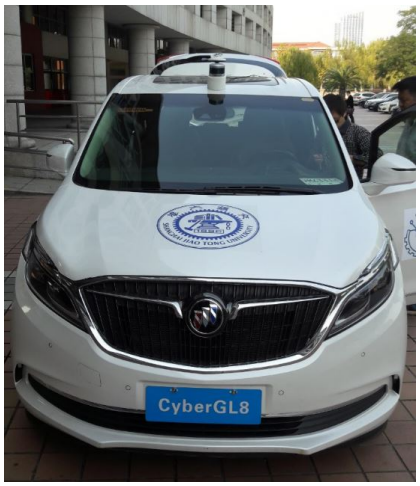


Figure 5.10 – Shanghai Jiao-Tong's CyberGL8 vehicle used for experimentations.

Experimentations on an actual vehicle enable to test the controllers in an environment comprising noise on the measurements, actuation delays and parameter uncertainties which are inherent to any real-world system.

The reference path $(X_i^r, Y_i^r)_{i\in[\![1,n]\!]}$ was obtained by recording the successive positions of the vehicle driven by a human driver on the campus of Shanghai Jiao-Tong University. The position is the one given by the EKF at 100Hz. The velocity was kept low, below 20km/h. Therefore, the distance between two waypoints on the reference path is small (less than 10cm). The reference path is shown in Figure 5.11. In order to compute the lateral error $\tilde{y}$ and the angular error $\tilde{\psi}$, we used method 2 described in section 5.3.1 called the "lateral displacement method". Interpolation of the waypoints is not necessary due to the small distance between two points.

Figure 5.11 – Reference path (in red) in the Shanghai Jiao-Tong University campus.

5.3.3 Implementation and tuning of the controllers

For the three lateral controllers tested, the reference path is given at the center of the rear axle. While this is naturally the case for the KBCF and the pure-pursuit controllers according to their definitions, it is less obvious for the Stanley controller as it corrects the lateral error at the front wheel. However, after some tests on a reference path given at the center of the front axle, it clearly appears that the Stanley controller turns too late. In fact, the correction on the front error enables to anticipate curves as Stanley is not equipped of a feedforward term (for example taking into account the curvature of the path).

Tuning of the controllers

The tuning complexity of the different controllers varies. While tuning the KBCF and the Stanley controller are quite straightforward, the tuning of the pure-pursuit controller appears to be a challenging task.

For the KBFC controller, in order to obtain the fastest response without oscillations for the error given by Equation (5.15), the gains K_P and K_D are linked by Equation (5.30) corresponding to a damping ratio of 1. In the case of noise measurements and delays, K_D can be chosen a little bigger to avoid oscillations.

$$K_D = 2\sqrt{K_P} \tag{5.30}$$

Thus, there is only one parameter to tune, say K_P. However, the system will converge in a given distance that does not depend on time, only on K_P. This can cause some smoothness and even stability problems as for a human driver, the convergence time depends on the speed of the vehicle: the smaller the speed is, the more the steering angle is turned. Therefore, it is better to tune different K_P depending on the range of speeds. Equation (5.15) becomes then a Linear Parameter Varying (LPV) system as K_P and K_D varies. After several trials, we chose $K_P = 0.035$ and $K_D = 0.37$ regardless of the speed in our experiments.

The tuning of the Stanley controller depends only on one gain k that is a trade-off between accuracy (high values) and stability (small values). We chose $k = 0.75$ for the experimentations.

Finally, the tuning of the pure-pursuit depends only on the look-ahead distance L_P. However, this parameter depends not only on the speed of the vehicle but also on the scenario. A larger preview distance is better for higher speed in order to be more stable. However, a smaller preview distance is expected in curves in order to avoid cutting corners. In our experimentations, we chose to tune the preview distance as function of the speed as shown in Figure 5.12.

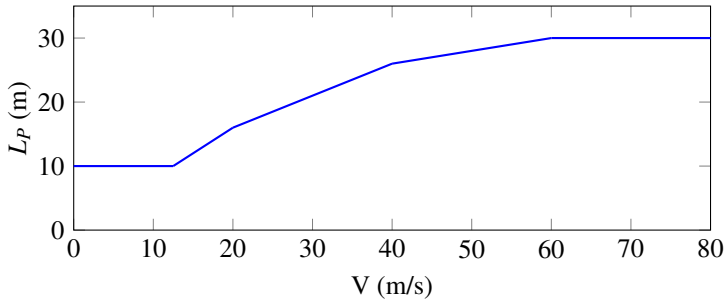


Figure 5.12 – Preview distance of the pure-pursuit controller for different speeds.

Estimation of the curvature and its derivative

One of the main difficulties to apply the KBCF controller is to estimate the curvature κ of the reference path and its derivative with respect to time $\frac{d\kappa}{dt}$.

In order to estimate the curvature κ, we used the expression given by Equation (5.31) where x and y are two functions of the curvilinear abscissa s.

$$\kappa = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{3/2}} \tag{5.31}$$

To obtain x and y, a polynomial function is fit on the waypoints as illustrated by Equation (5.32) where a curvilinear abscissa is associated to each waypoint i as described by Equation (5.33).

$$x = \text{polyn\_fit}(s^r, \mathrm{X}^r) \tag{5.32a}$$
$$y = \text{polyn\_fit}(s^r, \mathrm{Y}^r) \tag{5.32b}$$

$$s_i^r = \begin{cases} 0, & \text{if } i = 1 \\ s_{i-1}^r + \sqrt{(\mathrm{X}_i^r - \mathrm{X}_{i-1}^r)^2 + (\mathrm{Y}_i^r - \mathrm{Y}_{i-1}^r)^2}, & \text{otherwise} \end{cases} \tag{5.33}$$

More precisely, for each waypoint $(\mathrm{X}_{i_0}^r, \mathrm{Y}_{i_0}^r)$, we fit a polynomial of order p to a list of $2m+1$ waypoints $\{(\mathrm{X}_i^r, \mathrm{Y}_i^r)\}_{i \in \{i_0-m, i_0+m\}}$. The order of the polynomial and the number of reference points must be chosen carefully in order to avoid big variation of the shape of the polynomial. Therefore, p should be chosen relatively small but at least equal to 2 as the computation of the curvature requires the second derivative of the polynomial, and the number of points should be relatively large (at least 5). In our case, we set $p = 3$ and $m = 50$ (corresponding to approximately 5m on each side of the closest waypoint).

In order to estimate the curvature $\frac{d\kappa}{dt}$, we used a first-order ALIEN filter, see Equation (5.34), that will be presented in chapter 6 (see also Appendix C.2). However, any estimator of the derivative of a signal can be used.

$$\widehat{\mathrm{F}}_{filt} = -\frac{6}{\mathrm{T}^3} \int_0^{\mathrm{T}} (\mathrm{T} - 2\tau)\, y(\tau)\, \mathrm{d}\tau \tag{5.34}$$

Filtering of the control command

All the controllers output at each time step a constant control signal δ. Therefore, the control signal is piecewise continuous. In order to mitigate the impact of big control variation, we used the same filter for all the controllers, given by Equation (5.35): the steering angle sent to the vehicle is a ponderation between the new value obtained by the control law δ and the value applied at the previous time step δ_{old}. We set the ponderation factor to $\lambda = 0.8$.

$$\delta = (1-\lambda)\delta_{old} + \lambda\delta \tag{5.35}$$

5.3.4 Comparison of lateral controllers

Experimental tests were run on the Shanghai Jiao-Tong University campus, that is an open road environment. Thus for safety reasons, the speed of the vehicle is managed by a human driver. However, the latter tried to keep the same speed profile from one experiment to another despite the traffic.

In order to compare the different controllers, all the plots are given as a function of the curvilinear abscissa s traveled. Moreover, the latter are readjusted such that a given value s corresponds to the same position on the reference track for each controller as the experiments did not start from exactly the same position. All the lateral and heading angle errors are computed at the center of the rear axle even in the case of the Stanley controller. Therefore, a direct comparison between controllers can be made.

The trajectories obtained for each controller are displayed in Figure 5.13: we observe that they were all able to track the reference path without any problems. Figure 5.14 and Figure 5.15 are zooms of respectively the sharp turn on the top right of Figure 5.13 and the long turn on the bottom

right of Figure 5.13. These parts of the reference path are probably the most challenging ones: the first case requires a good anticipation and accuracy due to the strong curvature and the small space margin available; the second one requires a good trade-off between accuracy (the vehicle should at least stay in its lane) and smoothness as the turn is long. The video recordings for each controller are available at https://youtu.be/sgUxGhGDS8I.

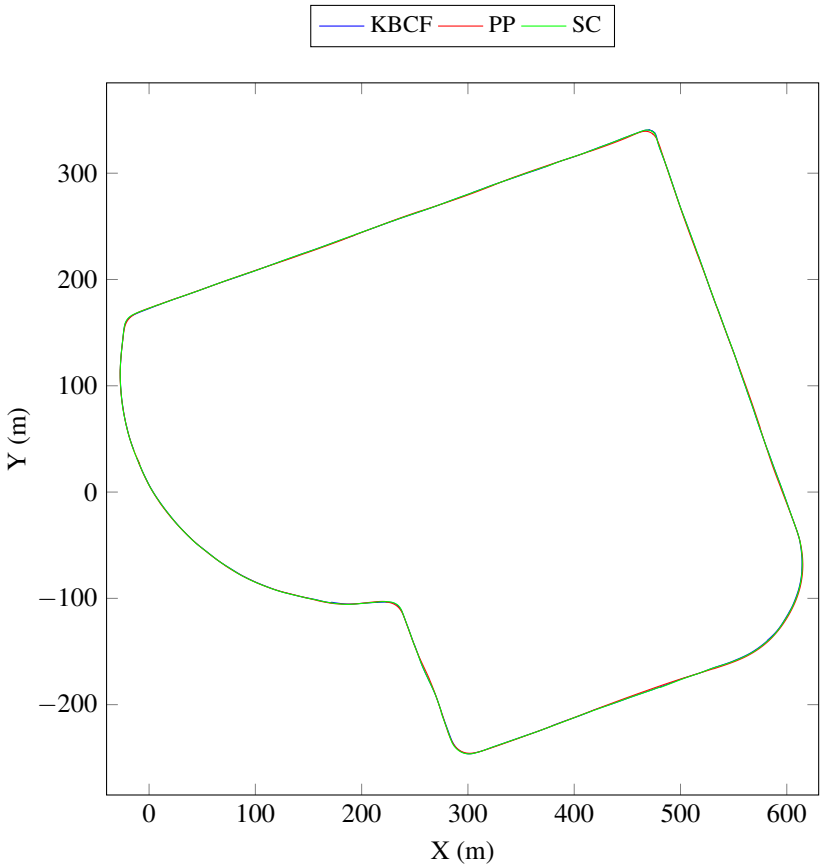


Figure 5.13 – Trajectories obtained with the different controllers.

The speed profiles are displayed in Figure 5.16: despite the traffic and the regulation performed by a human driver, they all look similar making a comparison of the results possible. The differences are mainly due to traffic: for example, the KBCF in blue is slightly slower at the beginning due to a bus coming on the opposite lane, and the Stanley controller is slower around $s = 1700$m due to some pedestrians crossing the road.

The lateral errors $\tilde{y}$ at the center of the rear axle are shown in Figure 5.17. Interestingly, we observe that the KBFC is very accurate in general as the lateral error remains always between -0.35 and 0.30m except in the sharp turns where the lateral error can be big. In particular, in the first sharp turn (see Figure 5.14), the lateral error goes up to $+1.32$m. This might be caused by a bad estimation of the curvature which is responsible of the feedforward steering of the vehicle, caused for example by the fitting of polynomials. On the contrary, the pure-pursuit has a poor accuracy: the lateral error makes large oscillations between -1.03m and $+0.81$m. However it gives relatively good results in the sharp turns due to its geometrical approach for reaching a preview point on the reference path. The lateral error is large in the long turn displayed in Figure 5.15 (up to $+1.17$m) meaning that the vehicle almost overshoots on the opposite lane. This can lead to dangerous situations if a vehicle is coming in the opposite direction. Finally, the performance of the Stanley controller in terms of lateral error accuracy is between the KBCF and the PP controllers, both in the sharp turns, the long turns and the other situations. It varies between -0.2m and 0.79 in general, and up to -0.80m and $+1.05$ in the sharp turns. The main characteristics of the lateral errors for each controller, such as the root mean square error (RMSE), the average error, the standard

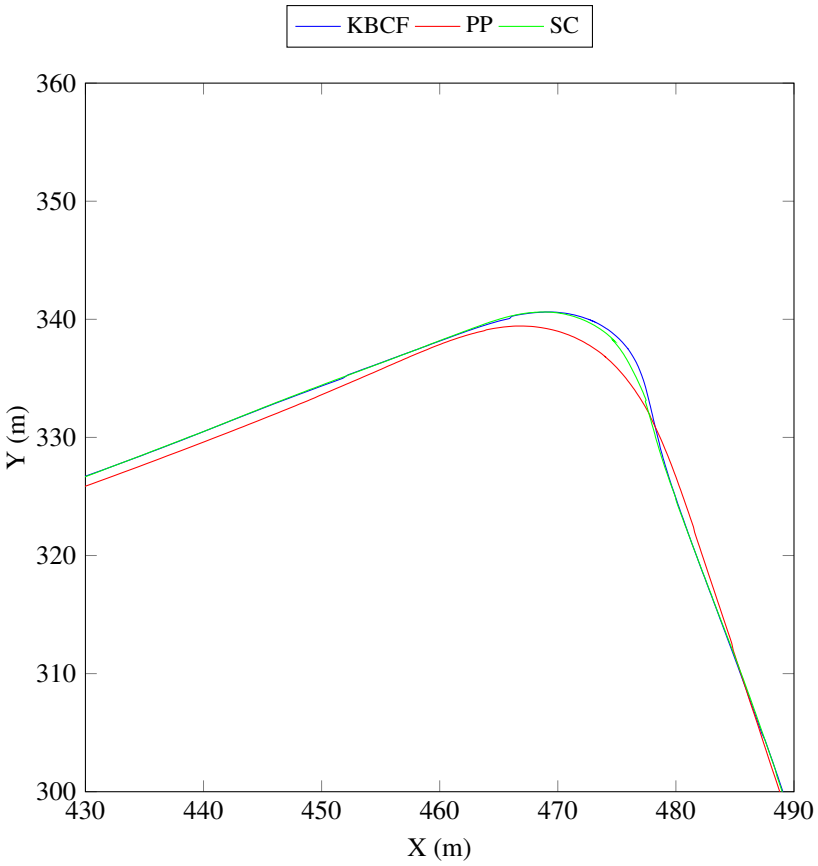


Figure 5.14 – Zoom on the sharp turn.

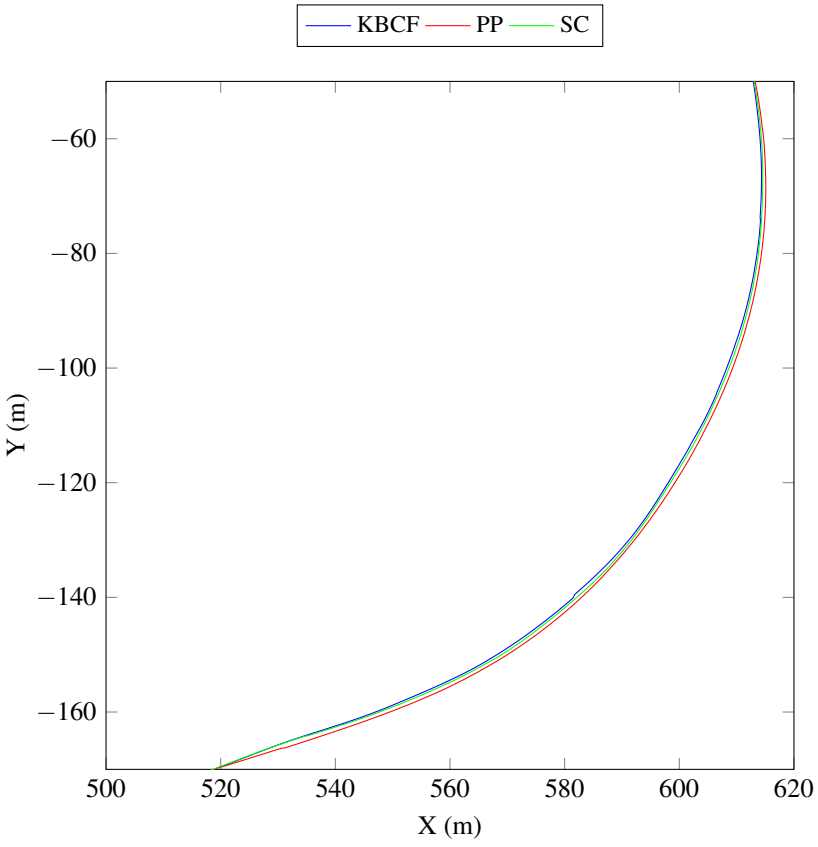


Figure 5.15 – Zoom on the long turn.

deviation of the error, the minimum and the maximum lateral errors are given in Table 5.2.

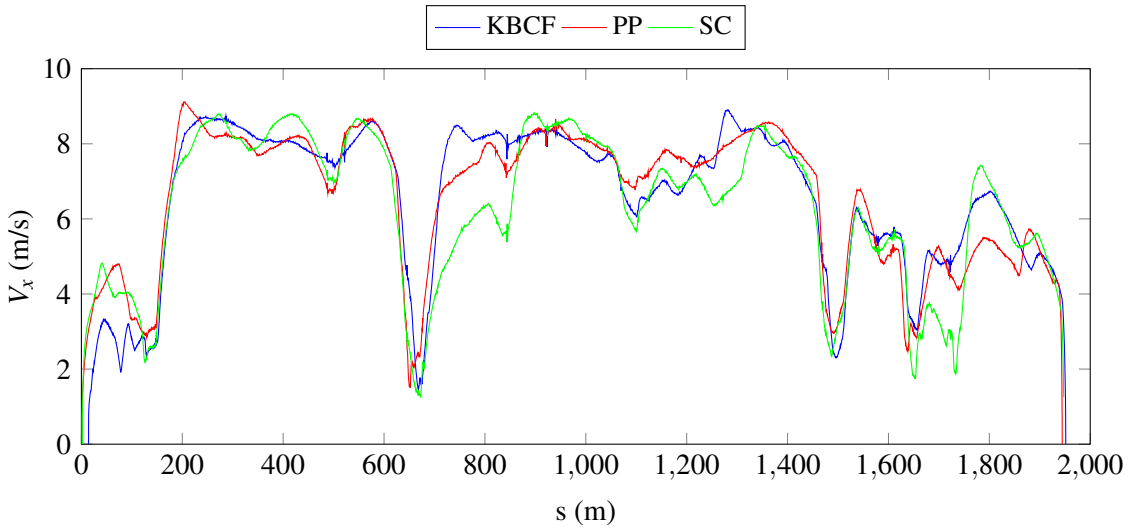


Figure 5.16 – Speed profiles of the vehicle for each controller.

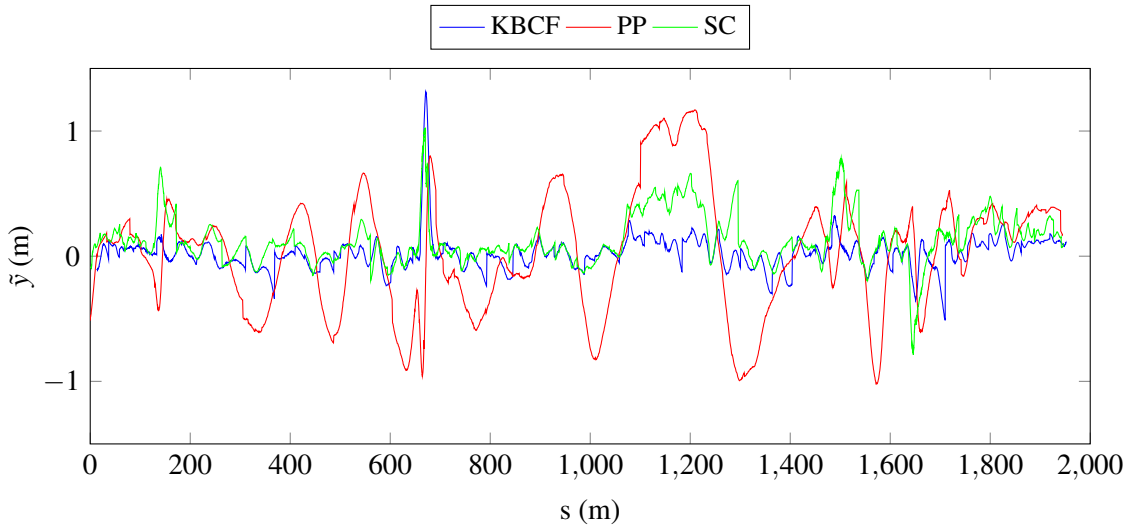


Figure 5.17 – Lateral error as a function of the curvilinear abscissa for each controller.

	RMSE	Mean	Std	Min	Max
KBCF	0.20	0.05	0.19	-0.52	1.32
PP	0.46	0.01	0.46	-1.02	1.17
SC	0.27	0.14	0.24	-0.79	1.04

Table 5.2 – Comparison of the lateral errors (m).

Finally, the distribution of the absolute value of the lateral error is given in Figure 5.18. We observe that the absolute value of the lateral error $|\tilde{y}|$ exceeds 0.3m only 4.38% of the time with the KBCF, 21.43% with a SC but up to 50.25% with the PP.

Similarly, we output the heading angle error at the center of the rear axle in Figure 5.19. We observe that the KBCF and the Stanley controllers give similar results: they remain respectively between $[-0.028; +0.026]$rad and $[-0.067; +0.043]$rad except in the sharp turns. In particular, in the sharp turn presented in Figure 5.14, the heading angle error reaches $[-0.132; +0.156]$rad for the primer case and $[-0.105; +0.093]$rad for the latter. The PP oscillates less than the two others but

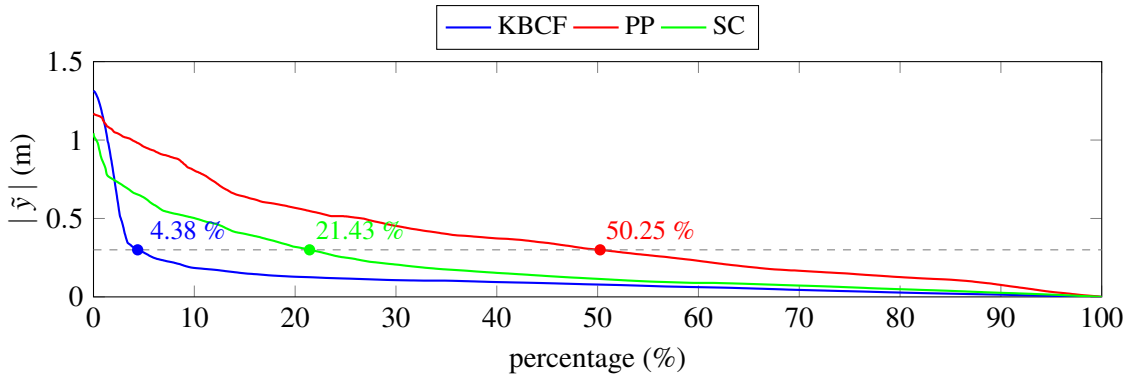


Figure 5.18 – Distribution of the absolute value of the lateral error for each controller.

its amplitude is bigger. Therefore, it remains between $[-0.048; +0.087]$rad except during turns. In the sharp turn presented in Figure 5.14, the heading angle error reaches $[-0.101; +0.216]$rad. Note that due to the noise measurements of the IMU, the precision of the heading angle measurement is estimated to $0.2° = 0.0035$rad after Kalman filtering. The main characteristics of the heading angle errors are given in Table 5.3 for each controller.

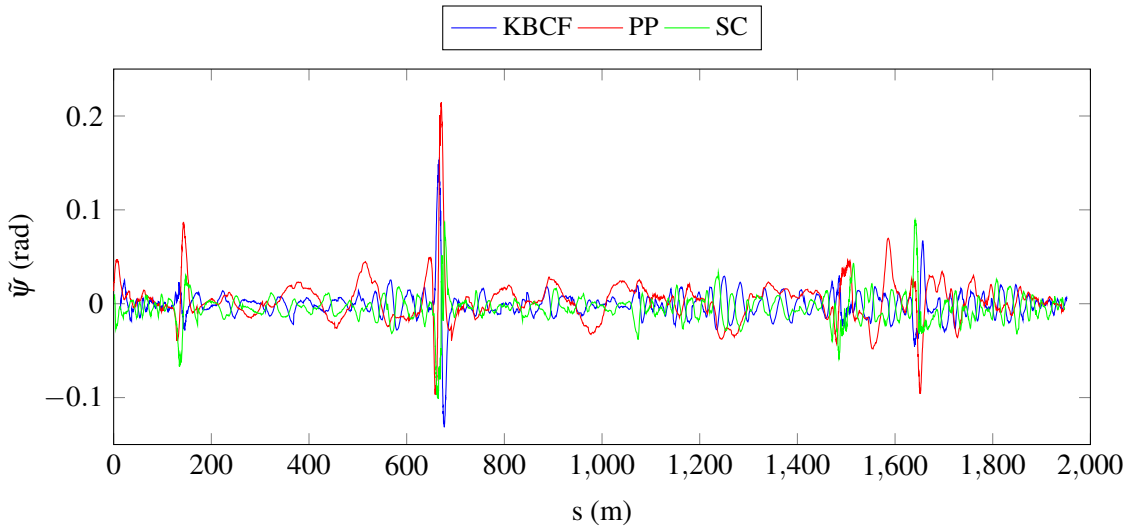


Figure 5.19 – Heading angle error as a function of the curvilinear abscissa for each controller.

	RMSE	Mean	Std	Min	Max
KBCF	0.023	-0.0007	0.023	-0.132	0.156
PP	0.030	0.005	0.029	-0.101	0.216
SC	0.021	-0.006	0.020	-0.105	0.093

Table 5.3 – Comparison of the angular errors (rad).

Finally, the distribution of the absolute value of the heading angle error is given in Figure 5.20. For a limit value of 0.05rad, the results are quite similar: the absolute value of the heading angle error $|\tilde{\psi}|$ exceeds 0.05rad 3.63% of the time with the KBCF, 4.50% with a SC and 4.75% for the PP. On this plot, we clearly observe that the PP controller is less accurate that the two others.

The accuracy of the controller is very important in order to achieve a desired performance. In particular, a good accuracy improves the safety of the vehicle as it will track the desired safe reference trajectory with a great precision. However, the stability and smoothness of the drive should also be taken into account. They are not only responsible for the comfort of the passenger but also for the safety of the vehicle: if the control starts to oscillate too much, the vehicle model on which

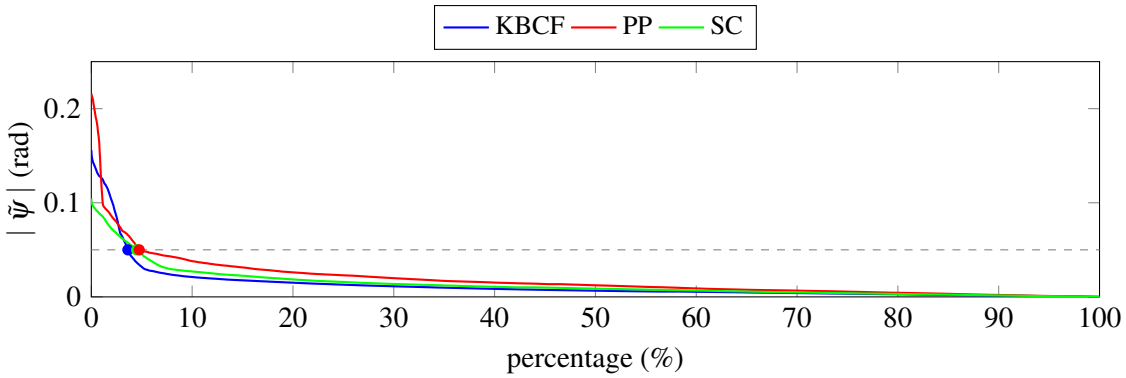


Figure 5.20 – Distribution of the absolute value of the heading angle error for each controller.

stability has been proven might not be valid anymore; dynamic phenomena such as slip or skid might appear for example. However, in control theory, it is well known that improving the accuracy decreases the stability and conversely. This is known as the *stability-accuracy dilemma*. Therefore, we displayed in Figure 5.21 the steering angle applied at the front wheels for each controller.

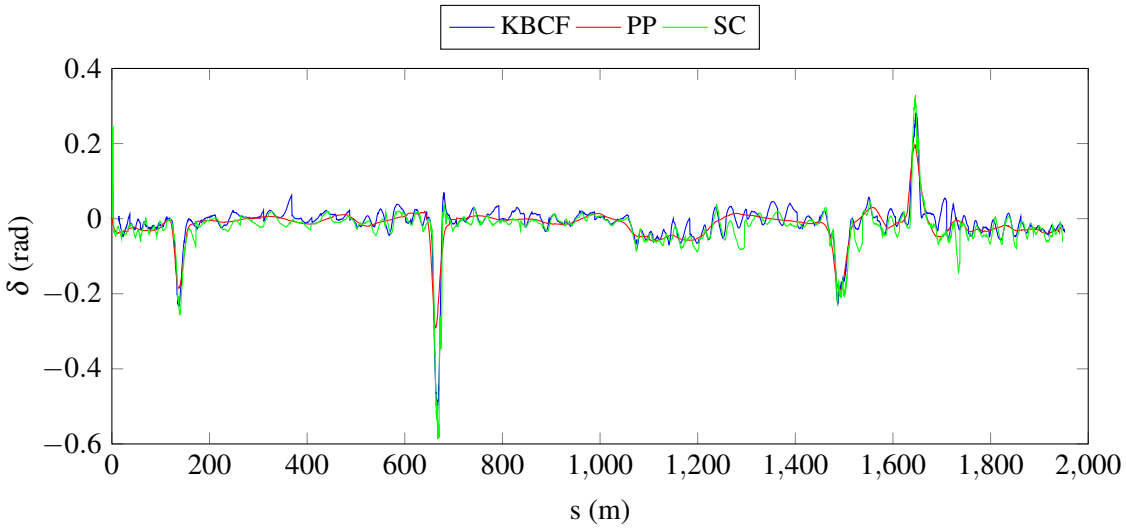


Figure 5.21 – Comparison of the steering angle δ for each controller.

Clearly, it appears that the PP controllers provides a very smooth drive as the steering angle does almost not oscillate at all during the drive. In the KBCF and Stanley controllers, small oscillations of amplitude between 0.01rad and 0.02rad occurs constantly, even when the reference path is a straight line. These oscillations can be perceived by the passengers and cause him some inconveniences if they are too large. Moreover, we observe that the peak values during turns are smaller for the PP than in the other cases although the tracking accuracy is better. This is amplified during sharp turns: for example, a zoom on the steering angles during the sharp turn presented in Figure 5.14 is shown in Figure 5.22. We observe that the peak value of the PP controller corresponds to −0.29rad while for the KBCF and the SC, it corresponds respectively to −0.51rad and −0.59rad. However, the PP controller reaches −0.05rad already at $s = 650.9$m while the KBCF and the SC reaches the same value at respectively $s = 655.5$m and $s = 656.8$m thus more than 4m later!

In the case of the KBCF controller, the feedforward control depends mostly on the curvature κ of the reference track and its time derivative $\frac{d\kappa}{dt}$. These are displayed respectively in Figure 5.23 and 5.24. In the latter, we observe constantly some “jumps" to zero, especially in the curves. These are due to the slow speed of the vehicle while the frequency of measurement is high (100Hz). Therefore, between two acquisitions, the reference waypoint is often the same leading to no variation of the curvature.

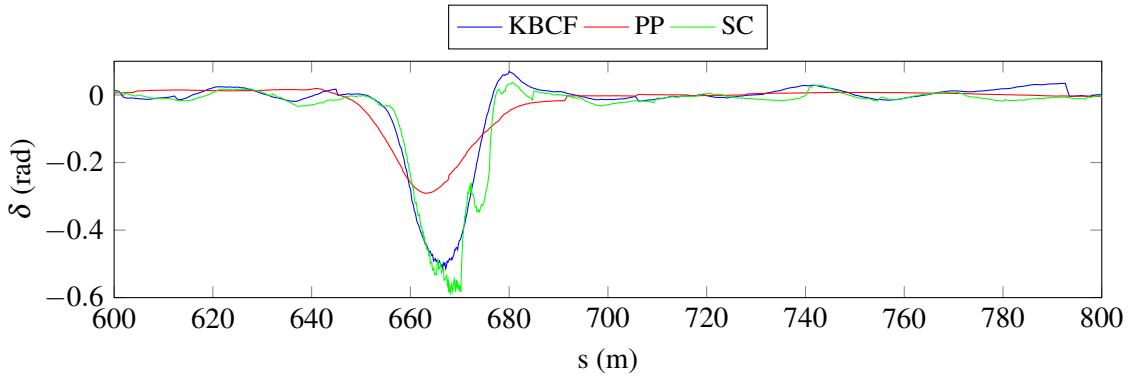


Figure 5.22 – Comparison of the steering angle obtained at the front wheel for each controller in the sharp turn.

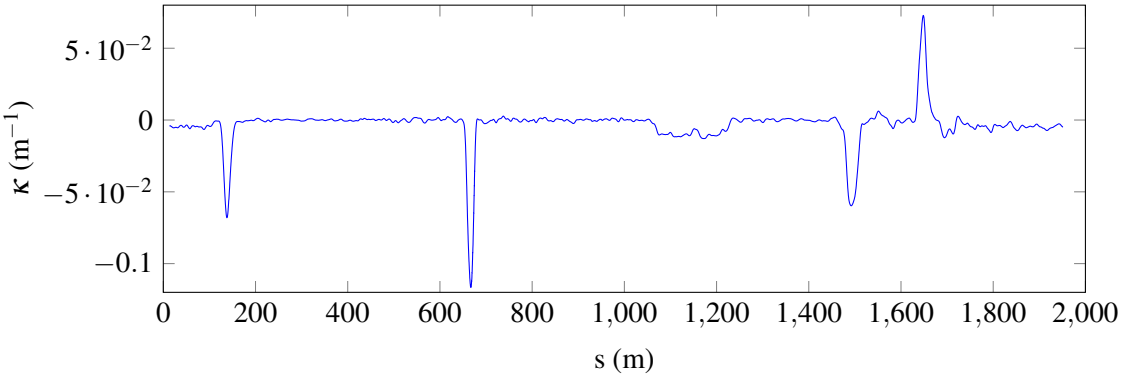


Figure 5.23 – Estimation of the curvature κ of the reference path for the KBCF controller.

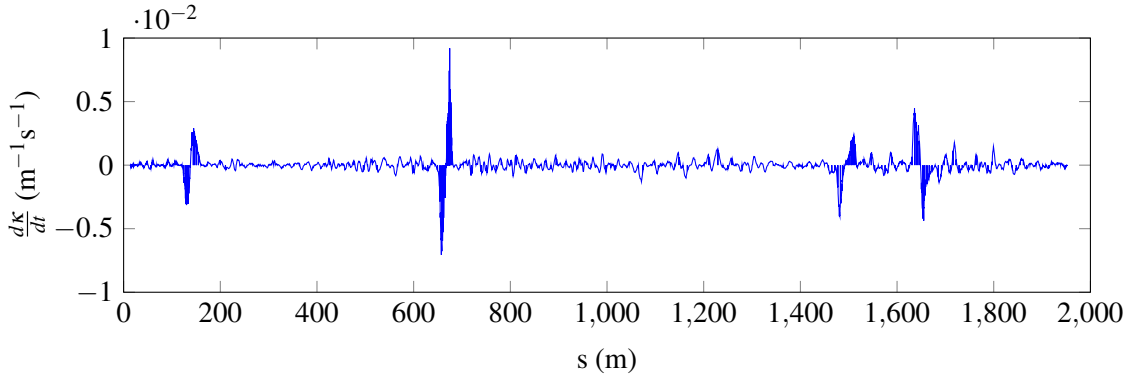


Figure 5.24 – Estimation of the derivative of the curvature $\frac{d\kappa}{dt}$ of the reference path for the KBCF controller.

5.3.5 Conclusion

The driving task can be divided into two levels [Donges, 1978]: an *anticipation* level responsible for the feedforward or open-loop control, and a *stabilization* level responsible for the feedback or closed-loop control. In the primer level, the driver takes into account the future shape of the road while in the second, he corrects the actual deviations from the reference path.

Therefore, a good lateral controller should be composed of both an anticipation and a stabilization term. *Anticipation* can be done either using a preview distance such as in the case of the pure-pursuit or Stanley controller or using information on the road geometry, typically the road curvature as in the case of the non-linear controller based on the kinematic bicycle model (KBCF) or the controller based on the linearized dynamic bicycle model (LDBM). *Stabilization* is done by correcting the lateral error and/or the heading angle error so they converge to 0: the Stanley controller corrects the angular error and the front lateral error, the KBCF corrects both the angular error and the lateral error at the center of the rear axle through the gain K_P and K_D and the linearized dynamic bicycle model corrects both errors and their derivative through the gains k_1, k_2, k_3 and k_4. Only the pure-pursuit controller does not dispose directly of a stabilization term. However, this can be done by adding a PI controller on the lateral error for example.

The choice of a controller is necessarily a trade-off between accuracy and stability/smoothness. This choice depends on the situation encountered: while a high precision is required for narrow roads for examples, smoothness is more important in the case of highways. Therefore, we believe that the “best and universal" controller does not exist. Instead, there exists a range of controllers suited for different cases. It becomes then important to switch between them according to the driving situation. However, the resulting switched system should remain stable which is not necessarily the case even though each individual controller is stable (see [Branicky, 1998] for example).

In order to guide the reader in a choice of a controller, a summary of the one tested in section 5.3 together with a PID controller is given in Table 5.4. This table gives the main assets and drawbacks of each controller, as well as the underlying assumptions on the reference path and the model.

Controller	Tuning	Accuracy	Smoothness	Ref. path	max$\lvert a_y \rvert$	Model
PID	-	-	+	C^0	?	*
PP	-	-	+	any	$0.5\mu g$	kbm
SC	+	(+)	(+)	C^0	?	dbm
KBCF	+	+	-	C^2/C^3	$0.5\mu g$	kbm

- drawback ; + asset ; ? unknown ; * depends on the model used to tune it.

Table 5.4 – Summary of lateral controllers.

5.4 Coupled longitudinal and lateral control

In most driving situations, the coupling between longitudinal and lateral dynamics can be neglected. As a matter of fact, drivers either accelerate/decelerate the vehicle or turns, but do not both at the same time. However, in some situations such as aggressive driving, emergency maneuvers or low friction coefficient roads, the coupling between longitudinal and lateral dynamics should be considered to ensure the stability of the vehicle. This is in particular due to the friction circle presented in section 2.2: when the longitudinal tire force F_{xp} and the lateral tire force F_{yp} are such that $\sqrt{F_{xp}^2 + F_{yp}^2} = \mu F_z$, the tire cannot generate any extra force and tends to prioritize longitudinal over lateral dynamics.

The overall vehicle dynamics beeing highly non-linear, deriving a controller that takes into account the coupling between longitudinal and lateral dynamics is very complicated. In the literature, most of the work have been focusing on using Model Predictive Control with complex

non-linear models. [Attia et al., 2012a] used a non-linear MPC based on a dynamic bicycle model, with the equation of the wheel dynamics. The longitudinal tire model is a Burckhardt model while the lateral tire model is a modified Pacejka magic formula. [Falcone et al., 2007b] used both a non-linear MPC and a linearized MPC based on a dynamic bicycle model where load transfer between the front and the rear axles are taken into account. Longitudinal and lateral Pacejka tire model are used where the longitudinal slip ratio is assumed to be constant over the prediction horizon. This assumption is necessary due to the strong non-linearities in the expression of the longitudinal slip ratio but is quite restrictive as the time constant of the longitudinal slip ratio is very small compared to the vehicle dynamics. In [Kritayakirana and Gerdes, 2012], a coupled longitudinal and lateral controller have been derived from a dynamic bicycle model using a linear lateral tire model. The longitudinal velocity is determined using a gg-diagram (see chapter 3.4). Similarly, [Goh and Gerdes, 2016] proposed a similar control law but uses a modified Fiala tire model for computing the lateral tire forces. However, using MPC with complex vehicle and tire dynamic models is computationally expensive. Thus it cannot be computed at high-frequency: the control is in open-loop on a long duration (typically around 50 to 200ms) and is therefore less robust to modeling errors and external disturbances. Moreover, it becomes less reactive to changes of the reference trajectory. [Menhour et al., 2011a] proposed a coupled longitudinal and lateral controller based on the flatness property of the dynamic bicycle model with a linear tire model which enables to compute the control input at a high frequency in closed-loop.

One perspective to avoid the complexity of the models is to learn them offline using deep learning methods, in particular supervised learning. Some preliminary results were obtained with Guillaume Devineau in [Devineau et al., 2018]: first, we generated a dataset of 43241 instances each containing a three-second trajectory and the corresponding control inputs. The trajectories were obtained by applying the control inputs in the 9 DoF vehicle simulator presented in section 2.3. Then, a Multi-Layer Perceptron and a Convolutionnal Neural Network were trained on the dataset to learn from the trajectories the control that was applied. During the training, the loss function penalized more steering angle errors than wheel torque errors. These artificial neural networks were then used as controllers with some enhancement on a track in simulation and proved their ability to cope with highly coupled maneuvers. One interesting result is that it takes only about 2ms to look-up the longitudinal and lateral control signals to apply given a reference trajectory. Such techniques should be further investigated in future works and might improve drastically the computational time required to solve such problems. However, any proofs of convergence of the proposed approach should not be expected.

Chapter 6

Model-free control

“ You must unlearn what you have learned ”

Master Yoda

Contents

Applying model-based control techniques on an actual system raises several difficulties. First, a model of the plant is necessary. However, obtaining a model can be either too costly due to the required engineering time and hardware, or simply impossible (ex: the brake system of a vehicle). Secondly, a non-linear control law needs to be derived from the model, using techniques such as flatness-based control [Fliess et al., 1995], chained-form systems [Samson, 1993] or by linearizing the system around its operational point. However, this is not always possible. Thirdly, a model contains parameters whose values can be difficult to access in practice. Worse, they might be changing over time due to the aging, the wear or the use of the equipment: in the case of a vehicle for example, the mass changes with the loading and the tire parameters depend on the tire pressure and wear. Finally, ensuring the robustness of the controller to model mismatchs and external perturbations such as wind, slope, road-bank angles or the friction coefficient of the roads in the case of an autonomous vehicle can be challenging. Thus, it is not surprising if Proportional-Integral-Derivative (PID) controllers [Astrom and Hagglund, 1995], which are model-free controllers by excellence as they rely solely on the output error of the system and its derivative, are among the most popular ones in industrial applications[1].

In this chapter, the model-free control (MFC) paradigm introduced in [Fliess and Join, 2013] is presented. Although it has been applied successfully in many applications, among other energy management [Bara et al., 2017] and flapping wings [Chand et al., 2016], we propose some improvements for its implementation on an actual plant, in particular an autonomous vehicle. MFC is able to handle some complex non-linear control problem [Fliess and Join, 2009] at a low computational cost as these control laws can be implemented on cheap and small programmable devices [Join et al., 2013]. Among the main assets of MFC compared to conventional PID controllers, let's mention its ability to reject perturbations naturally, its ability to track a non-constant reference output and its robustness to the aging of the system.

After recalling the fundamentals of the model-free control paradigm in section 6.1, in particular the notion of "intelligent" controllers and ALIEN filters [Fliess and Sira-Ramírez, 2003], we will discuss some implementation details on an actual system in section 6.2 which is our main contribution to this type of controllers. Finally, the MFC paradigm is applied for the longitudinal control of an autonomous vehicle both in simulation and on an actual vehicle in section 6.3.

6.1 Model-free control: a new control paradigm for non-linear systems

6.1.1 The ultralocal model

The principle of the model-free control paradigm introduced in [Fliess and Join, 2013] is to replace a complex non-linear model by an ultralocal model given by Equation (6.1), valid only on a very short time period T (see Appendix C.1 for more details):

$$y^{(\nu)}(t) = \mathrm{F}(t) + \alpha u(t) \tag{6.1}$$

u is the control input, y is the observed output, ν is the order of the system, F represents both the unmodeled dynamics and the disturbances and α is a parameter chosen by the practitioner (the sign of α is supposed to be known). In practice, the order ν is always 1 or 2. When there is no ambiguity, the time dependency will be omitted.

6.1.2 Estimation of $\widehat{\mathrm{F}}$

At each time step $t_k = k\mathrm{T}_s$ where T_s is the sampling time step of the measurements, the dynamics F is estimated from the previous control inputs u applied to the system and outputs y observed. The estimation of F at time t, denoted $\widehat{\mathrm{F}}$, is assumed to remain constant over $[t; t+\mathrm{T}]$. Therefore, using the model-free control approach requires to have a high measurement frequency compared

[1] https://taskforce.ifac-control.org/industry-committee/legacy-materials-pilot-industry-committee-2014-2017/meeting-materials/meeting-presentation-17-june-2015/view, visited on March 6th 2017.

to the time constant of the system one wants to control. A direct method for estimating $\widehat{\mathrm{F}}_k$ at time t_k is given by Equation (6.2):

$$\widehat{\mathrm{F}}_k = y_k^{(\nu)} - \alpha u_{k-1} \tag{6.2}$$

However, measurements are usually noisy. Therefore, using a filter is highly recommended, such as the ALIEN[2] filters introduced by [Fliess and Sira-Ramírez, 2003]. For a first-order ($\nu = 1$) or a second-order ($\nu = 2$) ultralocal model, one obtains respectively Equations (6.3) and (6.4), where T denotes the time window of the filter. The computation details can be found in Appendix C.2.

$$\text{if } \nu = 1: \quad \widehat{\mathrm{F}} = -\frac{6}{\mathrm{T}^3}\int_0^{\mathrm{T}} \left[(\mathrm{T}-2\tau)\,y(\tau) + \alpha\tau(\mathrm{T}-\tau)\,u(\tau))\right] \mathrm{d}\tau \tag{6.3}$$

$$\text{if } \nu = 2: \quad \widehat{\mathrm{F}} = -\frac{5!}{2\mathrm{T}^5}\int_0^{\mathrm{T}} \left[(-\mathrm{T}^2 + 6\mathrm{T}\tau - 6\tau^2)\,y(\tau) + \frac{\alpha}{2}\tau^2(\mathrm{T}-\tau)^2 u(\tau))\right] \mathrm{d}\tau \tag{6.4}$$

6.1.3 Intelligent controllers

The "intelligent" controllers [Fliess and Join, 2008] have been introduced within the model-free control paradigm. They can be written in the form of Equation (6.5):

$$u = -\frac{\widehat{\mathrm{F}} - \dot{y}_r}{\alpha} - \mathscr{K}(e) \tag{6.5}$$

where $e = y - y_r$ is the tracking error.

There exist commonly four types of "intelligent" controllers: intelligent-Proportional (i-P) and intelligent-Proportional-Integral (i-PI) for a first-order system ($\nu = 1$); intelligent-Proportional-Derivative (i-PD) and intelligent-Proportional-Integral-Derivative (i-PID) for a second-order system ($\nu = 2$). They are composed of a feedforward term $\frac{y_r^{(\nu)}}{\alpha}$, an ultralocal term $-\frac{\widehat{\mathrm{F}}}{\alpha}$ and a feedback term $-\mathscr{K}(e)$, as illustrated by Equations (6.6a-6.6d). The naming depends on the form of the function $\mathscr{K}$ which can be either a P, PI, PD or PID controller.

$$\textbf{i-P:} \quad u = -\frac{\widehat{\mathrm{F}} - \dot{y}_r}{\alpha} - \mathrm{K_P} e \tag{6.6a}$$

$$\textbf{i-PI:} \quad u = -\frac{\widehat{\mathrm{F}} - \dot{y}_r}{\alpha} - \mathrm{K_P} e - \mathrm{K_I}\int e\,dt \tag{6.6b}$$

$$\textbf{i-PD:} \quad u = -\frac{\widehat{\mathrm{F}} - \ddot{y}_r}{\alpha} - \mathrm{K_P} e - \mathrm{K_D}\dot{e} \tag{6.6c}$$

$$\textbf{i-PID:} \quad u = -\frac{\widehat{\mathrm{F}} - \ddot{y}_r}{\alpha} - \mathrm{K_P} e - \mathrm{K_D}\dot{e} - \mathrm{K_I}\int e\,dt \tag{6.6d}$$

$\mathrm{K_P}$, $\mathrm{K_I}$ and $\mathrm{K_D}$ denote respectively the proportional, the integral and the derivative gains.

Remark 14. *Note that the feedforward term differs between Equations (6.6a-6.6b) and Equations (6.6c-6.6d) due to the order ν of the system.*

Remark 15. *The reference signal y_r must be at least a C^1 signal in the case of a first-order system and a C^2 signal in the case of a second-order system. This can be achieve using a low-pass filter on the reference signal which also enables to compute its first-order and/or second-order derivative.*

For a first-order system, combining Equations (6.1) and (6.5), leads to:

$$\dot{e}(t) = -\alpha\mathscr{K}(e(t)) + (\mathrm{F}(t) - \widehat{\mathrm{F}}(t)) \tag{6.7}$$

[2] ALgebra for Numerical Identification and Estimation

Therefore, if the estimation of $\widehat{F}$ is "good", *i.e.* $(F - \widehat{F}) \approx 0$, i-P or i-PI controllers ensure the asymptotic stability of the closed-loop system with $\alpha > 0$, $K_P > 0$ and $K_I > 0$. For example, for an i-P controller, Equation (6.7) becomes:

$$\dot{e}(t) = -\alpha K_P e(t) \tag{6.8}$$

For a second-order system, combining Equations (6.1) and (6.5), leads to:

$$\ddot{e}(t) = -\alpha \mathcal{K}(e(t)) + (F(t) - \widehat{F}(t)) \tag{6.9}$$

Again, if the estimation of $\widehat{F}$ is "good", i-PD or i-PID controllers ensure the asymptotic stability of the closed-loop system with $\alpha > 0$, $K_P > 0$, $K_D > 0$ and $K_I > 0$. For example, for an i-PD controller, Equation (6.9) becomes:

$$\ddot{e}(t) = -\alpha K_D \dot{e}(t) - \alpha K_P e(t) \tag{6.10}$$

Therefore, unlike conventional PID controllers, "intelligent" controllers provide a theoretical guarantee of the asymptotic stability of the system in the case of no estimation error, even in the transient phases where the reference signal is not constant. This is why they are called "intelligent".

In practice, an i-P (resp. i-PD) controller is sufficient for a first-order (resp. second-order) system. The disturbances are implicitely embedded within the unknown term F in the model given by Equation (6.1) and are therefore estimated by the ALIEN filter. Thus, they are rejected by the ultralocal term $-\widehat{F}/\alpha$ of the control law in Equation (6.5). The feedback term $\mathcal{K}(e)$ is mostly used to overcome the error due to initial conditions and estimation errors of F due to the dynamics of the filter.

In [d'Andréa et al., 2010], the authors have compared "intelligent" controllers with conventional PID controllers given by the form of Equation (6.11). It turns out that in their discretized version with a constant sampling time step T_s, i-P, i-PI, i-PD and i-PID controllers are equivalent respectively to PI, PI^2, PID and PI^{2}D controllers. This could explain why PID controllers work well in practice. Table 6.1 gives the correspondance between "intelligent" and conventional PID controllers[3]. The Equation of the PI^{2}D controller is given by Equation (6.11).

$$u(t) = -k_d \dot{e}(t) - k_p e(t) - k_i \int_0^t e(\tau) d\tau - k_{ii} \int_0^t \int_0^\tau e(\tau') d\tau' d\tau \tag{6.11}$$

		i-P	i-PD	i-PI	i- PID
PI	k_p	$1/(\alpha T_s)$			
	k_i	K_P/T_s			
PID	k_p		K_D/T_s		
	k_i		K_P/T_s		
	k_d		$1/(\alpha T_s)$		
PI^2	k_p			$1/(\alpha T_s)$	
	k_i			K_P/T_s	
	k_{ii}			K_I/T_s	
PI^{2}D	k_p				K_D/T_s
	k_i				K_P/T_s
	k_{ii}				K_I/T_s
	k_d				$1/(\alpha T_s)$

Table 6.1 – Comparison between discrete conventional and "intelligent" controllers [d'Andréa et al., 2010].

[3] Note that the gains K_P, K_D and K_I were divided by α in [d'Andréa et al., 2010] but not here.

6.2 Implementation on an actual system

In order to apply the model-free control paradigm on an actual system, several difficulties need to be overcome. In particular, one of the major assumption in the model-free control theory is that the estimation error $F - \widehat{F}$ is zero (or almost zero). Therefore, we need to ensure that this assumption remains valid at all time.

However, as mentioned in subsection 6.1.2, filters are necessary in order to mitigate the noise of the measurements, especially to compute derivative signals such as $y^{(v)}$ from the successive measurements of y. These filters have their own dynamics which lead to an estimation error during transient phases. Moreover, other phenomena such as delays can play a similar role. In the case of ALIEN filters, we will show that a bad numerical implementation of the integral given by Equations (6.3) and (6.4) leads also to estimation errors at steady-state.

Therefore, this section will present on how to implement a model-free controller based on ALIEN filters in order to get the best possible results. Subsection 6.2.1 will discuss how to implement numerically the ALIEN filters in order to obtain a perfect estimation at steady-state. Then, in subsection 6.2.2, a tuning procedure for setting the parameter α is introduced. We will show that this parameter has a strong impact on the performances of the MFC, in particular during transient phases. Finally, as it is never possible to guarantee a perfect estimation of F, we will make a stability analysis of the system when the estimation error is non-zero in section 6.2.3.

6.2.1 Numerical quadrature for ALIEN filters

One of the main difficulty for applying the MFC paradigm using ALIEN filters on an actual system is to compute numerically the expression of the filter given by Equation (6.3) for a first-order system (respectively Equation (6.4) for a second-order system). First of all, it is important to provide the actual control u that was applied on the system and not the one send by the controller as they might differ due to filters or limitations on the actuators.

Order of the quadrature

Theorem 2. *In order to have a perfect estimation of F at steady-state, the numerical quadrature to compute Equation (6.3) should be at least of order 2.*

Proof. Assume that the system has reach its steady-state, denoted by the subscript ∞. In this case, Equation (6.1) becomes:

$$F_\infty = -\alpha u_\infty \tag{6.12}$$

and the corresponding expression of the ALIEN filter given by Equation (6.3) becomes:

$$\widehat{F} = -\frac{6}{T^3}\int_0^T P_2(\tau)\,d\tau \tag{6.13}$$

where $P_2(\tau) = Ty_\infty + (-2y_\infty + \alpha T u_\infty)\tau - \alpha u_\infty \tau^2$ is a second-order polynomial in τ.

The *order of accuracy* n of a quadrature is defined as the biggest integer value such that the numerical approximation is correct for any polynomial of degree less or equal to n (see for example [Dahlquist and Björck, 2008]), which concludes the proof. □

Theorem 3. *In order to have a perfect estimation of F at steady-state, the numerical quadrature to compute Equation (6.4) should be at least of order 4.*

Proof. Similarly to the proof of Theorem 2, we obtain at steady-state for a second-order system:

$$\widehat{F} = -\frac{5!}{2T^5}\int_0^T P_4(\tau)\,d\tau \tag{6.14}$$

where $P_4(\tau) = -T^2 y_\infty + 6Ty_\infty\tau + \left(-6y_\infty + \frac{\alpha u_\infty}{2}T^2\right)\tau^2 - \alpha u_\infty T\tau^3 + \frac{\alpha u_\infty}{2}\tau^4$ is a fourth-order polynomial in τ, which concludes the proof. □

Thus, a conventional numerical approximation such as the trapezoidal rule which is of order 1 is not sufficient: in this case, for a first-order system, we have an estimation error on F of $+\frac{1}{2}\left(\frac{\mathrm{T}_s}{\mathrm{T}}\right)^2\mathrm{F}$ (see Appendix C.3). However, choosing $\mathrm{T} >> \mathrm{T}_s$ enables to improve the accuracy but T has to remain "small" in order for the ultralocal model to remain valid.

Therefore, Equation (6.3) should be approximated using one of the two Simpson's rules given by Equations (6.15) and (6.16), which are both of order 3:

- **Simpson's 1/3 rule** (or three-point Newton-Cotes quadrature rule)

$$\int_a^b f(x)dx \quad \approx \quad \frac{b-a}{6}\left(f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right) \tag{6.15}$$

- **Simpson's 3/8 rule** (or four-point Newton-Cotes quadrature rule)

$$\int_a^b f(x)dx \quad \approx \quad \frac{b-a}{8}\left(f(a)+3f\left(\frac{a+b}{3}\right)+3f\left(\frac{2(a+b)}{3}\right)+f(b)\right) \tag{6.16}$$

Newton-Cotes formulas generalize the Simpson rule for polynomial interpolation of arbitrary degree . For example, for the numerical quadrature of Equation (6.4), the five-point Newton-Cotes quadrature rule (also named the Boole-Villarceau method) given by Equation (6.17) can be used as it is of order 5. However, in practice, using high-degree polynomials for interpolation can lead to instability. This is referred to as Runge's phenomenon [Schlömilch et al., 1901]. Therefore, cutting the computation of the integral given by Equation (6.3) (respectively Equation (6.4)) into subdivisions where a method of order at least 2 (respectively 4) can be applied is recommended.

- **Boole-Villarceau rule** (or five-point Newton-Cotes quadrature rule)

$$\int_a^b f(x)dx \quad \approx \quad \frac{b-a}{90}\left(7f(a)+32f\left(\frac{a+b}{4}\right)+12f\left(\frac{2(a+b)}{4}\right)+32f\left(\frac{3(a+b)}{4}\right)+7f(b)\right) \tag{6.17}$$

Hence, it is necessary to choose the right order of the numerical quadrature for implementing the ALIEN filters. For that purpose, it is easier to choose the number n_{rep} of repetitions of the scheme given by Equation (6.15) for a first-order system (respectively Equation (6.17) for a second-order system). The time window T is then chosen accordingly: $\mathrm{T} = \mathrm{T}_s(2n_{rep}+1)$ for a first-order system and $\mathrm{T} = \mathrm{T}_s(4n_{rep}+1)$ for a second-order system. This method will be chosen in the rest of this work.

Remark 16. *Note that for a first-order system, if* $\lfloor \mathrm{T}/\mathrm{T}_s \rfloor \geq 2$*, a decomposition of the integral given by Equation (6.3) into subintegrals where Equation (6.15) and (6.16) can be applied is always possible (see demonstration in Appendix C.4).*

Numerical examples

Tables 6.2 to 6.5 present some examples of numerical coefficients of an ALIEN filter, depending on the order ν and the number of repetition of the numerical integration scheme n_{rep}. There are n_y coefficients applied on the observation signal y and n_u on the control signal u. If we denote $t_k = k\mathrm{T}_s$ the actual time of the system, the time integral ALIEN filter is given by Equation (6.18) where $c_{\text{ALIEN}} = [c_{y_{i,i\in[\![1;n_y]\!]}} \,|\, c_{u_{j,j\in[\![1;n_u]\!]}}]$ are the list of coefficients computed by applying the appropriate numerical quadrature.

$$\widehat{\mathrm{F}}_k \quad = \quad \sum_{i=1}^{n_y} c_{y_i}\, y(k+1-i) + \sum_{j=1}^{n_u} c_{u_j}\, u(k-j) \tag{6.18}$$

- **First-order derivative:** In this case, $\nu = 1$ and $n_u = 0$ as we are only estimating the first order derivative of the signal y. The number of coefficients necessary is $n_y = 1+2n_{rep}$. The coefficients are given in Table 6.2.

ν	n_{rep}	n_y	c_{ALIEN}
1	1	3	$[50,\ 0,\ -50]$
1	2	5	$[12.5,\ 25,\ 0,\ -25,\ -12.5]$
1	3	7	$\left[\frac{50}{9},\ \frac{400}{27},\ \frac{100}{27},\ 0,\ \frac{-100}{27},\ \frac{-400}{27},\ \frac{-50}{9}\right]$

Table 6.2 – ALIEN filter coefficients for estimating the first-order derivative of a signal y.

ν	n_{rep}	n_y	c_{ALIEN}
2	1	5	$\left[\frac{8750}{3},\ \frac{-5000}{3},\ -2500,\ \frac{-5000}{3},\ \frac{8750}{3}\right]$
2	2	9	$\left[\frac{4375}{12},\ \frac{6875}{12},\ \frac{-625}{8},\ \frac{-8125}{12},\ \frac{-4375}{12},\ \frac{-8125}{12},\ \frac{-625}{8},\ \frac{6875}{12},\ \frac{4375}{12}\right]$

Table 6.3 – ALIEN filter coefficients for estimating the second-order derivative of a signal y.

- **Second-order derivative:** In this case, $\nu = 2$ and $n_u = 0$ as we are only estimating the second order derivative of the signal y. The number of coefficients necessary is $n_y = 1 + 4n_{rep}$. The coefficients are given in Table 6.3.

- **First-order system:** In this case, $\nu = 1$, $n_y = 1 + 2n_{rep}$ and $n_u = n_y - 1$. The coefficients are given in Table 6.4.

ν	n_{rep}	n_y	n_u	c_{ALIEN}
1	1	3	2	$[50,\ 0,\ -50,\ \vdots\ 0,\ -1]$
1	2	5	4	$[12.5,\ 25,\ 0,\ -25,\ -12.5,\ \vdots\ 0,\ \frac{-3}{8},\ \frac{-1}{4},\ \frac{-3}{8}]$
1	3	7	6	$\left[\frac{50}{9},\ \frac{400}{27},\ \frac{100}{27},\ 0,\ \frac{-100}{27},\ \frac{-400}{27},\ \frac{-50}{9},\ \vdots\ 0,\ \frac{-5}{27},\ \frac{-4}{27},\ \frac{-1}{3},\ \frac{-4}{27},\ \frac{-5}{27}\right]$

Table 6.4 – ALIEN filter coefficients for estimating $\widehat{\text{F}}$ for a first-order system.

- **Second-order system:** In this case, $\nu = 2$, $n_y = 1 + 4n_{rep}$ and $n_u = n_y - 1$. The coefficients are given in Table 6.5.

ν	n_{rep}	n_y	n_u	c_{ALIEN}
2	1	5	4	$\left[\frac{8750}{3},\ \frac{-5000}{3},\ -2500,\ \frac{-5000}{3},\ \frac{8750}{3},\ \vdots\ 0,\ \frac{-3}{8},\ \frac{-1}{4},\ \frac{-3}{8}\right]$
2	2	9	8	$\left[\frac{4375}{12},\ \frac{6875}{12},\ \frac{-625}{8},\ \frac{-8125}{12},\ \frac{-4375}{12},\ \frac{-8125}{12},\ \frac{-625}{8},\ \frac{6875}{12},\ \frac{4375}{12},\ \vdots\ 0,\ \frac{-49}{768},\ \frac{-9}{128},\ \frac{-75}{256},\ \frac{-7}{48},\ \frac{-75}{256},\ \frac{-9}{128},\ \frac{-49}{768}\right]$

Table 6.5 – ALIEN filter coefficients for estimating $\widehat{\text{F}}$ for a second-order system.

6.2.2 Tuning procedure of the model-free controller

Suppose that the non-linear system to be controlled is ruled by the dynamics given by Equation (6.19), where f and g are two unknown functions:

$$y^{(\nu)} = f(y) + g(y)u \tag{6.19}$$

In that case:

$$\mathrm{F} = f(y) + (g(y) - \alpha)u \tag{6.20}$$

$$u = -\left(\frac{\widehat{\mathrm{F}} - y_r^{(\nu)}}{\alpha}\right) - \mathcal{K}(e) \tag{6.21}$$

where $\mathcal{K}(e)$ is the correction on the tracking error e.

Combining Equation (6.19) with (6.21) and choosing $\mathcal{K}(e) = 0$, *i.e.* considering only the feedforward and ultralocal term of the control law, we obtain Equation (6.22):

$$y^{(\nu)} = \frac{g(y)}{\alpha} y_r^{(\nu)} + \left(f(y) - \frac{g(y)}{\alpha}\widehat{\mathrm{F}}\right) \tag{6.22}$$

Assuming a perfect estimation of F, i.e. $\widehat{\mathrm{F}} = \mathrm{F}$ given by Equation (6.20), Equation (6.22) becomes Equation (6.23). Therefore, in theory, any value of α should work. Nevertheless, if $\widehat{\mathrm{F}} = \mathrm{F}$, the solution of Equations (6.1) and (6.5) is independent of the control: the control u cannot be computed anymore.

$$y^{(\nu)} = y_r^{(\nu)} \tag{6.23}$$

However, in practice, the estimation $\widehat{\mathrm{F}}$ is subject to some dynamics, and the closed-loop cannot be reduced to Equation (6.23). In subsection 6.2.3, we will show that the quality of the controller depends mostly on the estimation error $\mathrm{F} - \widehat{\mathrm{F}}$. For this reason, the value of α needs to be chosen carefully, as close to the real gain of the system, to avoid big variations of F. This will be illustrated in section 6.3 with an example on an actual vehicle.

If $\alpha \to +\infty$, the control u given by Equation (6.21) is reduced to $\mathcal{K}(e)$. In other words, the controller is reduced to a conventional PID.

If $\alpha \to 0^+$, the control u depends mostly on the estimation $\widehat{\mathrm{F}}$. As F depends on u, the behavior of the closed-loop mostly depends both on the unknown system dynamics and the ALIEN filter dynamics. The resulting dynamics is quite unpredictable and this situation should be avoided.

It should be noticed that when α is close to the input gain $g(y)$, F is almost independent of u and its estimation becomes easier in practice, leading to a smaller estimation error.

Therefore, tuning α becomes quite straightforward for a stable open-loop system: the practitioner should set $\mathcal{K}(e) = 0$ and observe, for different values of α, the reponse of the system to the control law:

$$u = -\left(\frac{\widehat{\mathrm{F}} - \dot{y}_r}{\alpha}\right) \tag{6.24}$$

Starting with an α large enough, the control signal will be almost zero. Then, decreasing the value of α will increase the control value until the desired response given by Equation (6.23) is achieved. If α is decreased too much, the control amplitude will become too high and the system may oscillate. The tuning guidelines of α are summarized in Table 6.6.

Once α has been chosen correctly, the gains of $\mathcal{K}(e)$ have to be tuned. Let us notice that the feedforward control given by Equation (6.24) only ensures a good tracking of the reference derivative $y_r^{(\nu)}$. $\mathcal{K}(e)$ is used to track the reference y_r and its tuning is similar to the one of conventional P, PI, PID or PI^{2}D controllers.

If a perfect estimation of $\widehat{\mathrm{F}}$ is assumed, the gains $\mathrm{K_P}$ and $\mathrm{K_I}$ for a first-order system can be obtained by pole placement on the linear system given by Equation (6.25). In practice, an i-P controller is usually enough. Choosing $\mathrm{K_P}$ is a trade-off between accuracy and stability: while bigger $\mathrm{K_P}$ values lead to a faster response of the system (which may start to oscillate at some point), smaller $\mathrm{K_P}$ values lead to a more stable response.

$$\dot{e}(t) = -\alpha \mathrm{K_P} e(t) - \alpha \mathrm{K_I} \int_0^t e(\tau) d\tau \tag{6.25}$$

Initialization:
set $\mathcal{K}(e) = 0$;
choose α large enough;

Feedforward Tuning
Rule n°1: run the experiment; while the average value of $|\dot{y} - \dot{y}_r|$ decreases, decrease α and repeat rule n°1.
Rule n°2: if the system output oscillates around the setpoint, increase α;

Feedback Tuning
Rule n°1: increase K_p progressively until (i) tracking error $|y - y_r|$ is low and (ii) disturbances are rejected

Table 6.6 – Model-free controller tuning guidelines.

Similarly, for a second-order system, an i-PD controller is usually sufficient and can be tuned by pole placement on the linear system given by Equation (6.26). More precisely, choosing K_P and K_D such that $\alpha K_P = \omega_n^2$ and $\alpha K_D = 2\xi\omega_n$ where ω_n is the undamped natural frequency and ξ the damping ratio, enables to easily design the response of the system.

$$\ddot{e}(t) = -\alpha K_D \dot{e}(t) - \alpha K_P e(t) \tag{6.26}$$

6.2.3 Stability with a non-zero estimation error

In practice, during transient phases, the estimation error $F - \widehat{F}$ is non zero, even if the order of the numerical quadrature of the ALIEN filter is chosen properly. This is mainly due to the dynamics of the estimation $\widehat{F}$. Therefore, for a first-order (resp. second-order) system, Equation (6.8) (resp. Equation (6.10)) does not hold anymore and the asymptotic stability of the system is no more guaranteed. Thus, further stability studies are required.

First-order system

Consider a discrete version of Equation (6.7) with $\mathcal{K}(e) = K_P e$ (i-P controller) for a first-order system. Set $x_k = x(t_k) = x(kT_s)$ where x is the variable considered. The derivative of the error at time k, denoted $\dot{e}_k$, is approximated by a forward Euler scheme: $\dot{e}_k = (e_{k+1} - e_k)/T_s$. The following equation on the error is thus obtained:

$$e_{k+1} = (1 - T_s\alpha K_P)e_k + T_s(F_k - \widehat{F}_k) \tag{6.27}$$

Set $A = (1 - T_s\alpha K_P)$, $B = T_s$ and $f_k = (F_k - \widehat{F}_k)$. Iterating from $t = 0$, the error e_k at time t_k is given by the following relation [Kermani et al., 2012]:

$$e_k = A^k e_0 + \sum_{i=0}^{k-1} A^i B f_{k-1-i} \tag{6.28}$$

Assuming that the ∞-norm of the estimation error is bounded ($\forall k \in \mathbb{N}, |f_k| \leq f_{max} = |F - \widehat{F}|_{max}$), then Equation (6.28) becomes:

$$\begin{aligned} |e_k| &\leq |A^k e_0| + \sum_{i=0}^{k-1} |A|^i |B| f_{max} \\ &= |A^k e_0| + (1 - |A|)^{-1}(1 - |A|^k)|B| f_{max} \end{aligned} \tag{6.29}$$

If K_P and α are well chosen, *i.e.* $\frac{2}{T_s} > \alpha K_P > 0$, then for sufficiently large k, $|A^k| \approx 0$. Thus for any $\delta > 0$ sufficiently small (see Appendix C.5 for further details):

$$|e_k| \leq (1 + \delta)\left|\frac{1}{\alpha K_P}\right| |F - \widehat{F}|_{max} \tag{6.30}$$

Therefore, the system is stable as the tracking error e_k remains in a ball $\mathscr{B}\left(0, \left|\frac{f_{max}}{\alpha K_P}\right|\right)$ of radius $\frac{f_{max}}{\alpha K_P}$ centered at 0. Limiting the estimation error on F both at steady-state and in transition phases is therefore crucial. The first case is ensured by using an appropriate numerical quadrature for Equation (6.3) as mentioned in subsection 6.2.1 while the second one depends mostly on the tuning of α, where the procedure described in subsection 6.2.2 should be used.

Second-order system

For a second-order system, combining Equation (6.9) with $\mathscr{K}(e) = K_P e + K_D \dot{e}$ (i-PD controller), one obtains:

$$\dot{E} = A_c E + B_c (F - \widehat{F}) \tag{6.31}$$

with $E = \begin{pmatrix} e & \dot{e} \end{pmatrix}^T$, $A_c = \begin{pmatrix} 0 & 1 \\ -\alpha K_P & -\alpha K_D \end{pmatrix}$ and $B_c = \begin{pmatrix} 0 & 1 \end{pmatrix}^T$.

Discretizing Equation (6.31), Equation (6.32) is obtained with $A = I_2 + T_s A_c$ and $B = T_s B_c$.

$$E_{k+1} = AE_k + B(F_k - \widehat{F}_k) \tag{6.32}$$

As in the case of a first-order system, denoting $f_k = (F_k - \widehat{F}_k)$, we obtain:

$$E_k = A^k E_0 + \sum_{i=0}^{k-1} A^i B f_{k-1-i} \tag{6.33}$$

Set K_P and K_D such that $\Delta = (\alpha K_D)^2 - 4\alpha K_P \geq 0$. The two eigen values of A are then:

$$\lambda_1 = 1 - \frac{T_s}{2}\left(\sqrt{\Delta} + \alpha K_D\right) \tag{6.34}$$

$$\lambda_2 = 1 - \frac{T_s}{2}\left(-\sqrt{\Delta} + \alpha K_D\right) \tag{6.35}$$

K_P, K_D and α are all positives. As $\alpha K_D > \sqrt{\Delta}$, we get $\lambda_1 \leq 1$ and $\lambda_2 \leq 1$. Choosing $\sqrt{\Delta} + \alpha K_D \leq 2$, *i.e.* $2 \geq \alpha(2K_P - K_D)$ we get $|\lambda_1| \leq 1$ and $|\lambda_2| \leq 1$. Thus, for k sufficiently large, $||A||^k \approx 0$, which implies that for any δ sufficiently small:

$$|E_k| \leq (1+\delta)(1 - ||A||)^{-1} ||B|| f_{max} \tag{6.36}$$

Taking $||.|| = ||.||_\infty$, *i.e.* the maximum of the singular-values, we get $||A|| = \left|1 - \frac{T_s}{2}\left(-\sqrt{\Delta} + \alpha K_D\right)\right|$ and $||B|| = T_s$.

Therefore, the system is stable as the tracking error remains for the ∞-norm in a ball centered in 0 with a radius proportional to f_{max} and depending also on αK_P and $\sqrt{\Delta}$. Again, the importance of limiting the estimation error on F both in transition phases and at steady-state is highlighted.

6.3 Application to vehicle control

After having presented how to implement model-free controllers in practice, applications to coordinated brake and engine longitudinal speed control on an actual vehicle are proposed respectively in sections 6.3.1 and 6.3.2. Section 6.3.3 shows some preliminary results obtained for the lateral control of a vehicle. All the experiments were run on the vehicle presented in Figure 6.1 which belongs to ENSIAME engineering school in Valenciennes.

The structure of the vehicle for longitudinal control is depicted in Figure 6.2. All the sensors and actuators are connected to the can bus and work with a fixed sampling period of 10ms. The vehicle velocity is provided by the original sensor of the vehicle located on the gearbox output shaft. The propelling is ensured by the torque produced by the engine which is controlled by the

Figure 6.1 – The experimental vehicle belonging to ENSIAME engineering school.

Engine Control Unit (ECU). The Internal Combustion Engine (ICE) torque, whose dynamics is unknown, is transmitted to the wheels through the clutch and the gearbox which are controlled by the Gearbox Control Unit (GCU). This unit is independent and operates using its own unknown rules. The braking circuit comprises the brake pedal, the master cylinder, the Electronic Stability Program (ESP) block and the mechanical brakes.

MFC has been applied to the longitudinal dynamics of a vehicle in several works. They can be divided as following:

(i) simulation results based on actual driver data [Menhour et al., 2013, Menhour et al., 2015, d'Andréa-Novel et al., 2016, d'Andréa-Novel, 2018];

(ii) a combinaison with event-triggered control [Wang et al., 2011];

(iii) a combination with high-level Stop-and-Go strategies [Choi et al., 2009, Villagrá et al., 2010, Milanés et al., 2012].

Let us mention also [Join et al., 2008] which applies MFC for controlling the throttle of an engine.

However, only [Milanés et al., 2012] tested on an actual vehicle the coordinated control of both the brake and engine within a MFC framework. In this work, the authors compared MFC, PID and fuzzy control for controlling the longitudinal low-level of a Stop-and-Go controller. More precisely, they account for the use of two different MFC laws working alternatively: one for the throttle and one for the brake.

In this thesis, we suggest to use a cascading control structure as displayed in Figure 6.3. It comprises two parts. First, a high-level velocity controller computes the total force u_v to be applied on the wheels according to the actual velocity y^v of the vehicle, and the reference velocity y_r^v and acceleration $\dot{y}_r^v$. This high-level controller will be developed in subsection 6.3.2. The total force required u_v will then be generated by a low-level controller acting on both the brakes and the engine. For that purpose, u_v is split into two signals, the ICE force setpoint u_{ice} and the missing braking force u_{bmiss}. On the one hand, u_{ice} is sent as a setpoint of the original ECU of the vehicle which regulates the rotational speed of the engine; on the other hand, if the force generated by the engine is not sufficient during decelerations, the brakes are applied using a model-free i-P controller which will be described in section 6.3.1. For the latter case, a brake pedal position reference y_r^b is defined by a mapping from u_{bmiss} and the actual position of the brake pedal, y^b, is used as a feedback signal. The control input u_b of the brake system is the hydraulic pressures on the brakes.

The MFC framework is especially relevant in that case since the dynamics of the low-level actuators are complex and unknown. On the one hand, the vehicle propelling is ensured by an internal

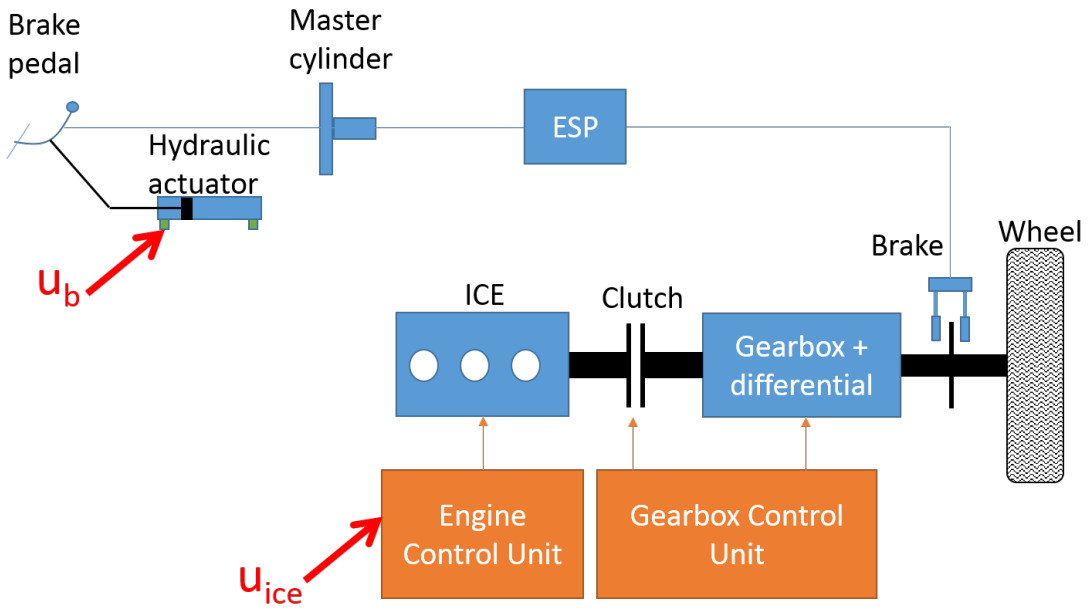


Figure 6.2 – Structure of the vehicle powertrain to be controlled.

combustion engine (ICE) controlled by the original engine control unit (ECU) of the vehicle. On the other hand, the mechanical brakes are actuated using a hydraulic actuator connected to the brake pedal. Therefore, the low-level dynamics is highly complex.

Compared to [Milanés et al., 2012], our work relied on the ALIEN filters to estimate the vehicle dynamics in order to be robust to noise measurements. Moreover, only the intelligent-Proportional (i-P) controller was investigated.

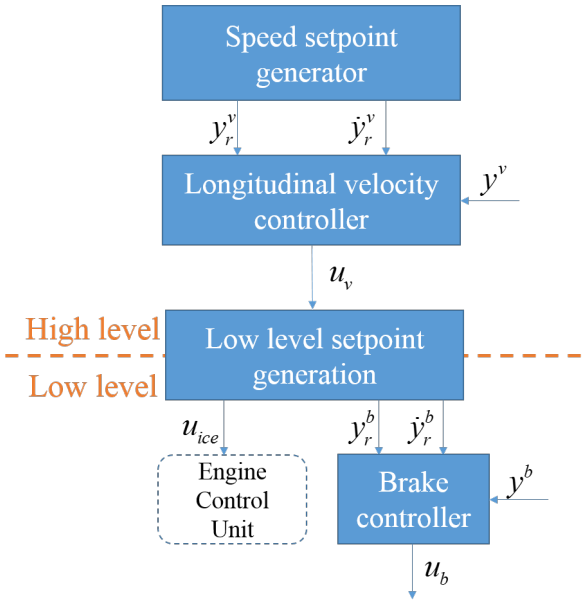


Figure 6.3 – Considered cascading control structure.

6.3.1 Brake control

The brake system consists of an hydraulic actuator that pushes or pulls the brake pedal. The position of the latter determines the pressure generated by the master cylinder on the brake disk.

The master cylinder and the brake pedal are connected together through a pushrod subject to

some stiction which makes the system very difficult to model. Also, the brake pedal dynamics is non-linear due to the master cylinder's resistive force. The objective is to control the brake pedal position y^b by opening the valves ($u_b \in [-80;80]\%$) located respectively at the top and at the bottom of the hydraulic actuator. The system is described in Figure 6.4. The main difficulties to control such a system are: (i) the possible occurence of limit cycles due to pushrod stiction; (ii) reduced stability margin due to communication delays; (iii) variations of the system gain due to the return spring and the internal pressure of the master cylinder; (iv) unpredictable variations of the system gain over time due to the pressure of the tank and the pump. Note that the system is subject to a known 20ms communication delay. The so-called "position" signal is generated by a control unit on the CAN bus. When the brake pedal is fully released, the "position" signal is 700; when it is fully pressed, the signal is 1600.

The open-loop response of the system to different input steps is depicted in Figure 6.5 and illustrates the non-linear behavior of the system. The cumulated effect of stiction and communication delays induces a response "delay" between 60ms and 180ms.

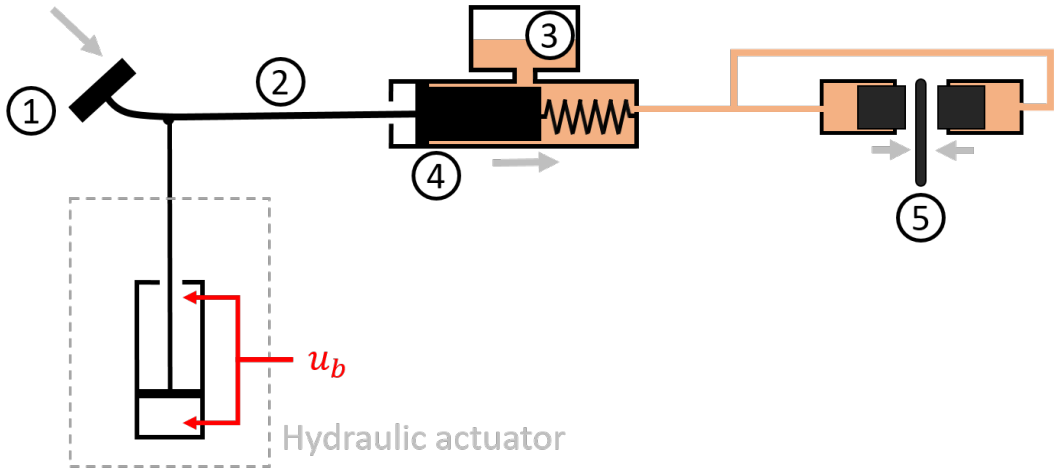


Figure 6.4 – Description of the brake system: the brake pedal (1) pushes the master cylinder (4) through the pushrod (2); the pressure of the fluid increases and activates the brakes on the brake disk (5). (3) is the fluid reservoir. The hydraulic actuator is used to control the brake pedal position.

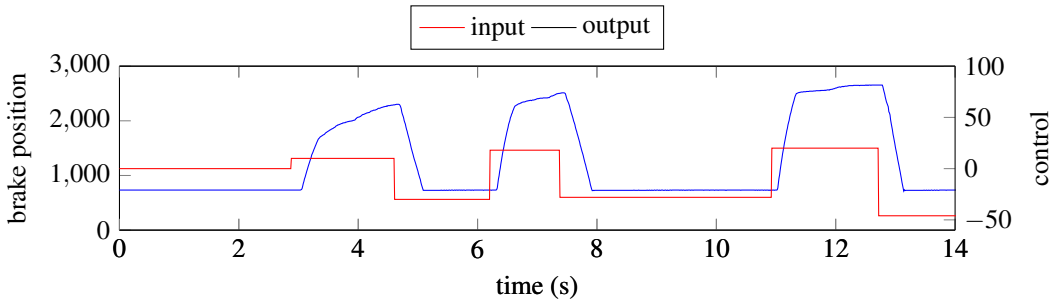


Figure 6.5 – Open-loop response of the brakes.

The MFC law was applied using the tuning method suggested in section 6.2.2 while the vehicle is at standstill with the engine turned on. The overall tuning takes only a few minutes. First, the proportional gain K_P is set to 0. The system being at the rest position, a reference signal is generated and the open-loop responses are recorded for different values of α (see Figure 6.6), starting from $\alpha = 10000$ and decreasing its value until the output and the control signal starts to oscillate (at $\alpha = 200$, Figure 6.7). Finally, the value of α is obtained : $\alpha = 800$. Then, the proportional gain K_P is increased such that the dynamics of the output error is acceptable, see Figure 6.8. The final tuning of the i-P controller is then $\alpha = 800$, $K_P = 2 \times 10^{-3}$.

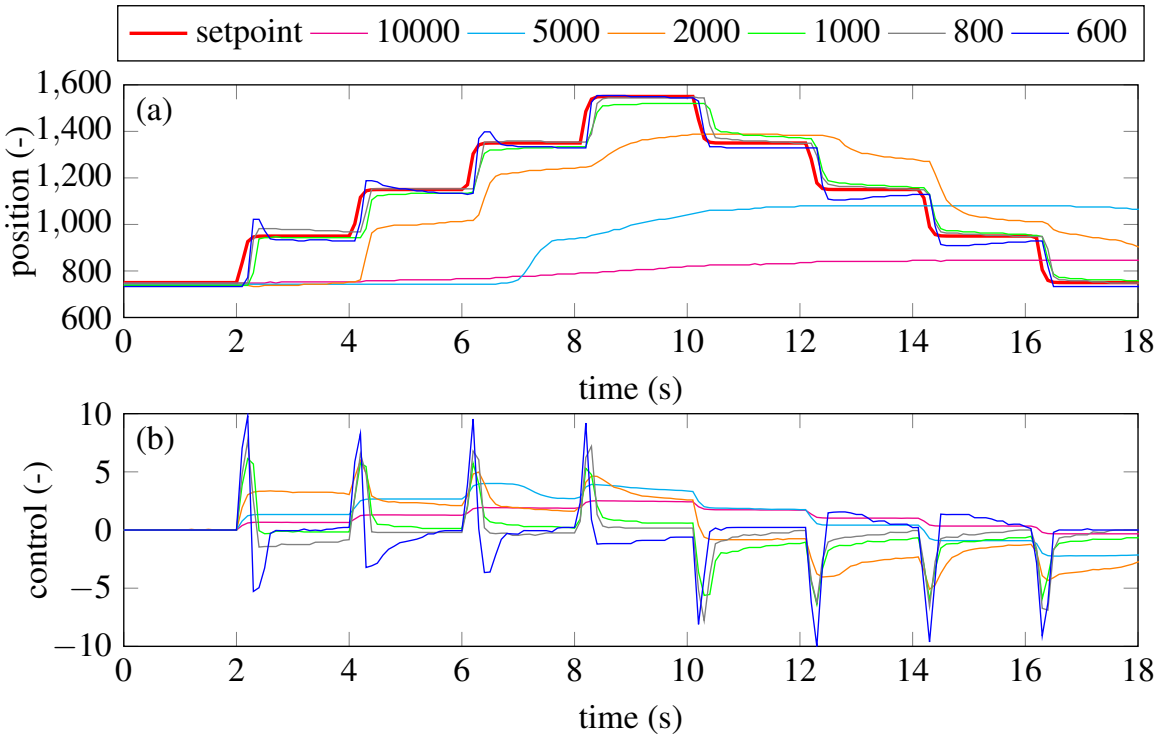


Figure 6.6 – Tuning of parameter α for the brake system in open-loop.

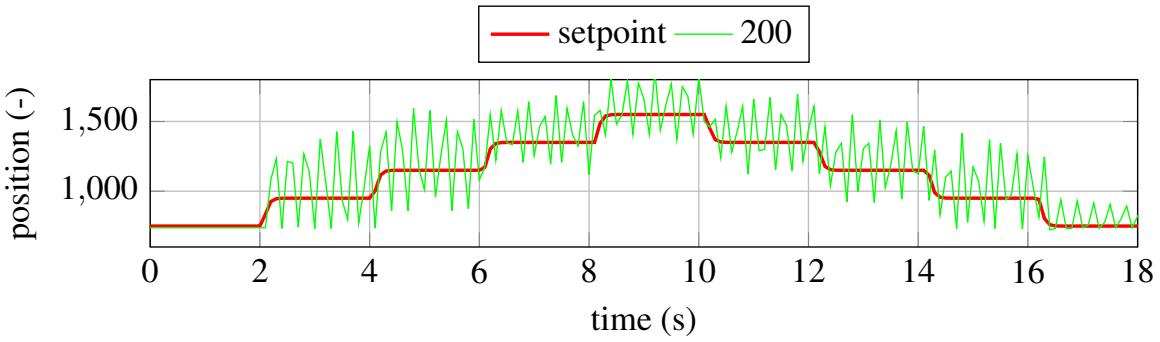


Figure 6.7 – Feedforward response for α = 200 of the brake system.

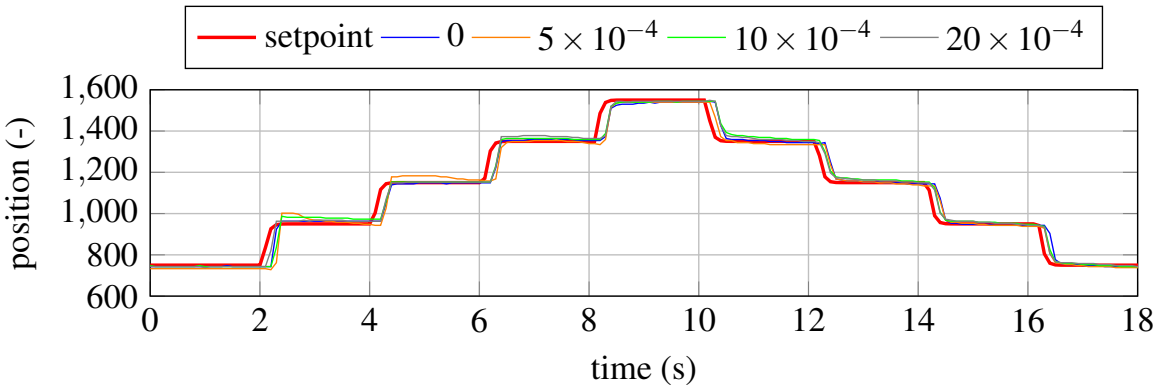


Figure 6.8 – Tuning of parameter K_P for the brake system (α = 800).

The obtained i-P controller is then compared to a PI controller using the same reference signal. The PI controller tuning is $K_P = 5 \times 10^{-2}$ and $K_I = 5 \times 10^{-3}$. The reference position of the brake pedal is shown in red in Figure 6.9. The i-P controller performance is slightly better than the PI one as it provides a lower steady-state error (but not much). Moreover, on a practical point of view,

the i-P controller has a simpler tuning procedure than the PI one: the parameter α is obtained in a systematic way, then the K_P parameter is tuned empirically whereas for the PI, both parameters need to be tuned simultaneously.

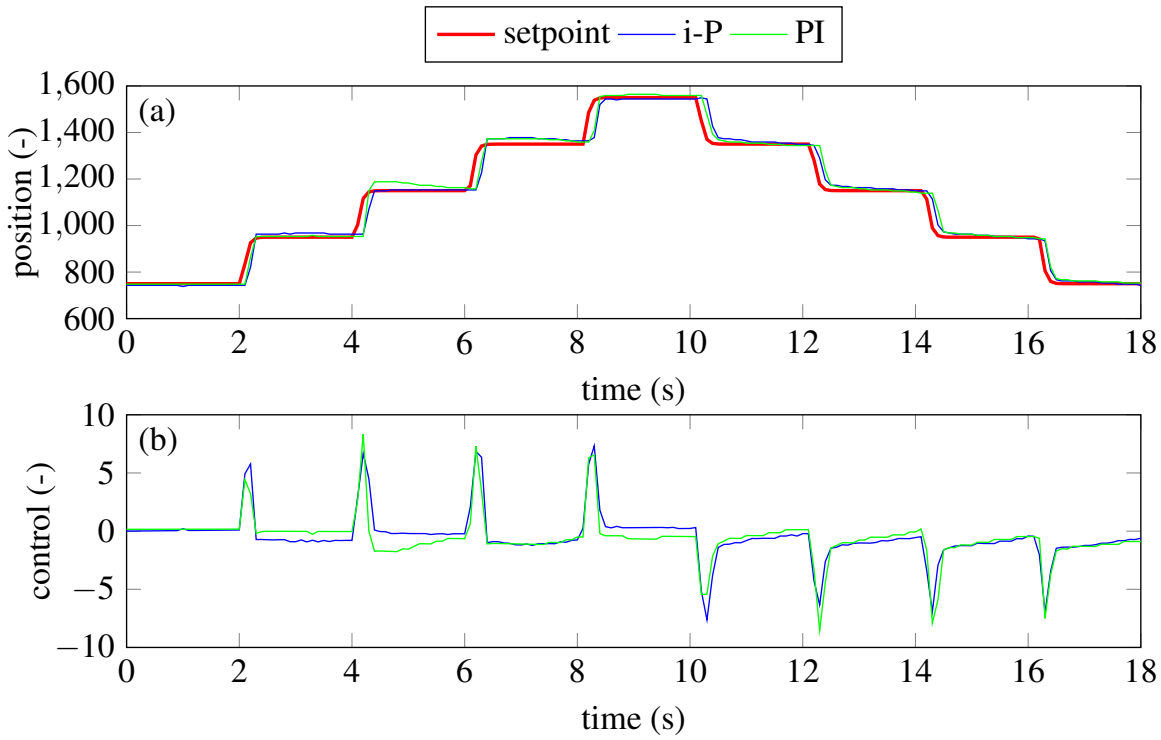


Figure 6.9 – Comparison between PI and i-P controllers for the brake system

6.3.2 Longitudinal speed control of the vehicle

The objective of this part is to control the longitudinal speed y^v of the vehicle presented on Figure 6.1 by computing the control u_v, which corresponds to the total wheel force requested on the wheels in order to track the reference speed y_r^v. As depicted in Figure 6.3, the actual total wheel force applied to the vehicle is subject to the dynamics of the low-level controllers. Although the velocity control is not highly challenging, it is still subject to the following difficulties: (i) at low speed, for small u_{ice}, as the GCU controls the vehicle velocity at 10km/h using the clutch (idle speed), the system is not controllable; (ii) during propelling, the system becomes non controllable when the clutch is open during a gear shift sequence; (iii) the low-level ICE torque is poorly controlled by the ECU and causes thus disturbances on the control signal.

The control signal is send to the ECU which regulates the rotational speed of the engine. If the force generated by the engine is not sufficient during decelerations, the brakes are applied, using the MFC i-P controller with the parameter values obtained in subsection 6.3.1.

Tests were performed on the track presented in Figure 6.10. The track altitude profile is depicted in Figure 6.11. The disturbance caused by the slope at the beginning of the track forces to use the brakes in order to maintain the vehicle at standstill.

In order to test the control law, the reference speed is composed of successive steps of ±10m/s triggered every 10s. This raw reference signal is then smoothen using a second-order filter with time constant 0.4s to obtain a C^1 speed reference y_r, shown in red in Figures 6.12 to 6.14.

Tuning of the control parameters

The procedure suggested in section 6.2.2 has been applied for the tuning of α . Throughout this subsection, the time window of the ALIEN filter T is chosen equal to 0.1s which is a good trade-

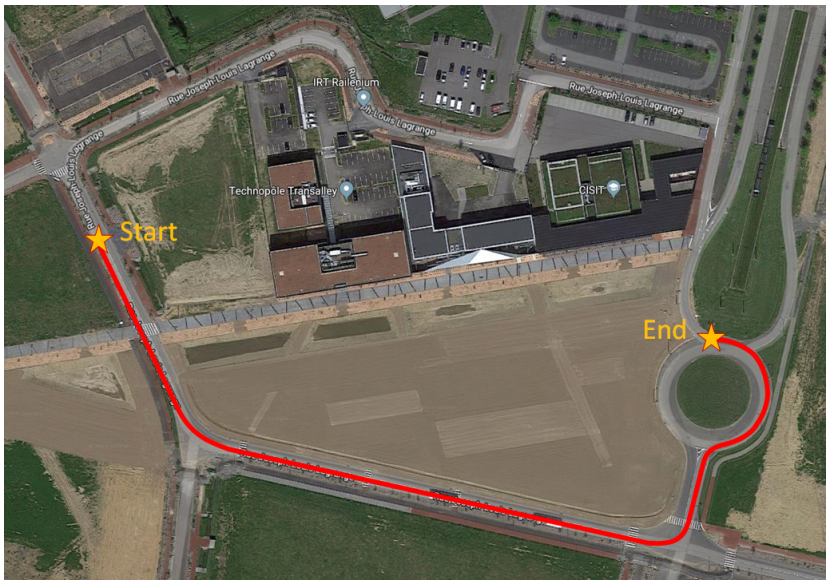


Figure 6.10 – Top view of the open road test field track.

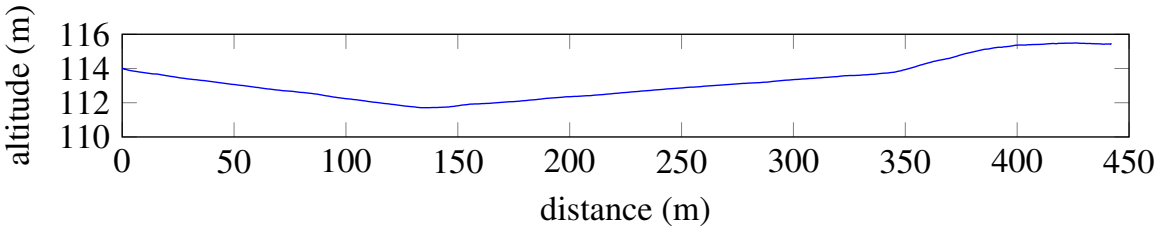


Figure 6.11 – Altitude of the track.

off between noise filtering and the filter bandwidth. The results obtained for different values of α using only the feedforward control law ($K_P = 0$) are shown in Figure 6.12.

For large values such as $\alpha \in [1, 1000]$, the control amplitude is too low. In such situations, the gearbox control unit hands over the clutch and the engine to maintain the vehicle speed at approximatively 12km/h (which corresponds to the idle speed). This feature is standard for vehicle equipped with an automatic transmission. The reference tracking is improved by decreasing α. Acceptable values are between 0.005 and 0.01. As expected, when α is too low ($\alpha = 1 \times 10^{-3}$), the closed loop becomes unstable and the experiment is stopped earlier for safety reasons. At this stage of the tuning procedure, $\alpha = 0.008$ is chosen.

The second part of the tuning procedure consists in choosing K_P such that the system output smoothly tracks the reference signal. If a perfect estimation for $\widehat{F}$ is assumed, then the closed-loop pole lay at $-\alpha K_P$ (see Equation (6.8)). The closed-loop response should be perfectly damped. In practice, as shown in Figure 6.13, it is not the case as some overshoots for large K_P can be observed. $K_P = 100$ is chosen as a trade-off between the closed-loop response time and overshoots.

After a fine tuning of the parameters, the following final values are obtained for the i-P controller: $\alpha = 0.006$ and $K_P = 100$. The results are shown in blue in Figure 6.14.

Comparison with a PI controller

Finally, the i-P controller is compared with a PI controller. The results are shown in Figure 6.14. First, between $t \in [0, 10]$s, both the i-P and the PI controllers are able to reject the disturbance due to the initial slope. Then, the i-P controller provides a faster tracking of the reference than the PI. This is due to the feedforward term $\dot{y}_r$ included in the control law, see Equation (6.5). The PI controllers exhibits a 26% overshoots on the first step at $t = 15.2$s. The i-P controller overshoot is only 7.8% at $t = 14.0$s. Due to the system non-linearities, the other overshoots are smaller for both

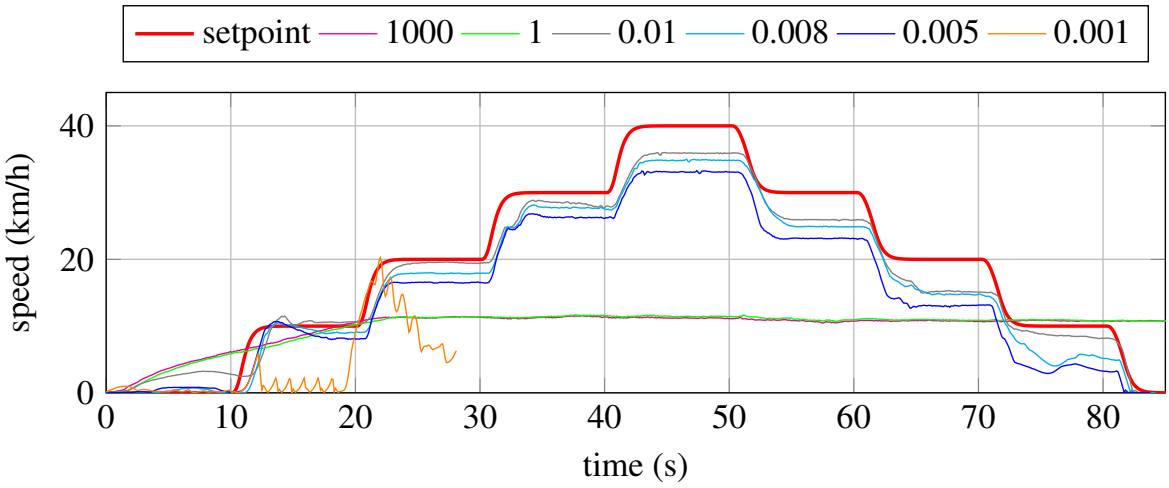


Figure 6.12 – Tuning of the α parameter for the high-level controller.

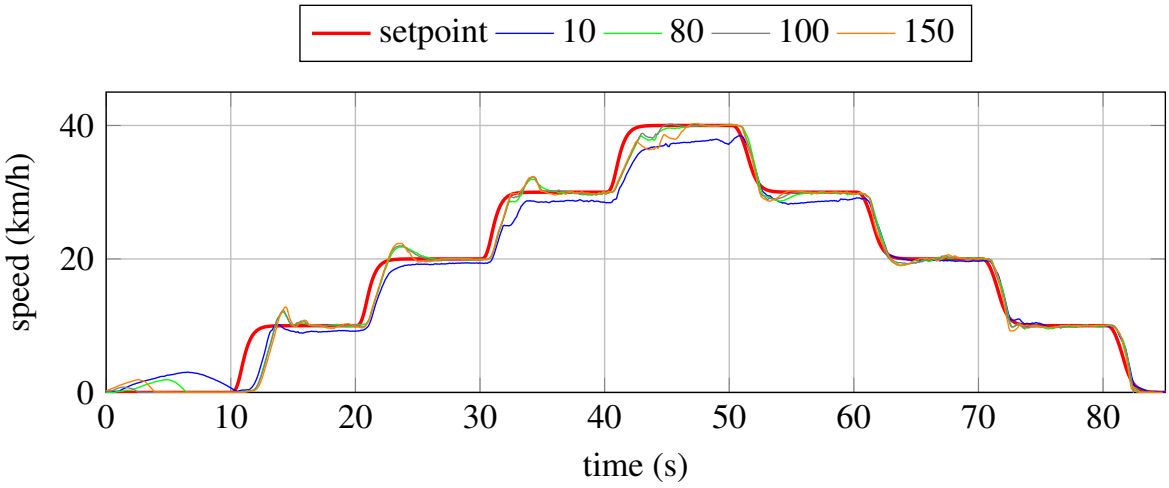


Figure 6.13 – Response of the model-free controller for different values of K_P and $\alpha = 0.008$.

controllers. The overshoots of the PI controller can be reduced at the price of a slower tracking. Finally, the overall controller performances can be measured using the root mean square (RMS) of the tracking error. Data are given in Table 6.7. The i-P performs 36% better than the PI.

According to the authors' experience, the tuning of the PI controller was less intuitive than the i-P and did require more time and experimental trials. The system dynamics is strongly affected by the engaged gear. As the chosen PI controller has fixed gain, the final tuning has been chosen to average the performances over the gears 1 to 3 used in this experiment. Better tuning can be obtained on individual gears at the price of worse performance on the other gears. On the contrary, the tuning of the i-P controller was relatively straightforward with the procedure proposed in section 6.2.2. Moreover, it is less sensible to gear changes since the system dynamics is implicitly captured through the estimation of the unknown F.

During acceleration phases, gear shifts occur as shown in Figure 6.14. They are engaged by the GCU whose control law is unknown. During this period, the clutch is opened. Therefore, the system is not controllable as the engine torque generated is not transmitted to the wheels. During deceleration phases, gear shifts happen too but do not impact the controllability of the system as the brakes can still be applied.

Controller	RMS of the tracking error (km/h)
PI	2.32
i-P	1.48

Table 6.7 – Comparison of the root mean square of the tracking error between PI and i-P controllers.

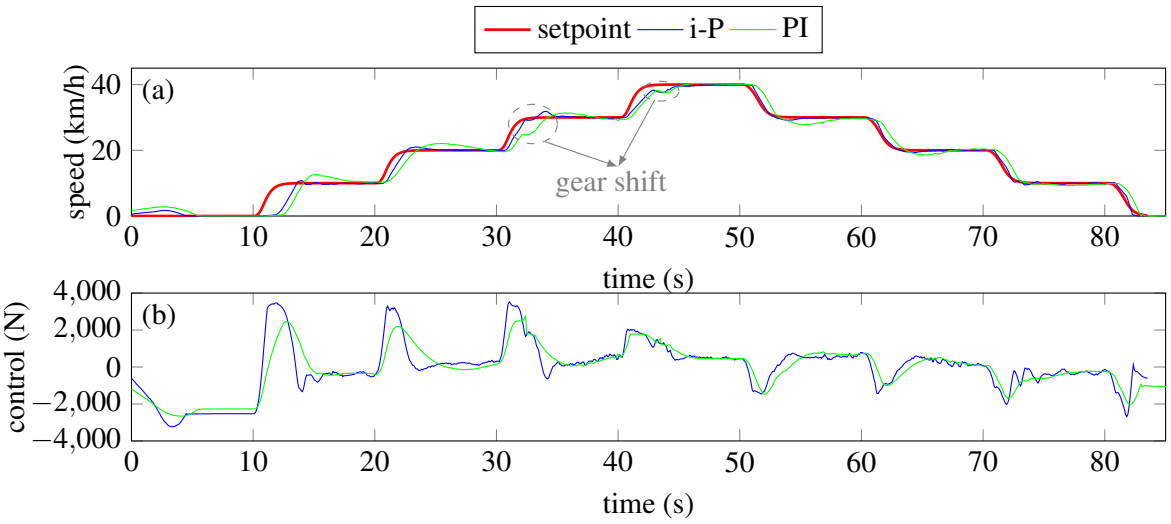


Figure 6.14 – Comparison of the speed profile between the i-P and PI controllers.

6.3.3 Lateral control

The purpose of the lateral controller is to guide the vehicle along a given reference path while the longitudinal speed of the vehicle is being controlled by another independent system.

Existing works on lateral model-free controllers comprise tests in perfect simulation environment, where neither noise nor delays were added. The output considered is either the lateral deviation [Menhour et al., 2015, d'Andréa-Novel, 2018] or the flat lateral output of the dynamic bicycle model $l_f M_T V_y - I_z \dot{\psi}$ [Menhour et al., 2013]. In both cases, a second-order ultralocal model is considered. However, neither the ALIEN filters were used due the huge numerical errors caused by a trapezoidal rule for computing the time integral, nor noise and delays were added to the measurements.

After some tests on a simple 3 DoF dynamic bicycle vehicle model with linear tire model, it turns out that lateral MFC is not very robust to actuation delays or dynamics on the actuator. This can be observed in Figure 6.15, 6.16 and 6.17 where strong oscillations occurs and the system never becomes stable again for a 50ms delay. In particular, the tuning of parameter K_P and K_D becomes almost impossible.

Moreover, when considering the lateral error as the observation output, the reference output y_r is always zero as well as its derivative: in particular $\ddot{y}_r = 0$. Therefore, unlike the longitudinal model-free controller, the feedforward term of the lateral controller is zero. Thus, the structure of the model-free controller present little interest in this case compared to a conventional PID controller.

Therefore, we proposed in [Delprat et al., 2019] to control the output $y = \gamma_0 \tilde{y} + \tilde{\psi}$, which is a ponderation between the lateral error $\tilde{y}$ and the heading angle error $\tilde{\psi}$. The ponderation factor γ_0 was set equal to 10. Due to the absence of a feedforward term, the heading error is defined as the angle between the direction of the vehicle and the projection of the look-ahead point.

Remark 17. *In [Delprat et al., 2019], we proposed a slight modification of the ultralocal equation model which will be mentioned in conclusion of this chapter, in order to improve the stability of the system. This also easen the tuning of the feedback gains.*

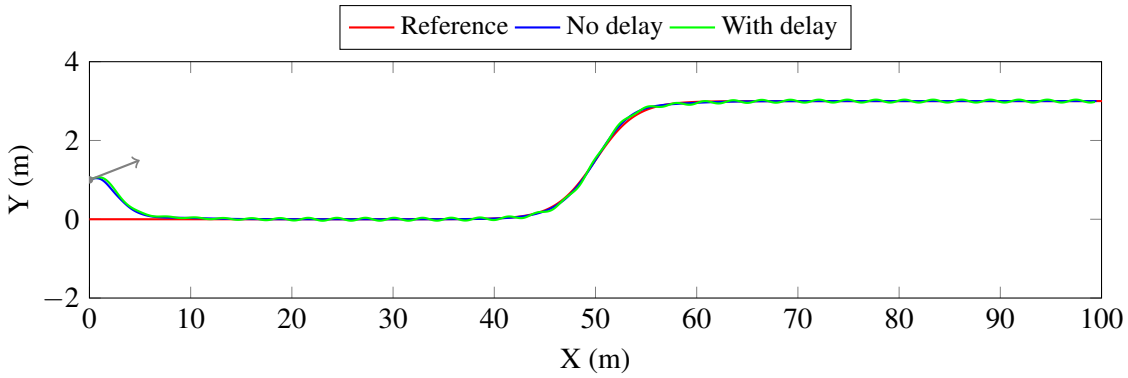


Figure 6.15 – Lane change maneuver with a lateral MFC at V = 5m/s without (blue) and with 50ms actuation delays (green). The initial configuration of the vehicle is given in grey and $K_P = 5$, $K_D = 6.47$.

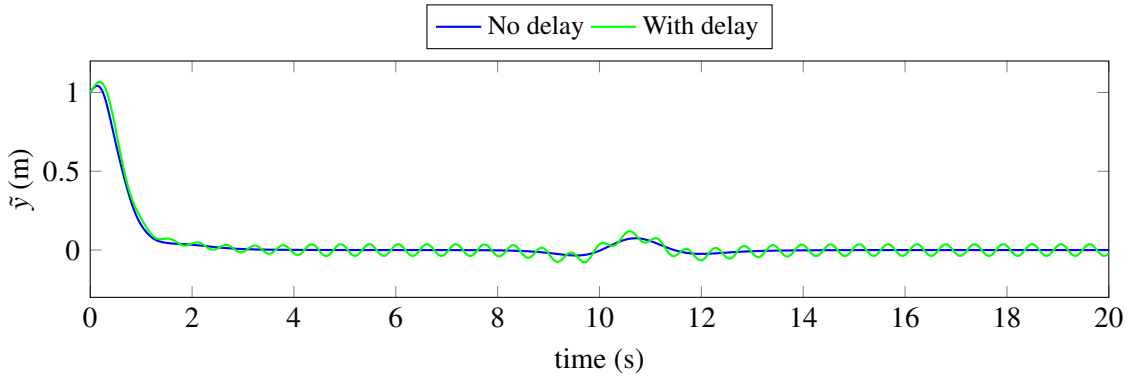


Figure 6.16 – Lateral error for a lane change maneuver with a lateral MFC at V = 5m/s without (blue) and with 50ms actuation delays (green) for $K_P = 5$, $K_D = 6.47$.

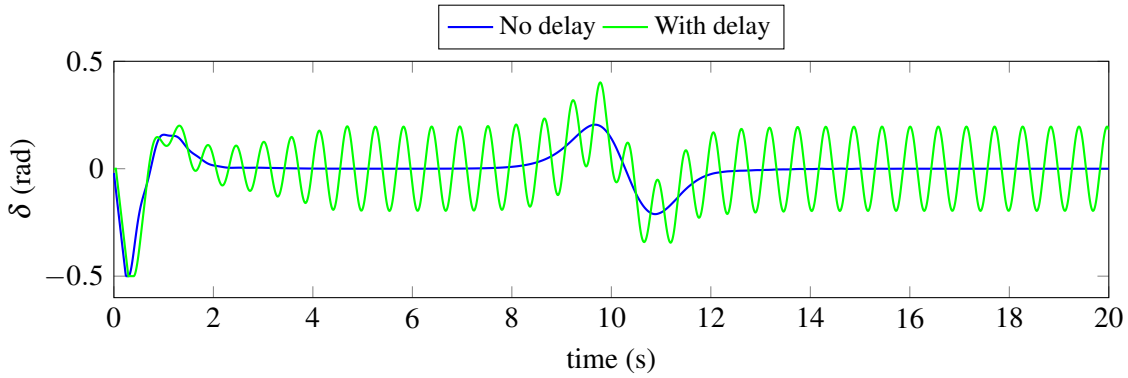


Figure 6.17 – Steering angle for a lane change maneuver with a lateral MFC at V = 5m/s without (blue) and with 50ms actuation delays (green) for $K_P = 5$, $K_D = 6.47$.

6.4 Conclusion

The model-free control paradigm introduced in [Fliess and Join, 2013] is still at its early stages but could represent an interesting alternative to PID controllers in the automotive industry. While PID are pure regulators, model-free controllers can also track a varying reference output. Moreover, they are able to capture variation of parameters through the estimation of the dynamics F.

However, more research still needs to be conducted. In particular, a better understanding of the impact of the filter dynamics on the overall performance of the controller should be studied. Also, the impact of the tuning of α on the stability of the system could be further analyzed. Finally, one interesting improvement would be to enable time-varying α gains, by adapting its value on-

line. Some attempts in that direction have been made in [Doublet et al., 2016, Polack et al., 2017].

At last, in [Delprat et al., 2019], we proposed a new ultralocal model given by Equation (6.37) for systems of order $\nu \geq 2$. For $\beta_i > 0$, $i = 1..\nu-1$, the system is stable if satisfying the Routh stability criterion and the tuning of the feedback term of the control law is greatly simplified. This improvement has enabled to easen the implementation of lateral controllers based on MFC for autonomous vehicles as shown in section 6.3.3 but is not developed in this manuscript.

$$y^{(\nu)} + \sum_{i=1}^{\nu-1} \beta_i y^{(i)} = F + \alpha u \tag{6.37}$$

Chapter 7

Conclusion

" Eternity is a very long time, especially towards the end "

Woody Allen

As we move towards mass deployment of autonomous vehicles, it is important to be able to guarantee their safety at all time. However, many techniques still used nowadays for motion planning and control of autonomous vehicles are derived from earlier work on wheeled robots. The latters are usually lighter and operate at lower speed compared to actual vehicles, which make their safe operation easier due to less inertia. Moreover, they are often evolving in a controlled environment with few dynamic obstacles, where the longitudinal and lateral motions of the robots can be considered separately. Even when applying more recent techniques such as Model Predictive Control, the level of modeling chosen is often inconsistent with the reality, in particular at the motion planning level, which could bring the whole system in jeopardy.

7.1 The “intention = action" equation

In order to ensure the safety at anytime of the autonomous vehicles of tomorrow, it is crucial to ensure that their intention, expressed as a planned reference trajectory computed by the motion planner, matches with their actual actions, *i.e.* the trajectory actually followed by the low-level controllers of the vehicle. For that purpose, two keypoints must be ensured in the motion planning and control architecture of an autonomous vehicle. First, the intention should be dynamically feasible, *i.e.* the reference trajectory respects the laws of motion of the vehicle; this implies to use appropriate models at the motion planning layer. Secondly, the low-level controllers, responsible for the actions of the vehicle, should be designed such that they can track the given reference trajectory.

7.1.1 The feasibility of the intention

Although the kinematic constraints also referred to as the non-holonomic constraints of the vehicle have been well considered in the motion planning literature, using for example Dubins'path or a kinematic bicycle model, the impact of wheel dynamics on the generation of safe trajectories has been clearly understudied. Many works neglect these effects at the motion planning level which might lead to an unfeasible planned trajectory whatever the chosen low-level controller is. Such problems are more likely to occur in the future as the speed of autonomous vehicles will increase, the road friction coefficients decrease and overtaking maneuvers be more common.

Therefore, this thesis proposed to investigate the limits of the kinematic bicycle model used for motion planning. Some macroscopic safety criteria, in particular a dynamic constraint on the steering angle depending on the speed, was obtained in order to ensure that the wheel dynamics does not interfere with the feasibility of the planned trajectory. Although this solution might seem

a bit restrictive in terms of efficiency, it corresponds to how a human drives. Moreover, a heuristic speed has been defined and added to the motion planning and control architecture to improve the driving efficiency. The obtained trajectory is ensured to be not only safe but also dynamically feasible. Several convincing simulations were performed for different road conditions (dry, wet and icy roads). Moreover, this safety criterion also ensures that any conventional low-level controllers, in particular those based on a kinematic bicycle model such as the pure-pursuit or the kinematic chained form controller, will be able to track the reference trajectory.

Hence, this work claims to use more computational resources for the motion planning layer in order to obtain better reference trajectories. Instead, most of the existing work in the literature focuses on increasing the computational power of the low-level controllers in order to achieve the handling limits of the vehicle. This can be seen as a paradigm shift: not only is the obtained driving behavior closer to the one of an actual human driver, but also safety guarantees can be obtained. However, to limit the increase of computational burden at the motion planning level, model reduction is necessary which comes at a cost of less accuracy.

7.1.2 The suitability of the low-level controller for performing the action

In order to ensure the consistency of the planning and control architecture, low-level controllers, in particular lateral ones, have been compared too. The conclusion is that the “perfect controller", so dear to the motion planning community, does not exist. Instead, there is a dilemma between accuracy and stability[1] which needs to be considered. Therefore, we believe that the “best and universal" controller does not exist. Instead, different controllers should be used in different driving situations depending on the smoothness and the accuracy desired.

The controller should be chosen carefully in order to ensure it is able to perform the intention of the vehicle. In particular, if a highly dynamic emergency maneuver is intended, the controller should be able to reach the handling limits of the vehicle while ensuring its stability.

7.1.3 A cornerstone for cooperative driving

The adequation between the intention of the autonomous vehicle and its actual action is not only important for the vehicle's own safety. It is actually a cornerstone for safe cooperation between vehicles through V2V or V2I communications. In particular, for intelligent intersection management, cooperative merging maneuvers or overtaking, a vehicle that does not respect its own intention that has been broadcasted to the other vehicles or infrastructure can put all the other participants in jeopardy.

7.2 Perspectives

7.2.1 A unified framework for motion planning from decision-making to dynamic feasibility

At the decision-making level, simple models such as a point-mass or a unicycle model are used for computational efficiency. However, noticing that a unicycle model is a kinematic bicycle model expressed at the rear wheels, some further simplifications could be conducted in order to obtain an integrated motion planner, able to deal simultaneously with decision-making and dynamic feasibility of the reference trajectory.

7.2.2 Machine learning for initial guess and emergency scenarios

Although we believe that the application of machine learning algorithms in autonomous vehicles should be limited for explanability reasons, they can be used at several stages of the motion plan-

[1] Or more evoking to the passenger: the comfort.

ning and control architecture of an autonomous vehicle to enhance the overall performances. In particular, two interesting applications of machine learning algorithms are:

- Choosing the almost-optimal class of homotopy - Everytime a vehicle faces a static or dynamic obstacle, it has to make a binary choice: for a vehicle moving in the same direction, it can either overtake it on the left or on the right; for a vehicle crossing its lane at some point, the ego-vehicle can either pass before or after the other one. Every choice corresponds to a class of homotopy [Bender et al., 2015, Gregoire et al., 2014]. Hence, if n_{obs} is the number of obstacles, there exist $2^{n_{obs}}$ classes of homotopy. Making a combinatorial search to know in which homotopy class is the optimal trajectory would require a lot of computational resources. Instead, reinforcement learning algorithms or deep-learning algorithms could learn the most suited homotopy class from a driving situations to give an initial guess to the optimal control problem and avoid local minima. The motion planning MPC layer presented in this thesis could then be applied afterwards for finding the optimal trajectory within this homotopy class.
- Learning the vehicle dynamics - The equations of the dynamics, in particular at the wheel levels, are complex. Therefore, supervised learning algorithms for time-series could be applied to learn the vehicle dynamics from a dataset of recorded trajectories and control input sequences. In [Devineau et al., 2018], we have conducted some preliminary research on this topic and derived a coupled longitudinal and lateral controller which will be tested on an actual vehicle in the future. Learning the vehicle dynamics can also be used at the motion planning level, in particular for generating safe control sequences in emergency situations.

7.2.3 Addind uncertainties about the environment

One limitation of this work is that we consider that the environment is known and fully observable: this is not the case in reality as the intention of the other drivers cannot be known if there is no communication between vehicles. Therefore, the results obtained should be integrated in a probabilistic framework at the planning level to take into account uncertainties. Such situations could be tested in a ADAS prototyping simulation platform such as PreScan[TASS International,].

7.2.4 Taking into account the dynamics of the actuators at the motion planning level

Another consistency problem between the motion planning and control levels that has not been tackled in this thesis is the dynamics of the actuators. In particular, in chapter 6, a one-second response delay to a longitudinal acceleration demand was observed due to the dynamic of the Engine Control Unit and communication delays. However, these effects have not been considered at the motion planning level where the desired acceleration profile is assumed to be performed immediately. These actuator dynamics can have a strong impact on the safety of the reference trajectory in the case of emergency maneuvers. Tube-based MPC taking into account the actuators imperfections (delay, noise,...) could be used for example [Gao et al., 2014].

Bibliography

[Abbas et al., 2014] Abbas, M. A., Milman, R., and Eklund, J. M. (2014). Obstacle avoidance in real time with Nonlinear Model Predictive Control of autonomous vehicles. In *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering*, volume 40, pages 1–6. IEEE. 65

[Altché et al., 2017] Altché, F., Polack, P., and de La Fortelle, A. (2017). High-speed trajectory planning for autonomous vehicles using a simple dynamic model. In *Proc. of the Intelligent Transportation Systems Conference*. 60, 66

[Astrom and Hagglund, 1995] Astrom, K. J. and Hagglund, T. (1995). *PID Controllers: Theory, Design, and Tuning*. Instrument Society of America: Research Triangle Park. 77, 96, 118

[Attia et al., 2012a] Attia, R., Orjuela, R., and Basset, M. (2012a). Coupled longitudinal and lateral control strategy improving lateral stability for autonomous vehicle. In *American Control Conference (ACC), 2012*, pages 6509–6514. IEEE. 116

[Attia et al., 2012b] Attia, R., Orjuela, R., and Basset, M. (2012b). *Longitudinal control for automated vehicle guidance*, volume 45. IFAC. 93

[Baffet et al., 2009] Baffet, G., Charara, A., and Lechner, D. (2009). Estimation of vehicle sideslip, tire force and wheel cornering stiffness. *Control Engineering Practice*, 17(11):1255 – 1264. 100

[Bara et al., 2017] Bara, O., Olama, M., Djouadi, S., Kuruganti, T., Fliess, M., and Join, C. (2017). Model-free load control for high penetration of solar photovoltaic generation. In *2017 North American Power Symposium (NAPS)*, Morgantown, United States. 118

[Barraquand et al., 1996] Barraquand, J., Kavraki, L., Latombe, J.-C., Li, T.-Y., Motwani, R., and Raghavan, P. (1996). A random sampling scheme for path planning. In Giralt, G. and Hirzinger, G., editors, *Robotics Research*, pages 249–264, London. Springer London. 50

[Bender et al., 2015] Bender, P., Tas, O. S., Ziegler, J., and Stiller, C. (2015). The combinatorial aspect of motion planning: Maneuver variants in structured environments. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, number IV, pages 1386–1392. IEEE. 87, 139

[Berntorp, 2017] Berntorp, K. (2017). Path planning and integrated collision avoidance for autonomous vehicles. In *2017 American Control Conference (ACC)*, pages 4023–4028, Seattle, WA, USA. IEEE. 65

[Besselmann and Morari, 2009] Besselmann, T. and Morari, M. (2009). Autonomous vehicle steering using explicit LPV-MPC. In *2009 European Control Conference (ECC)*, number 1, pages 2628–2633. IEEE. 60

[Branicky, 1998] Branicky, M. (1998). Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):475–482. 115

[Brynjolfsson and McAfee, 2014] Brynjolfsson, E. and McAfee, A. (2014). *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*. W. W. Norton & Company, 1st edition. 6, 54

[Burckhardt, 1993] Burckhardt, M. (1993). Fahrwerktechnik: Radschlupf-regelsysteme. 1993. *Vogel-Verlag, Wurtzburg*. 36

[Cadena et al., 2016] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6):1309–1332. 12

[Camacho and Bordons, 1999] Camacho, E. F. and Bordons, C. (1999). *Model Predictive Control*. 55

[Canny, 1988] Canny, J. (1988). *The complexity of robot motion planning*. 49

[Canudas-de Wit and Horowitz, 1999] Canudas-de Wit, C. and Horowitz, R. (1999). Observers for tire/road contact friction using only wheel angular velocity information. In *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*, volume 4, pages 3932–3937. IEEE. 35

[Canudas de Wit et al., 1995] Canudas de Wit, C., Olsson, H., Astrom, K., and Lischinsky, P. (1995). A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425. 35

[Canudas de Wit and Tsiotras, 1999] Canudas de Wit, C. and Tsiotras, P. (1999). Dynamic tire friction models for vehicle traction control. In *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*, volume 4, pages 3746–3751. IEEE. 35

[Canudas-de Wit et al., 2003] Canudas-de Wit, C., Tsiotras, P., Velenis, E., Basset, M., and Gissinger, G. (2003). Dynamic Friction Models for Road/Tire Longitudinal Interaction. *Vehicle System Dynamics*, 39(3):189–226. 35

[Chaib et al., 2004] Chaib, S., Netto, M. S., and Mammar, S. (2004). H/sub/spl infin//, adaptive, pid and fuzzy control: a comparison of controllers for vehicle lane keeping. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 139–144. IEEE. 95

[Chand et al., 2016] Chand, A. N., Kawanishi, M., and Narikiyo, T. (2016). Non-linear model-free control of flapping wing flying robot using iPID. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2016-June, pages 2930–2937. IEEE. 118

[Choi et al., 2009] Choi, S., d'Andréa-Novel, B., Fliess, M., Mounier, H., and Villagra, J. (2009). Model-free control of automotive engine and brake for stop-and-go scenarios. In *Control Conference (ECC), 2009 European*, pages 3622–3627. IEEE. 127

[Coulter, 1992] Coulter, R. C. (1992). *Implementation of the Pure Pursuit Path Tracking Algorithm*. Carnegie Mellon University. 64, 97

[Dahlquist and Björck, 2008] Dahlquist, G. and Björck, Å. (2008). *Numerical Methods in Scientific Computing, Volume I*, volume 1. Society for Industrial and Applied Mathematics. 121

[d'Andréa et al., 2010] d'Andréa, B., Fliess, M., Join, C., Mounier, H., and Steux, B. (2010). A mathematical explanation via "intelligent" PID controllers of the strange ubiquity of PIDs. In *18th Mediterranean Conference on Control and Automation, MED'10*, pages 395–400. IEEE. 120

[d'Andréa Novel, 1995] d'Andréa Novel, B. (1995). Systèmes mécaniques non holonomes. In *Introduction à la Commande Non Linéaire des Sysèmes Dynamiques. Applications à la Robotique*. XI

[d'Andréa-Novel, 2018] d'Andréa-Novel, B. (2018). Model-Free Control of Longitudinal and Lateral Dynamics for Automated Vehicles. *JTEKT ENGINEERING JOURNAL English Edition*, (1015):1–7. 127, 134

[d'Andréa-Novel et al., 2016] d'Andréa-Novel, B., Menhour, L., Fliess, M., and Mounier, H. (2016). Some remarks on wheeled autonomous vehicles and the evolution of their control design. *IFAC-PapersOnLine*, 49(15):199–204. 127

[Delprat et al., 2019] Delprat, S., Polack, P., and d'Andréa-Novel, B. (2019). A revised look at intelligent-pid model free control (in submission). 134, 136

[Devineau et al., 2018] Devineau, G., Polack, P., Altché, F., and Moutarde, F. (2018). Coupled Longitudinal and Lateral Control of a Vehicle using Deep Learning . *2018 21st International IEEE Conference on Intelligent Transportation Systems (ITSC) (accepted)*. 87, 116, 139

[Dijkstra, 1959] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271. 50

[Donges, 1978] Donges, E. (1978). A Two-Level Model of Driver Steering Behavior. *Human Factors*, 20(6):691–707. 115

[Dormand and Prince, 1980] Dormand, J. and Prince, P. (1980). A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26. 38

[Doublet et al., 2016] Doublet, M., Join, C., and Hamelin, F. (2016). Model-free control for unknown delayed systems. In *3rd Conf. on Control and Fault-Tolerant Systems*, Barcelona, Spain. 136

[Dubins, 1957] Dubins, L. E. (1957). On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents. *American Journal of Mathematics*, 79(3):497. 58

[Dugoff et al., 1969] Dugoff, H., Fancher, P. S., and Segel, L. (1969). Tire perfomance characteristics affecting vehicle response to steering and braking control inputs. *Highway Safety Research Institute,University of Michigan*, 460. 34

[Englund et al., 2016] Englund, C., Chen, L., Ploeg, J., Semsar-Kazerooni, E., Voronov, A., Bengtsson, H. H., and Didoff, J. (2016). The Grand Cooperative Driving Challenge 2016: boosting the introduction of cooperative automated vehicles. *IEEE Wireless Communications*, 23(4):146–152. 7

[Ersal et al., 2008] Ersal, T., Fathy, H. K., Rideout, D. G., Louca, L. S., and Stein, J. L. (2008). A Review of Proper Modeling Techniques. *Journal of Dynamic Systems, Measurement, and Control*, 130(6):061008. 15, 66

[Falcone et al., 2007a] Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., and Hrovat, D. (2007a). Predictive Active Steering Control for Autonomous Vehicle Systems. *IEEE Transactions on Control Systems Technology*, 15(3):566–580. 59, 66

[Falcone et al., 2007b] Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., and Hrovat, D. (2007b). Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on control systems technology*, 15(3):566–580. 65, 94, 116

[Fiala, 1954] Fiala, E. (1954). Seitenkrafte am rollenden luftreifen'z. vdi bd. 96, no. 29. 37

[Fliess and Join, 2008] Fliess, M. and Join, C. (2008). Intelligent PID controllers. In *2008 16th Mediterranean Conference on Control and Automation*, volume 44, pages 326–331. IEEE. 119

[Fliess and Join, 2009] Fliess, M. and Join, C. (2009). Model-free Control and Intelligent PID Controllers: Towards a Possible Trivialization of Nonlinear Control? *IFAC Proceedings Volumes*, 42(10):1531–1550. 118

[Fliess and Join, 2013] Fliess, M. and Join, C. (2013). Model-free control. *International Journal of Control*, 86(12):2228–2252. 16, 118, 135

[Fliess et al., 1995] Fliess, M., Lévine, J., Martin, P., and Rouchon, P. (1995). Flatness and defect of non-linear systems: introductory theory and examples. *International Journal of Control*, 61(6):1327–1361. 118

[Fliess and Sira-Ramírez, 2003] Fliess, M. and Sira-Ramírez, H. (2003). An algebraic framework for linear identification. *ESAIM: Control, Optimisation and Calculus of Variations*, 9:151–168. 118, 119, XIII

[Funke et al., 2017] Funke, J., Brown, M., Erlien, S. M., and Gerdes, J. C. (2017). Collision Avoidance and Stabilization for Autonomous Vehicles in Emergency Scenarios. *IEEE Transactions on Control Systems Technology*, 25(4):1204–1216. 60

[Funke et al., 2012] Funke, J., Theodosis, P., Hindiyeh, R., Stanek, G., Kritatakirana, K., Gerdes, C., Langer, D., Hernandez, M., Muller-Bessler, B., and Huhnke, B. (2012). Up to the limits: Autonomous Audi TTS. In *2012 IEEE Intelligent Vehicles Symposium*, pages 541–547. IEEE. 59

[Gao et al., 2014] Gao, Y., Gray, A., Tseng, H. E., and Borrelli, F. (2014). A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Vehicle System Dynamics*, 52(6):802–823. 139

[Gao et al., 2010] Gao, Y., Lin, T., Borrelli, F., Tseng, E., and Hrovat, D. (2010). Predictive Control of Autonomous Ground Vehicles With Obstacle Avoidance on Slippery Roads. In *ASME Dynamic Systems and Control Conference*, volume 1, pages 265–272. ASME. 65, 66

[Gillespie, 1997] Gillespie, T. D. (1997). *Fundamentals of Vehicle Dynamics*. Society of Automotive Engineers. 20

[Goh and Gerdes, 2016] Goh, J. Y. and Gerdes, J. C. (2016). Simultaneous stabilization and tracking of basic automobile drifting trajectories. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, number Iv, pages 597–602. IEEE. 37, 116

[Gonzalez et al., 2016] Gonzalez, D., Perez, J., Milanes, V., and Nashashibi, F. (2016). A Review of Motion Planning Techniques for Automated Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1135–1145. 48

[Gray et al., 2012] Gray, A., Gao, Y., Lin, T., Hedrick, J. K., Tseng, H. E., and Borrelli, F. (2012). Predictive control for agile semi-autonomous ground vehicles using motion primitives. In *American Control Conference (ACC)*, pages 4239–4244. 60, 66

[Gregoire et al., 2014] Gregoire, J., Bonnabel, S., and De La Fortelle, A. (2014). Priority-based coordination of robots. 139

[Guntur and Sankar, 1980] Guntur, R. and Sankar, S. (1980). A friction circle concept for dugoff's tyre friction model. *International Journal of Vehicle Design*, 1(4):373–377. 34

[Hessburg et al., 1991] Hessburg, T., Peng, H., Tomizuka, M., Zhang, W.-B., and Kamei, E. (1991). An experimental study on lateral control of a vehicle. In *American Control Conference, 1991*, pages 3084–3089. IEEE. 94

[Hessburg and Tomizuka, 1994] Hessburg, T. and Tomizuka, M. (1994). Fuzzy logic control for lateral vehicle guidance. *IEEE control systems*, 14(4):55–63. 94

[Hindiyeh and Gerdes, 2014] Hindiyeh, R. Y. and Gerdes, J. C. (2014). A Controller Framework for Autonomous Drifting: Design, Stability, and Experimental Validation. *Journal of Dynamic Systems, Measurement, and Control*, 136(5):051015. 37

[Hingwe et al., 2002] Hingwe, P., Tan, H.-S., Packard, A. K., and Tomizuka, M. (2002). Linear parameter varying controller for automated lane guidance: experimental study on tractor-trailers. *IEEE Transactions on control systems technology*, 10(6):793–806. 94

[Hingwe and Tomizuka, 1997] Hingwe, P. and Tomizuka, M. (1997). Experimental evaluation of a chatter free sliding mode control for lateral control in ahs. In *American Control Conference, 1997. Proceedings of the 1997*, volume 5, pages 3365–3369. IEEE. 94

[Houska et al., 2011a] Houska, B., Ferreau, H., and Diehl, M. (2011a). ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3):298–312. 55, 64, 76

[Houska et al., 2011b] Houska, B., Ferreau, H. J., and Diehl, M. (2011b). An auto-generated real-time iteration algorithm for nonlinear mpc in the microsecond range. *Automatica*, 47(10):2279–2285. 54

[Jiang et al., 2016] Jiang, H., Wang, Z., Chen, Q., and Zhu, J. (2016). Obstacle avoidance of autonomous vehicles with CQP-based model predictive control. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, volume 223, pages 001668–001673. IEEE. 65

[Jiménez et al., 1998] Jiménez, P., Thomas, F., and Torras, C. (1998). Collision detection algorithms for motion planning. In *Robot Motion Planning and Control*, pages 305–343. Springer. 53

[Join et al., 2013] Join, C., Chaxel, F., and Fliess, M. (2013). "Intelligent" controllers on cheap and small programmable devices. In *2013 Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 554–559. IEEE. 118

[Join et al., 2008] Join, C., Masse, J., Fliess, M., Join, C., Masse, J., and Fliess, M. (2008). Etude préliminaire d ' une commande sans modèle pour papillon de moteur. 127

[Kalman and Bucy, 1961] Kalman, R. E. and Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Journal of basic engineering*, 83(1):95–108. 13

[Katrakazas et al., 2015] Katrakazas, C., Quddus, M., Chen, W.-H., and Deka, L. (2015). Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60:416–442. 48

[Kavraki et al., 1996] Kavraki, L., Svestka, P., Latombe, J.-C., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580. 53

[Kermani et al., 2012] Kermani, S., Delprat, S., Guerra, T., Trigui, R., and Jeanneret, B. (2012). Predictive energy management for hybrid vehicle. *Control Engineering Practice*, 20(4):408–420. 125

[Khalil, 1996] Khalil, H. K. (1996). Noninear systems. *Prentice-Hall, New Jersey*, 2(5):5–1. 29

[Khatib, 1986] Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, 2:500–505. 49

[Khodayari et al., 2010] Khodayari, A., Ghaffari, A., Ameli, S., and Flahatgar, J. (2010). A historical review on lateral and longitudinal control of autonomous vehicle motions. In *Mechanical and Electrical Technology (ICMET), 2010 2nd International Conference on*, pages 421–429. IEEE. 92

[Kiencke and Nielsen, 2005] Kiencke, U. and Nielsen, L. (2005). *Automotive Control Systems*, volume 38. Springer Berlin Heidelberg, Berlin, Heidelberg. 20, 36

[Kokotović et al., 1999] Kokotović, P., Khalil, H. K., and O'reilly, J. (1999). *Singular perturbation methods in control: analysis and design*. SIAM. 29

[Kong et al., 2015] Kong, J., Pfeiffer, M., Schildbach, G., and Borrelli, F. (2015). Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design. In *IEEE Intelligent Vehicles Symposium*, pages 1094–1099, Seoul, Korea. 67

[Kozlov et al., 1979] Kozlov, M. K., Tarasov, S. P., and KHACHIIAN, L. (1979). Polynomial solvability of the convex quadratic programming. *Doklady Akademii Nauk SSSR*, 248(5):1049–1051. 65

[Kritayakirana and Gerdes, 2012] Kritayakirana, K. and Gerdes, J. C. (2012). Autonomous vehicle control at the limits of handling. *International Journal of Vehicle Autonomous Systems*, 10(4):271. 59, 116

[Kuwata et al., 2008] Kuwata, Y., Fiore, G., Teo, J., Frazzoli, E., and How, J. (2008). Motion planning for urban driving using RRT. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1681–1686. IEEE. 58

[Latombe, 2013] Latombe, J.-c. (2013). *Robot motion plannig*, volume 53. 48

[Laumond et al., 1994] Laumond, J.-P., Jacobs, P., Taix, M., and Murray, R. (1994). A motion planner for nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 10(5):577–593. 57

[Laumond et al., 1998] Laumond, J. P., Sekhavat, S., and Lamiraux, F. (1998). Guidelines in nonholonomic motion planning for mobile robots. In *Robot Motion Planning and Control*, volume 229, pages 1–53. Springer-Verlag, London. XI

[Laurgeau, 2009] Laurgeau, C. (2009). *Le siècle de la voiture intelligente*. Collection mathématiques & informatique. Ecole nationale supérieure des mines. 6

[Lavalle, 2006] Lavalle, S. M. (2006). Planning Algorithms. *Cambridge*, page 842. 48

[LaValle and Kuffner, 2001] LaValle, S. M. and Kuffner, J. J. (2001). Randomized Kinodynamic Planning. *The International Journal of Robotics Research*, 20(5):378–400. 53

[Lenain, 2005] Lenain, R. (2005). *Thèse Contribution à la modélisation et à la commande de robots mobiles en présence de glissement*. PhD thesis, Université Blaise Pascal Clermont II. 100

[Lenain et al., 2010] Lenain, R., Thuilot, B., Cariou, C., and Martinet, P. (2010). Mixed kinematic and dynamic sideslip angle observer for accurate control of fast off-road mobile robots. *Journal of Field Robotics*, 27(2):181–196. 100

[Li et al., 2017] Li, X., Sun, Z., Cao, D., Liu, D., and He, H. (2017). Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles. *Mechanical Systems and Signal Processing*, 87:118–137. 65

[Liang and Peng, 1999] Liang, C.-Y. and Peng, H. (1999). Optimal adaptive cruise control with guaranteed string stability. *Vehicle System Dynamics*, 32(4-5):313–330. 92

[Liniger et al., 2015] Liniger, A., Domahidi, A., and Morari, M. (2015). Optimization-based autonomous racing of 1:43 scale RC cars. *Optimal Control Applications and Methods*, 36(5):628–647. 65

[Liu et al., 2013] Liu, J., Jayakumar, P., Overholt, J. L., Stein, J. L., and Ersal, T. (2013). The Role of Model Fidelity in Model Predictive Control Based Hazard Avoidance in Unmanned Ground Vehicles Using LIDAR Sensors. In *ASME 2013 Dynamic Systems and Control Conference*. 67

[Lozano-Perez, 1983] Lozano-Perez (1983). Spatial Planning: A Configuration Space Approach. *IEEE Transactions on Computers*, C-32(2):108–120. 48

[Luenberger, 1971] Luenberger, D. (1971). An introduction to observers. *IEEE Transactions on automatic control*, 16(6):596–602. 13

[Martinez and Canudas-de Wit, 2007] Martinez, J.-J. and Canudas-de Wit, C. (2007). A Safe Longitudinal Control for Adaptive Cruise Control and Stop-and-Go Scenarios. *IEEE Transactions on Control Systems Technology*, 15(2):246–258. 92

[Menhour et al., 2011a] Menhour, L., D'Andréa-Novel, B., Boussard, C., Fliess, M., and Mounier, H. (2011a). Algebraic nonlinear estimation and flatness-based lateral/longitudinal control for automotive vehicles. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, number Umr 8506, pages 463–468. IEEE. 116

[Menhour et al., 2015] Menhour, L., d'Andréa-Novel, B., Fliess, M., Gruyer, D., and Mounier, H. (2015). A new model-free design for vehicle control and its validation through an advanced simulation platform. *European Control Conference*. 127, 134

[Menhour et al., 2013] Menhour, L., d'Andréa-Novel, B., Fliess, M., and Mounier, H. (2013). Multivariable decoupled longitudinal and lateral vehicle control: A model-free design. In *52nd IEEE Conference on Decision and Control*, pages 2834–2839. IEEE. 127, 134

[Menhour et al., 2011b] Menhour, L., Lechner, D., and Charara, A. (2011b). Two degrees of freedom PID multi-controllers to design a mathematical driver model: experimental validation and robustness tests. *Vehicle System Dynamics*, 49(4):595–624. 94, 97

[Milanés et al., 2012] Milanés, V., Villagrá, J., Godoy, J., and González, C. (2012). Comparing Fuzzy and Intelligent PI Controllers in Stop-and-Go Manoeuvres. *IEEE Transactions on Control Systems Technology*, 20(3):770–778. 127, 128

[Minorsky, 1922] Minorsky, N. (1922). Directional stability of automatically steered bodies. *Journal of the American Society for Naval Engineers*, 34(2):280–309. 95

[Murgovski and Sjöberg, 2015] Murgovski, N. and Sjöberg, J. (2015). Predictive cruise control with autonomous overtaking. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 644–649. 41

[Murphy, 2000] Murphy, R. R. (2000). *Introduction to AI Robotics*. MIT Press. 11

[Netto et al., 2004] Netto, M. S., Chaib, S., and Mammar, S. (2004). Lateral adaptive control for vehicle lane keeping. In *American Control Conference, 2004. Proceedings of the 2004*, volume 3, pages 2693–2698. IEEE. 95

[Nilsson et al., 2015a] Nilsson, J., Brannstrom, M., Coelingh, E., and Fredriksson, J. (2015a). Longitudinal and lateral control for automated lane change maneuvers. In *2015 American Control Conference (ACC)*, pages 1399–1404. IEEE. 65

[Nilsson et al., 2015b] Nilsson, J., Falcone, P., Ali, M., and Sjoberg, J. (2015b). Receding horizon maneuver generation for automated highway driving. 41. 41

[Pacejka, 2002] Pacejka, H. B. (2002). *Tyre and Vehicle Dynamics*. Butterworth-Heinemann. 29, 30, 32, 38, 82

[Pacejka and Besselink, 1997] Pacejka, H. B. and Besselink, I. J. M. (1997). Magic Formula Tyre Model with Transient Properties. *Vehicle System Dynamics*, 27(sup001):234–249. 29

[Paden et al., 2016] Paden, B., Cap, M., Yong, S. Z., Yershov, D., and Frazzoli, E. (2016). A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55. 48

[Park et al., 2009] Park, J.-m., Kim, D.-W., Yoon, Y.-s., Kim, H. J., and Yi, K.-s. (2009). Obstacle avoidance of autonomous vehicles based on model predictive control. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 223(12):1499–1516. 60, 67

[Persson et al., 2005] Persson, B. N. J., Tartaglino, U., Albohr, O., and Tosatti, E. (2005). Rubber friction on wet and dry road surfaces: The sealing effect. *Phys. Rev. B*, 71:035428. 71, 84

[Peter E. Hart et al., 1968] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael (1968). Formal Basis for the Heuristic Determination of Minimum Cost Paths. *Systems Science and Cybernetics*, (2):100–107. 50

[Ploeg et al., 2012] Ploeg, J., Shladover, S., Nijmeijer, H., and van de Wouw, N. (2012). Introduction to the Special Issue on the 2011 Grand Cooperative Driving Challenge. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):989–993. 7

[Polack et al., 2017] Polack, P., d'Andréa-Novel, B., Fliess, M., de La Fortelle, A., and Menhour, L. (2017). Finite-Time Stabilization of Longitudinal Control for Autonomous Vehicles via a Model-Free Approach. *IFAC-PapersOnLine*, 50(1):12533–12538. 136

[Qian et al., 2016] Qian, X., Altche, F., Bender, P., Stiller, C., and de La Fortelle, A. (2016). Optimal trajectory planning for autonomous driving integrating logical constraints: An MIQP perspective. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 205–210, Rio de Janeiro, Brazil. IEEE. 41, 65, 66

[Qian et al., 2017] Qian, X., Altché, F., Grégoire, J., and de La Fortelle, A. (2017). Autonomous intersection management systems: criteria, implementation and evaluation. *IET Intelligent Transport Systems*, 11(3):182–189. 13

[Rajamani, 2012] Rajamani, R. (2012). *Vehicle Dynamics and Control*. Springer. 20, 43, 67, 101

[Reeds and Shepp, 1990] Reeds, J. and Shepp, L. (1990). Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393. 58

[Reggiani et al., 1990] Reggiani, M., Mazzoli, M., and Caselli, S. (1990). An experimental evaluation of collision detection packages for robot motion planning. In *IEEE/RSJ International Conference on Intelligent Robots and System*, volume 3, pages 2329–2334. IEEE. 53

[Reif, 1979] Reif, J. H. (1979). Complexity of the mover's problem and generalizations. In *Proceedings of the 20th Annual Symposium on Foundations of Computer Science*, pages 421–427, Washington, DC, USA. IEEE Computer Society. 49

[Rice, 1973] Rice, R. S. (1973). Measuring car-driver interaction with the g-g diagram. In *1973 International Automotive Engineering Congress and Exposition*. SAE International. 59

[Rosolia et al., 2017] Rosolia, U., De Bruyne, S., and Alleyne, A. G. (2017). Autonomous Vehicle Control: A Nonconvex Approach for Obstacle Avoidance. *IEEE Transactions on Control Systems Technology*, 25(2):469–484. 65

[Samson, 1993] Samson, C. (1993). Time-varying Feedback Stabilization of Car-like Wheeled Mobile Robots. *The International Journal of Robotics Research*, 12(1):55–64. 118

[Samson, 1995] Samson, C. (1995). Control of chained systems application to path following and time-varying point-stabilization of mobile robots. *IEEE Transactions on Automatic Control*, 40(1):64–77. 99

[Schlömilch et al., 1901] Schlömilch, O., Witzschel, B., Cantor, M., Kahl, E., Mehmke, R., and Runge, C. (1901). *Zeitschrift für Mathematik und Physik*. Number vol. 46. B. G. Teubner. 122

[Schürmann et al., 2017] Schürmann, B., Heß, D., Eilbrecht, J., Stursberg, O., Köster, F., and Althoff, M. (2017). Ensuring drivability of planned motions using formal methods. In *Proc. of the Intelligent Transportation Systems Conference*. 60, 66

[Schwartz and Sharir, 1981] Schwartz, J. T. and Sharir, M. (1981). On the piano movers problem i: The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers, department of computer science, courant institute of'mathematical sciences, nyu, report 39. 48

[Schwartz and Sharir, 1983] Schwartz, J. T. and Sharir, M. (1983). On the "piano movers" problem. II. General techniques for computing topological properties of real algebraic manifolds. *Advances in Applied Mathematics*, 4(3):298–351. 49

[Sharp et al., 2000] Sharp, R. S., Casanova, D., and Symonds, P. (2000). A Mathematical Model for Driver Steering Control , with Design, Tuning and Performance results. *Vehicle System Dynamics*, 33(768414408):289–326. 94, 96

[Shinohara et al., 2016] Shinohara, M., Oda, T., Nonaka, K., and Sekiguchi, K. (2016). Model predictive path following control with acceleration constraints for front steering vehicles. In *2016 IEEE 14th International Workshop on Advanced Motion Control (AMC)*, pages 71–78. IEEE. 94

[Shladover, 2006] Shladover, S. (2006). PATH at 20 – History and Major Milestones. In *2006 IEEE Intelligent Transportation Systems Conference*, number 1, pages 1_22–1_29. IEEE. 6

[Snider, 2009] Snider, J. M. (2009). Automatic steering methods for autonomous automobile path tracking. Technical Report CMU-RI-TR-09-08, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. 95, 97, 98

[Sotelo, 2003] Sotelo, M. A. (2003). Lateral control strategy for autonomous steering of ackerman-like vehicles. *Robotics and Autonomous Systems*, 45(3-4):223–233. 100

[Sussmann, 1982] Sussmann, H. (1982). Lie brackets, real analyticity and geometric control, differential geometric control theory, brockett, millman, sussmann eds. XI

[Swaroop et al., 2001] Swaroop, D., Karl Hedrick, J., and Choi, S. B. (2001). Direct adaptive longitudinal control of vehicle platoons. *IEEE Transactions on Vehicular Technology*, 50(1):150–161. 92

[Talj et al., 2013] Talj, R., Tagne, G., Charara, A., Talj, R., Tagne, G., Immersion, A. C., Control, I., Talj, R., Tagne, G., and Charara, A. (2013). Immersion and Invariance Control for Lateral Dynamics of Autonomous Vehicles , with Experimental Validation To cite this version : Immersion and Invariance Control for Lateral Dynamics of Autonomous Vehicles , with Experimental Validation. *IEEE Intelligent Vehicles Symposium*. 94

[Tarski, 1951] Tarski, A. (1951). *A Decision Method for Elementary Algebra and Geometry*. University of California Press. 49

[TASS International,] TASS International. http://www.tassinternational.com/prescan. 39, 77, 139

[Thorpe et al., 1991a] Thorpe, C., Herbert, M., Kanade, T., and Shafer, S. (1991a). Toward autonomous driving: the cmu navlab. i. perception. *IEEE expert*, 6(4):31–42. 6

[Thorpe et al., 1991b] Thorpe, C., Herbert, M., Kanade, T., and Shafter, S. (1991b). Toward autonomous driving: the cmu navlab. ii. architecture and systems. *IEEE expert*, 6(4):44–52. 6

[Thrun et al., 2006] Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk,

J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., and Mahoney, P. (2006). Stanley: The robot that won the darpa grand challenge: Research articles. *J. Robot. Syst.*, 23(9):661–692. 93, 98

[Thuilot et al., 2004] Thuilot, B., Bom, J., Marmoiton, F., and Martinet, P. (2004). Accurate automatic guidance of an urban electric vehicle relying on a kinematic gps sensor. *IFAC Proceedings Volumes*, 37(8):155–160. 100

[Villagrá et al., 2010] Villagrá, J., Milanés, V., Pérez, J., and González, C. (2010). Model-free control techniques for Stop & Go systems. *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1899–1904. 127

[Waluś and Olszewski, 2011] Waluś, K. J. and Olszewski, Z. (2011). Analysis of tire-road contact under winter conditions. In *Proceedings of the World Congress on Engineering*, volume 3, London, U.K. 84

[Wang et al., 2011] Wang, J., Mounier, H., Cela, A., and Niculescu, S. I. (2011). *Event driven intelligent PID controllers with applications to motion control*, volume 18. IFAC. 127

[Wang et al., 2015] Wang, X., Fu, M., Ma, H., and Yang, Y. (2015). Lateral control of autonomous vehicles based on fuzzy logic. *Control Engineering Practice*, 34:1 – 17. 94

[Weiskircher and Ayalew, 2015] Weiskircher, T. and Ayalew, B. (2015). Predictive ADAS: A predictive trajectory guidance scheme for advanced driver assistance in public traffic. In *2015 European Control Conference (ECC)*, pages 3402–3407. IEEE. 98

[Williams et al., 2016] Williams, G., Drews, P., Goldfain, B., Rehg, J. M., and Theodorou, E. A. (2016). Aggressive driving with model predictive path integral control. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1433–1440, Stockholm, Sweden. IEEE. 57

[Yoshida, 1984] Yoshida, K. (1984). Operational Calculus, A Theory of Hyperfunction. 1. XIII

[Zalila et al., 1998] Zalila, Z., Bonnay, F., and Coffin, F. (1998). Lateral guidance of an autonomous vehicle by a fuzzy logic controller. In *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, volume 2, pages 1996–2001. IEEE. 94

[Zhan et al., 2016] Zhan, W., Liu, C., Chan, C.-Y., and Tomizuka, M. (2016). A non-conservatively defensive strategy for urban autonomous driving. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 459–464, Rio de Janeiro, Brazil. IEEE. 59, 65, 66

[Zhao et al., 2007] Zhao, S., Li, Y., Zheng, L., and Lu, S. (2007). Vehicle Lateral Stability Control Based on Sliding Mode Control *. *Proceedings of the IEEE International Conference on Automation and Logistics*, pages 638–642. 94

[Ziegler et al., 2014a] Ziegler, J., Bender, P., Dang, T., and Stiller, C. (2014a). Trajectory planning for Bertha - a local, continuous method. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 450–457, Dearborn, Michigan, USA. IEEE. 104

[Ziegler et al., 2014b] Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Thao Dang, Franke, U., Appenrodt, N., Keller, C. G., Kaus, E., Herrtwich, R. G., Rabe, C., Pfeiffer, D., Lindner, F., Stein, F., Erbs, F., Enzweiler, M., Knoppel, C., Hipp, J., Haueis, M., Trepte, M., Brenk, C., Tamke, A., Ghanaat, M., Braun, M., Joos, A., Fritz, H., Mock, H., Hein, M., and Zeeb, E. (2014b). Making Bertha Drive - An Autonomous Journey on a Historic Route. *IEEE Intelligent Transportation Systems Magazine*, 6(2):8–20. 7, 65, 66

[Ziegler and Nichols, 1993] Ziegler, J. G. and Nichols, N. B. (1993). Optimum Settings for Automatic Controllers. *Journal of Dynamic Systems, Measurement, and Control*, 115(2B):220. 96

Part IV

Appendices

Appendix A

Vehicle dynamics

A.1 Equations for the linear velocities of a four-wheel vehicle model

For the 10 DoF vehicle model, we apply the fundamental principle of dynamics to the whole vehicle in the inertial coordinate system $(\vec{e}_X, \vec{e}_Y, \vec{e}_Z)$, projected on the axes of the vehicle coordinate system $(\vec{e}_x, \vec{e}_y, \vec{e}_z)$:

$$M_T(\dot{\vec{V}}.\vec{e}_x) = \sum_{i=1}^{4} F_{xi} + F_{aero_x} + P_x \tag{A.1a}$$

$$M_T(\dot{\vec{V}}.\vec{e}_y) = \sum_{i=1}^{4} F_{yi} + P_y \tag{A.1b}$$

However, for the vertical dynamics, we only apply the fundamental principle to the suspended mass of the carbody whose mass is $M_S = M_T - 4m_r$ (m_r is the mass of a wheel):

$$M_S(\dot{\vec{V}}.\vec{e}_z) = \sum_{i=1}^{4} F_{zi} + P_z + F_{aero_z}$$

P is the weight of the vehicle and is given by Equations (A.2):

$$P_x = M_T g \sin(\phi - p_x) \tag{A.2a}$$

$$P_y = -M_T g \sin(\theta - p_y)\cos(\phi - p_x) \tag{A.2b}$$

$$P_z = -M_S g \cos(\theta - p_y)\cos(\phi - p_x) \tag{A.2c}$$

F_{aero} is the aerodynamic drag forces and is given by Equations (A.3):

$$F_{aero_x} = \frac{1}{2}\rho_{air} C_x S V_{aero}^2 \cos(\phi) \tag{A.3a}$$

$$F_{aero_y} = 0 \tag{A.3b}$$

$$F_{aero_z} = \frac{1}{2}\rho_{air} C_x S V_{aero}^2 \sin(\phi) \tag{A.3c}$$

The velocity of the vehicle $\vec{V}$ can be expressed in the vehicle coordinate system as following:

$$\vec{V} = V_x\vec{e}_x + V_y\vec{e}_y + V_z\vec{e}_z \tag{A.4}$$

Its derivative $\dot{\vec{V}}$ is expressed with respect to the inertial coordinate system, *i.e.* considering $(\vec{e}_X, \vec{e}_Y, \vec{e}_Z)$ as fixed. Thus, we have:

$$\dot{\vec{V}} = \dot{V}_x\vec{e}_x + V_x\dot{\vec{e}}_x + \dot{V}_y\vec{e}_y + V_y\dot{\vec{e}}_y + \dot{V}_z\vec{e}_z + V_z\dot{\vec{e}}_z \tag{A.5}$$

The mapping between the vehicle coordinate system $(\vec{e}_x, \vec{e}_y, \vec{e}_z)$ and the inertial coordinate system $(\vec{e}_X, \vec{e}_Y, \vec{e}_Z)$ is given by Equation (A.6)

$$\begin{pmatrix} e_x \\ e_y \\ e_z \end{pmatrix} = \mathrm{R}_\theta \mathrm{R}_\phi \mathrm{R}_\psi \begin{pmatrix} e_X \\ e_Y \\ e_Z \end{pmatrix} \tag{A.6}$$

where the expressions of the rotation matrices are given by Equations (A.7):

$$\mathrm{R}_\theta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \tag{A.7a}$$

$$\mathrm{R}_\phi = \begin{pmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{pmatrix} \tag{A.7b}$$

$$\mathrm{R}_\psi = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{A.7c}$$

Deriving with respect to time Equation (A.6) considering $(\vec{e}_X, \vec{e}_Y, \vec{e}_Z)$ as fixed, we obtain in the vehicle coordinate system:

$$\begin{aligned} \begin{pmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_z \end{pmatrix} &= \left(\dot{\mathrm{R}}_\theta \mathrm{R}_\phi \mathrm{R}_\psi + \mathrm{R}_\theta \dot{\mathrm{R}}_\phi \mathrm{R}_\psi + \mathrm{R}_\theta \mathrm{R}_\phi \dot{\mathrm{R}}_\psi\right) \mathrm{R}_\psi^{-1} \mathrm{R}_\phi^{-1} \mathrm{R}_\theta^{-1} \begin{pmatrix} e_x \\ e_y \\ e_z \end{pmatrix} \\ &= \begin{pmatrix} 0 & \dot{\psi}\cos\phi\cos\theta - \dot{\phi}\sin\theta & -\dot{\phi}\cos\theta - \dot{\psi}\cos\phi\sin\theta \\ \dot{\phi}\sin\theta - \dot{\psi}\cos\phi\cos\theta & 0 & \dot{\theta} - \dot{\psi}\sin\phi \\ \dot{\phi}\cos\theta + \dot{\psi}\cos\phi\sin\theta & \dot{\psi}\sin\phi - \dot{\theta} & 0 \end{pmatrix} \begin{pmatrix} e_x \\ e_y \\ e_z \end{pmatrix} \end{aligned} \tag{A.8}$$

The derivative of the rotational matrices are given by Equations (A.9):

$$\dot{\mathrm{R}}_\theta = \dot{\theta} \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\sin\theta & \cos\theta \\ 0 & -\cos\theta & -\sin\theta \end{pmatrix} \tag{A.9a}$$

$$\dot{\mathrm{R}}_\phi = \dot{\phi} \begin{pmatrix} -\sin\phi & 0 & -\cos\phi \\ 0 & 0 & 0 \\ \cos\phi & 0 & -\sin\phi \end{pmatrix} \tag{A.9b}$$

$$\dot{\mathrm{R}}_\psi = \dot{\psi} \begin{pmatrix} -\sin\psi & \cos\psi & 0 \\ -\cos\psi & -\sin\psi & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{A.9c}$$

Thus

$$\dot{\vec{\mathrm{V}}}.\vec{e}_x = \dot{\mathrm{V}}_x + \mathrm{V}_y(\dot{\phi}\sin\theta - \dot{\psi}\cos\phi\cos\theta) + \mathrm{V}_z(\dot{\phi}\cos\theta + \dot{\psi}\cos\phi\sin\theta) \tag{A.10a}$$

$$\dot{\vec{\mathrm{V}}}.\vec{e}_y = \dot{\mathrm{V}}_y + \mathrm{V}_x(\dot{\psi}\cos\phi\cos\theta - \dot{\phi}\sin\theta) + \mathrm{V}_z(\dot{\psi}\sin\phi - \dot{\theta}) \tag{A.10b}$$

$$\dot{\vec{\mathrm{V}}}.\vec{e}_z = \dot{\mathrm{V}}_z + \mathrm{V}_x(-\dot{\phi}\cos\theta - \dot{\psi}\cos\phi\sin\theta) + \mathrm{V}_y(\dot{\theta} - \dot{\psi}\sin\phi) \tag{A.10c}$$

Considering that ϕ and θ are small, Equations (A.10) are simplified to Equations (A.11)

$$\dot{\vec{\mathrm{V}}}.\vec{e}_x = \dot{\mathrm{V}}_x - \dot{\psi}\mathrm{V}_y + \dot{\phi}\mathrm{V}_z \tag{A.11a}$$

$$\dot{\vec{\mathrm{V}}}.\vec{e}_y = \dot{\mathrm{V}}_y + \dot{\psi}\mathrm{V}_x - \dot{\theta}\mathrm{V}_z \tag{A.11b}$$

$$\dot{\vec{\mathrm{V}}}.\vec{e}_z = \dot{\mathrm{V}}_z - \dot{\phi}\mathrm{V}_x + \dot{\theta}\mathrm{V}_y \tag{A.11c}$$

Finally, we end up with Equations (A.12) for the linear velocities of the carbody in the 10 DoF vehicle model after neglecting the terms $\dot{\phi}V_z$, $\dot{\theta}V_z$, $\dot{\theta}V_y$:

$$M_T(\dot{V}_x - V_y\dot{\psi}) = \sum_{i=1}^{4} F_{xi} - \frac{1}{2}\rho_{air}C_x S V_{aero}^2 \cos(\phi) + M_T g \sin(\phi - p_x) \tag{A.12a}$$

$$M_T(\dot{V}_y + V_x\dot{\psi}) = \sum_{i=1}^{4} F_{yi} - M_T g \sin(\theta - p_y)\cos(\phi - p_x) \tag{A.12b}$$

$$M_S(\dot{V}_z - V_x\dot{\phi}) = \sum_{i=1}^{4} F_{zi} - M_S g \cos(\theta - p_y)\cos(\phi - p_x) + \frac{1}{2}\rho_{air}C_x S V_{aero}^2 \sin(\phi) \tag{A.12c}$$

A.2 Equations for the angular velocities of a four-wheel vehicle model

A.2.1 Reminder about rigid body dynamics

Definition 1 (Angular Momentum). *The angular momentum* $\vec{L}_O$ *of a rigid body* $\mathscr{B}$ *in rotation around a point* O *that is fixed to an inertial frame* $\mathscr{R}_g$ *is defined as:*

$$\vec{L}_O = \int_{M\in\mathscr{B}} \overrightarrow{OM} \wedge \vec{V}(M \in \mathscr{B}) dm \tag{A.13}$$

where $\vec{V}(M \in \mathscr{B})$ *is the speed of a point* M *belonging to the rigid body* $\mathscr{B}$.

Definition 2 (Barycentric Angular Momentum). *The barycentric Angular Momentum* $\vec{L}^*$ *of a rigid body* $\mathscr{B}$ *in rotation around its center of inertia* G *is defined as:*

$$\vec{L}^* = \int_{M\in\mathscr{B}} \overrightarrow{GM} \wedge \vec{v}_M^* dm \tag{A.14}$$

where $\vec{v}_M^*$ *is the velocity of a point* M *expressed in the rigid body's coordinate frame* $\mathscr{R}'$.

Theorem 4 (König's theorem). *For a ridig body with total mass* $M = \int_{M\in\mathscr{B}} dm$:

$$\vec{L}_O = \overrightarrow{OG} \wedge M\vec{v}_G + \vec{L}^* \tag{A.15}$$

where $\vec{v}_G$ *is the velocity of the center of gravity in the inertial frame* $\mathscr{R}_g$.

Developing Equation (A.14) leads to the following expression for the barycentric Angular Momentum:

$$\vec{L}^* = I\vec{\Omega}_{\mathscr{R}'/\mathscr{R}_g} \tag{A.16}$$

where I is the inertia matrix of the rigid body $\mathscr{B}$ given by Equations (A.17) and $\vec{\Omega}_{\mathscr{R}'/\mathscr{R}_g}$ is the rotational speed between the rigid body's frame $\mathscr{R}'$ and the inertial frame $\mathscr{R}_g$ given by Equation (A.18).

$$I = \begin{pmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{pmatrix} \tag{A.17a}$$

$$= \begin{pmatrix} \int_{\mathscr{B}}(y^2+z^2)dm & -\int_{\mathscr{B}} xy dm & -\int_{\mathscr{B}} xz dm \\ -\int_{\mathscr{B}} xy dm & \int_{\mathscr{B}}(x^2+z^2)dm & -\int_{\mathscr{B}} yz dm \\ -\int_{\mathscr{B}} xz dm & -\int_{\mathscr{B}} yz dm & \int_{\mathscr{B}}(x^2+y^2)dm \end{pmatrix} \tag{A.17b}$$

$$\vec{\Omega}_{\mathscr{R}'/\mathscr{R}_g} = \begin{pmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{pmatrix} \tag{A.18}$$

The integral in Equation (A.17b) is made on all $M \in \mathscr{B}$, with (x, y, z) the coordinates of M in the rigid body's frame.

Theorem 5 (Angular Momentum theorem)**.** *The Angular Momentum theorem of a point-mass system in rotation around a fixed point* O *expressed in an inertial frame* $\mathscr{R}_g$ *links the angular momentum with the total torque applied on the system:*

$$\frac{d\vec{\mathrm{L}}_{\mathrm{O}}}{dt}|_{\mathscr{R}_g} = \sum_i \vec{\mathscr{M}}_{\mathrm{O}}(\vec{\mathrm{F}}_i) \tag{A.19}$$

where $\vec{\mathscr{M}}_{\mathrm{O}}(\vec{\mathrm{F}}_i)$ *is the moment generated by the force* $\vec{\mathrm{F}}_i$ *around the point* O *and* $\frac{d_{...}}{dt}|_{\mathscr{R}_g}$ *is the derivative with respect to time considering the inertial frame* $\mathscr{R}_g$ *as fixed.*

Theorem 6 (Barycentric Angular Momentum theorem)**.** *For a rigid body, the barycentric Angular Momentum theorem, where* G *is the center of inertia of the rigid body (even if it is not fixed in* $\mathscr{R}_g$*), leads to:*

$$\begin{aligned} \frac{d\vec{\mathrm{L}}_{\mathrm{G}}(\mathscr{B})}{dt}|_{\mathscr{R}_g} &= \frac{d\vec{\mathrm{L}}^*}{dt}|_{\mathscr{R}_g} \\ &= \sum_i \vec{\mathscr{M}}_{\mathrm{G}}(\vec{\mathrm{F}}_i^{ext}) \end{aligned} \tag{A.20}$$

where $\vec{\mathscr{M}}_{\mathrm{G}}(\vec{\mathrm{F}}_i^{ext})$ *is the moment generated by the external force* $\vec{\mathrm{F}}_i^{ext}$ *around the center of inertia* G.

Theorem 7 (Bour's formula)**.** *The relation between the derivatives with respect to time of a vector* $\vec{\mathrm{U}}$ *condisering two different fixed coordinate frame is given by Bour's formula:*

$$\frac{d\vec{\mathrm{U}}}{dt}|_{\mathscr{R}_g} = \frac{d\vec{\mathrm{U}}}{dt}|_{\mathscr{R}'} + \vec{\Omega}_{\mathscr{R}'/\mathscr{R}_g} \wedge \vec{\mathrm{U}} \tag{A.21}$$

A.2.2 Application to the vehicle

By applying Bour's formula to the barycentric angular momentum of the vehicle, we obtain:

$$\frac{d\vec{\mathrm{L}}^*}{dt}|_{\mathscr{R}_g} = \frac{d\vec{\mathrm{L}}^*}{dt}|_{\mathscr{R}'} + \vec{\Omega}_{\mathscr{R}'/\mathscr{R}_g} \wedge \vec{\mathrm{L}}^* \tag{A.22a}$$

$$= \mathrm{I}\dot{\vec{\Omega}}_{\mathscr{R}'/\mathscr{R}_g} + \vec{\Omega}_{\mathscr{R}'/\mathscr{R}_g} \wedge (\mathrm{I}\vec{\Omega}_{\mathscr{R}'/\mathscr{R}_g}) \tag{A.22b}$$

where $\mathrm{I}\dot{\vec{\Omega}}_{\mathscr{R}'/\mathscr{R}_g}$ is given by Equation (A.22c) and $\mathrm{I}\vec{\Omega}_{\mathscr{R}'/\mathscr{R}_g}$ by Equation (A.22d), neglecting the cross terms I_{xy}, I_{yz} and I_{xz}.

$$\mathrm{I}\dot{\vec{\Omega}}_{\mathscr{R}'/\mathscr{R}_g} = \begin{pmatrix} \mathrm{I}_x\ddot{\theta} \\ \mathrm{I}_y\ddot{\phi} \\ \mathrm{I}_z\ddot{\psi} \end{pmatrix} \tag{A.22c}$$

$$\mathrm{I}\vec{\Omega}_{\mathscr{R}'/\mathscr{R}_g} = \begin{pmatrix} \mathrm{I}_x\dot{\theta} \\ \mathrm{I}_y\dot{\phi} \\ \mathrm{I}_z\dot{\psi} \end{pmatrix} \tag{A.22d}$$

Thus:

$$\frac{d\vec{\mathrm{L}}^*}{dt}|_{\mathscr{R}_g} = \begin{pmatrix} \mathrm{I}_x\ddot{\theta} + (\mathrm{I}_z - \mathrm{I}_y)\dot{\phi}\dot{\psi} \\ \mathrm{I}_y\ddot{\phi} + (\mathrm{I}_x - \mathrm{I}_z)\dot{\theta}\dot{\psi} \\ \mathrm{I}_z\ddot{\psi} + (\mathrm{I}_y - \mathrm{I}_x)\dot{\theta}\dot{\phi} \end{pmatrix} \tag{A.23a}$$

$$\approx \begin{pmatrix} \mathrm{I}_x\ddot{\theta} \\ \mathrm{I}_y\ddot{\phi} \\ \mathrm{I}_z\ddot{\psi} \end{pmatrix} \tag{A.23b}$$

Remark 18. *If we want to take into account the fact that the roll center or pitch center* H *do not coincide with the center of gravity* G*, we need to use Huygens' theorem for the computation of the inertia matrix:*

$$\mathrm{I_H} = \mathrm{I_G} + \mathrm{M(GH)}^2 \tag{A.24}$$

However, due to the difficulty to get their accurate position, we assume that they coincide with the center of gravity G *in our simulation model.*

The barycentric angular momentum theorem leads to:

$$\frac{d\vec{\mathrm{L}}^*}{dt}|_{\mathscr{R}_g} = \sum_i \vec{\mathscr{M}}_{\mathrm{G}}(\vec{\mathrm{F}}_i^{ext}) \tag{A.25}$$

where the external forces are the normal reaction forces F_{zi}, the longitudinal and lateral tire forces F_{xi} and F_{yi} and the aerodynamic drag forces F_{aero}.

Remark 19. *Considering that the normal reaction forces at equilibrium induce no moment of force on the system, we can consider only their variation in the computation of the momentum applied to the system. This variation corresponds to the one of the suspension forces* $\Delta\mathrm{F}_{si}$*.*

Thus, by computing the moment of all theses forces around the center of gravity G and using Equation (A.23), we end up with Equation (A.26) which is the result we were looking for.

$$\mathrm{I}_x\ddot{\theta} = l_w(\Delta\mathrm{F}_{s1} + \Delta\mathrm{F}_{s3} - \Delta\mathrm{F}_{s2} - \Delta\mathrm{F}_{s4}) + \sum_{i=1}^{4} h_{\mathrm{CoG}}\mathrm{F}_{yi} \tag{A.26a}$$

$$\mathrm{I}_y\ddot{\phi} = -l_f(\Delta\mathrm{F}_{s1} + \Delta\mathrm{F}_{s2}) + l_r(\Delta\mathrm{F}_{s3} + \Delta\mathrm{F}_{s4}) - \sum_{i=1}^{4} h_{\mathrm{CoG}}\mathrm{F}_{xi} + h_{\mathrm{CoG}}\mathrm{F}_{aero} \tag{A.26b}$$

$$\mathrm{I}_z\ddot{\Psi} = l_f(\mathrm{F}_{y1} + \mathrm{F}_{y2}) - l_r(\mathrm{F}_{y3} + \mathrm{F}_{y4}) + l_w(\mathrm{F}_{x2} + \mathrm{F}_{x4} - \mathrm{F}_{x1} - \mathrm{F}_{x3}) \tag{A.26c}$$

A.3 Suspension displacements and forces

Let's proove that the displacement of the suspension i, denoted Δz_{si}, corresponds to Equation (A.27) for given roll θ and pitch ϕ angles:

$$\Delta z_{si}(\theta,\phi) = \epsilon_i l_w \sin\theta - l_i \cos\theta \sin\phi \tag{A.27}$$

with $\epsilon_i = \begin{cases} 1 & \text{if } i = \{2,4\} \\ -1 & \text{if } i = \{1,3\} \end{cases}$ and $l_i = \begin{cases} l_f & \text{if } i = \{1,2\} \\ -l_r & \text{if } i = \{3,4\} \end{cases}$.

Let's defined imaginary points attached to the carbody and located above the center of rotation of each wheel at the height of the center of gravity. In the vehicle frame, these points have the following coordinates:

$$z_{s_{fl}} = \begin{pmatrix} l_f \\ l_w \\ 0 \end{pmatrix} \quad ; \quad z_{s_{fr}} = \begin{pmatrix} l_f \\ -l_w \\ 0 \end{pmatrix} \quad ; \quad z_{s_{rl}} = \begin{pmatrix} -l_r \\ l_w \\ 0 \end{pmatrix} \quad ; \quad z_{s_{rr}} = \begin{pmatrix} -l_r \\ -l_w \\ 0 \end{pmatrix} \tag{A.28}$$

By applying successively the pitch and roll rotation matrices, respectively R_ϕ and R_θ defined in Equation (A.7), on these imaginary points, and projecting then the result onto the $\vec{e}_z$ axis, we obtain the displacement Δz_{si} of the suspension i:

$$\Delta z_{si} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \mathrm{R}_\theta \mathrm{R}_\phi z_{si} \tag{A.29}$$

Thus, we obtain:

$$\Delta z_{s_{fl}} = l_w \sin\theta - l_f \cos\theta \sin\phi \quad \text{(A.30a)}$$
$$\Delta z_{s_{fr}} = -l_w \sin\theta - l_f \cos\theta \sin\phi \quad \text{(A.30b)}$$
$$\Delta z_{s_{rl}} = l_w \sin\theta + l_r \cos\theta \sin\phi \quad \text{(A.30c)}$$
$$\Delta z_{s_{rr}} = -l_w \sin\theta + l_r \cos\theta \sin\phi \quad \text{(A.30d)}$$

Then deriving with respect to time:

$$\dot{\Delta z}_{s_{fl}} = l_w \dot{\theta} \cos\theta + l_f \dot{\theta} \sin\theta \sin\phi - l_f \dot{\phi} \cos\theta \cos\phi \quad \text{(A.31a)}$$
$$\dot{\Delta z}_{s_{fr}} = -l_w \dot{\theta} \cos\theta + lf \dot{\theta} \sin\theta \sin\phi - l_f \dot{\phi} \cos\theta \cos\phi \quad \text{(A.31b)}$$
$$\dot{\Delta z}_{s_{rl}} = l_w \dot{\theta} \cos\theta - l_r \dot{\theta} \sin\theta \sin\phi + l_r \dot{\phi} \cos\theta \cos\phi \quad \text{(A.31c)}$$
$$\dot{\Delta z}_{s_{rr}} = -l_w \dot{\theta} \cos\theta - l_r \dot{\theta} \sin\theta \sin\phi + l_r \dot{\phi} \cos\theta \cos\phi \quad \text{(A.31d)}$$

Remark 20. *At equilibrium (i.e.* $\theta = 0$, $\dot{\theta} = 0$,$\phi = 0$ *and* $\dot{\phi} = 0$*), we have* $\Delta z_{si} = 0$ *and* $\dot{\Delta z}_{si} = 0$ *for each suspension* i.

Remark 21. *The yaw rotation matrix* R_ψ *has already been applied to the whole vehicle but* R_θ *et* R_ϕ *are only applied to the carbody.*

The computation of the variation of the suspension forces ΔF_{si} and the normal reaction forces F_{zi} of wheel i are then obtained respectively by Equations (A.32) and (A.33). The latter is obtained by applying the fundamental principle of dynamics to the wheel i along the $\vec{e}_z$ axis, as shown on the right in Figure 2.7.

$$\Delta F_{si} = -k_s \Delta z_{si} - d_s \dot{\Delta z}_{si} \quad \text{(A.32)}$$

$$F_{zi} = \Delta F_{si} + F_{zi_0} \quad \text{(A.33)}$$

A.4 Normal reaction forces F_{zi_0} at equilibrium

By applying the fundamental principle of the dynamics along the vertical axes $\vec{e}_z$ in the vehicle frame at equilibrium, we obtain:

$$F_{z1_0} + F_{z2_0} + F_{z3_0} + F_{z4_0} - P = 0 \quad \text{(A.34a)}$$

where $P = M_T g$ is the total weight of the vehicle and F_{zi_0} is the normal reaction force at wheel i . Then the moment of inertia is zero along $\vec{e}_x$ and $\vec{e}_y$ axis at equilibrium. Thus:

$$l_w \left(F_{z1_0} + F_{z3_0} - F_{z2_0} - F_{z4_0}\right) = 0 \quad \text{(A.34b)}$$
$$l_r (F_{z3_0} + F_{z4_0}) - l_f (F_{z1_0} + F_{z2_0}) = 0 \quad \text{(A.34c)}$$

Moreover, as the distribution of the vehicle mass is supposed to be perfectly symmetric, we get $F_{z1_0} = F_{z2_0}$ and $F_{z3_0} = F_{z4_0}$. Hence:

$$F_{z1_0} = F_{z2_0} = \frac{l_r M_T g}{2(l_f + l_r)} \quad \text{(A.34d)}$$

$$F_{z3_0} = F_{z4_0} = \frac{l_f M_T g}{2(l_f + l_r)} \quad \text{(A.34e)}$$

A.5 Slip angle equations

The lateral slip-angle α_i of tire i is the angle between the direction given by the orientation of the wheel and the direction of the velocity of the wheel (see Figure A.1). There expressions are given by Equations (A.35).

$$\alpha_{fl} = \delta_f - \text{atan}\left(\frac{V_y + l_f\dot{\psi}}{V_x - l_w\dot{\psi}}\right) \tag{A.35a}$$

$$\alpha_{fr} = \delta_f - \text{atan}\left(\frac{V_y + l_f\dot{\psi}}{V_x + l_w\dot{\psi}}\right) \tag{A.35b}$$

$$\alpha_{rl} = -\text{atan}\left(\frac{V_y - l_r\dot{\psi}}{V_x - l_w\dot{\psi}}\right) \tag{A.35c}$$

$$\alpha_{rr} = -\text{atan}\left(\frac{V_y - l_r\dot{\psi}}{V_x + l_w\dot{\psi}}\right) \tag{A.35d}$$

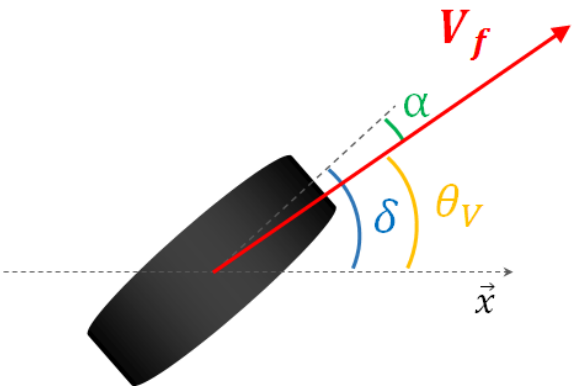


Figure A.1 – Slip angle and notations.

Proof. Set θ_{V_i} to be the angle between the speed vector $\vec{V}_i$ and the longitudinal axis $\vec{e}_x$ of the vehicle. Set C_{fl}, C_{fr}, C_{rl} and C_{rr} to be the center of rotation of each wheel (see Figure A.2). We have:

$$\tan(\theta_{V_i}) = \frac{V_y(C_i)}{V_x(C_i)} \tag{A.36}$$

The speed composition rule for a rigid body gives us the following relation where $\vec{\Omega}_{v/g} = \dot{\psi}\vec{e}_z$ represents the rotational speed between the vehicle and the inertial frame:

$$\vec{V}(C_i) = \vec{V}(G) + \overrightarrow{C_iG} \wedge \vec{\Omega}_{v/g} \tag{A.37}$$

Thus:

$$\vec{V}(C_{fl}) = \vec{V}(G) + \left(-l_f\vec{e}_x - l_w\vec{e}_y\right) \wedge \dot{\psi}\vec{e}_z \tag{A.38a}$$

$$\vec{V}(C_{fr}) = \vec{V}(G) + \left(-l_f\vec{e}_x + l_w\vec{e}_y\right) \wedge \dot{\psi}\vec{e}_z \tag{A.38b}$$

$$\vec{V}(C_{rl}) = \vec{V}(G) + \left(+l_r\vec{e}_x - l_w\vec{e}_y\right) \wedge \dot{\psi}\vec{e}_z \tag{A.38c}$$

$$\vec{V}(C_{rr}) = \vec{V}(G) + \left(+l_r\vec{e}_x + l_w\vec{e}_y\right) \wedge \dot{\psi}\vec{e}_z \tag{A.38d}$$

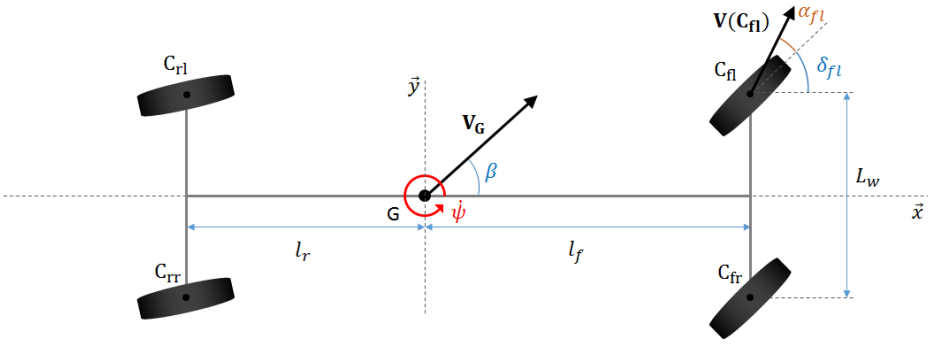


Figure A.2 – Top view of a four-wheel vehicle model.

Hence

$$\tan(\theta_{V_{fl}}) = \frac{V_y + l_f\dot{\psi}}{V_x - l_w\dot{\psi}} \tag{A.39a}$$

$$\tan(\theta_{V_{fr}}) = \frac{V_y + l_f\dot{\psi}}{V_x + l_w\dot{\psi}} \tag{A.39b}$$

$$\tan(\theta_{V_{rl}}) = \frac{V_y - l_r\dot{\psi}}{V_x - l_w\dot{\psi}} \tag{A.39c}$$

$$\tan(\theta_{V_{rr}}) = \frac{V_y - l_r\dot{\psi}}{V_x + l_w\dot{\psi}} \tag{A.39d}$$

Finally, as $\delta_i = \theta_{V_i} + \alpha_i$, we get the results given by Equation (A.35). □

Appendix B

Non-holonomic constraint and small-time controllability

B.1 The non-holonomic slip-free rolling condition

A kinematic constraint can be written in the form of Equation (B.1) where $a(q)$ is a vector and q the configuration of the robot:

$$a(q)^{\mathrm{T}}\dot{q} = 0 \tag{B.1}$$

In particular, this is the case of the slip-free rolling condition for a vehicle with $q = (\mathrm{X}, \mathrm{Y}, \psi)$ and $a(\mathrm{X}, \mathrm{Y}, \psi) = (-\sin\psi, \cos\psi, 0)^{\mathrm{T}}$.

Definition 3 (Holonomic constraint)**.** *A constraint is said to be holonomic if and only if it is integrable, i.e. that Equation (B.1) can be written:*

$$\alpha(q) = 0 \tag{B.2}$$

Remark 22. *If a constraint is holonomic, $\alpha(q)$ is constant along the trajectories $q(t)$ and is the first integral of the velocity field.*

$$\frac{d}{dt}\alpha(t) = \frac{\partial\alpha}{\partial q}(q)\dot{q} = 0 \quad \left(= a(q)^{\mathrm{T}}\dot{q}\right) \tag{B.3}$$

Suppose now that the system has m constraints of the form of Equation (B.1) that can be written:

$$\mathrm{A}(q)^{\mathrm{T}}\dot{q} = 0 \tag{B.4}$$

with $\mathrm{A}(q) = [a_1(q), a_2(q), ..., a_m(q)]$. We define n to be the number of configuration coordinates. Let's recall some definitions.

Definition 4 (Vector field)**.** *A vector field is a smooth map $f : \mathrm{CS} \to \mathrm{T}_q\mathrm{CS}$ which associates to each configuration q in the configuration space* CS *a vector in the tangent space* T_qCS *to q.*

Examples:

$$f : (\mathrm{X}, \mathrm{Y}, \psi)^{\mathrm{T}} \to (\cos\psi, \sin\psi, 0)^{\mathrm{T}} \tag{B.5a}$$
$$g : (\mathrm{X}, \mathrm{Y}, \psi)^{\mathrm{T}} \to (0, 0, 1)^{\mathrm{T}} \tag{B.5b}$$

f and g correspond respectively to the linear and angular velocity of the vehicle.

Definition 5 (Distribution)**.** *A distribution $\Delta = (f_1(q), ..., f_{n-m}(q))$ is a linear sub-space of vector fields.*

For a constraint defined by Equation (B.4), a distribution $(f_1(q),...,f_{n-m}(q))$ is given by the jth column of matrix S where:

$$\mathrm{A}(q)^{\mathrm{T}}\mathrm{S}(q) \quad = \quad 0 \tag{B.6}$$

Definition 6 (Lie bracket)**.** *Let f, g be two vector fields on the configuration space* CS*. The Lie bracket $[f,g]$ is the vector field defined by:*

$$[f,g] \quad = \quad \frac{\partial g}{\partial q}f - \frac{\partial f}{\partial q}g \tag{B.7}$$

The i-th component of the vector field $[f,g]$ is computed as following:

$$[f,g]_i \quad = \quad \sum_{j=1}^{n}\left(\frac{\partial f_i}{\partial q_j}g_j - \frac{\partial g_i}{\partial f_j}g_j\right) \tag{B.8}$$

Example:

$$\left[(\cos\psi,\sin\psi,0)^{\mathrm{T}},(\cos\psi,\sin\psi,1)^{\mathrm{T}}\right] \quad = \quad (-\sin\psi,\cos\psi,0)^{\mathrm{T}} \tag{B.9}$$

Definition 7 (Involutive distribution)**.** *A distribution $\Delta_q = (f_1(q),...,f_{n-m}(q))$ is involutive if it is closed under Lie brackets, i.e.:*

$$\forall i = 1,...,m, \forall j = 1,...,n-m, \quad \mathrm{L}_{f_j}f_i(q) \quad = \quad 0 \tag{B.10}$$

with $\mathrm{L}_{f_j}f_i(q) = [f_j,f_i](q)$.

Q1: Are the constraints holonomic?

Theorem 8 (Frobenius' theorem)**.** *A regular distribution Δ is integrable if and only if it is involutive, i.e.:*

$$\mathrm{rank}(\mathrm{LA}(\Delta)) \quad = \quad \mathrm{rank}(\Delta) \tag{B.11}$$

where LA *is the Lie Algebra operator.*

Therefore, Frobenius' theorem provides a very powerful formal method to test if the system is non-holonomic.

Application 1. *Non-holonomy of the slip-free rolling condition:*
$\Delta_q = (f_1(q) = (\cos\psi,\sin\psi,0)^{\mathrm{T}}, f_2(q) = (0,0,1)^{\mathrm{T}})$ *defines a distribution of the vector field in q. Thus* $\mathrm{rank}(\Delta_q) = 2$. *However:*

$$[f_1,f_2] \quad = \quad (-\sin\psi,\cos\psi,0)^{\mathrm{T}} \tag{B.12a}$$

Thus, $\mathrm{rank}(\mathrm{LA}(\Delta_q)) = 3$ *and the system is non-holonomic.*

Q2: If the system is non-holonomic, is it small-time controllable?

Definition 8 (locally controllable)**.** *A system is locally controllable from state ξ if the set of points reachable from ξ by an admissible trajectory contains a neighborhood of ξ.*

Definition 9 (small-time controllable)**.** *A system is small-time controllable from state ξ if the set of points reachable from ξ before a given time* T *contains a neighborhood of ξ for any* T.

The system is said to be small-time controllable if it is small-time controllable for every state ξ.

Theorem 9 (Lie Algebra Rank Condition)**.** *The system is small-time controllable from a configuration q if:*

$$\text{rank}(\text{LA}(\Delta_q)) \quad = \quad \dim(\text{CS}) \tag{B.13}$$

Application 2. *Small-time controllability of the slip-free rolling condition:*

- $\text{CS} = \mathbf{R}^2 \times [0; 2\pi[$ *thus* $\dim(\text{CS}) = 3$
- $\left(f_1(q) = (\cos\psi, \sin\psi, 0)^{\text{T}}, f_2(q) = (0,0,1)^{\text{T}}, [f_1, f_2](q) = (-\sin\psi, \cos\psi, 0)^{\text{T}}\right)$ *is a base for* $\text{LA}(\Delta_q)$. *Thus* $\forall q \in \text{CS}, \text{rank}(\text{LA}(\Delta_q)) = 3$.

Therefore, the Lie Algebra Rank Condition is true and the system is small-time controllable.

Lastly, from any configuration q, it is possible to find an open neighborhood $\mathcal{V}(q)$, reachable from q as the system is small-time controllable. Then, from a configuration $q' \in \mathcal{V}(q)$, it is possible to find an open neighborhood $\mathcal{V}(q')$, reachable from q' and so forth. Therefore, if there exists a collision free path from configuration q_{init} to q_{goal} with no constraints on the motion and no contact, then there exists a set of open sets joining q_{init} to q_{goal} where the system is small-time controllable. Finally, applying the Heine-Borel theorem, there exists a finite number of open sets joining q_{init} to q_{goal} which can thus be obtained using a computer.

Remark 23. *More literature on Lie Algebra, non-holonomic constraints and small-time controllability can be found [Sussmann, 1982], [d'Andréa Novel, 1995], [Laumond et al., 1998].*

Appendix C

Model-free Control

C.1 Deriving the ultralocal model

Let's consider a system where y is the output and u the input. It can be written as an input-output differential equation in the form of Equation (C.1) where $\mathscr{E}$ is a function not necessarily linear but assumed to be sufficiently smooth.

$$\mathscr{E}(t, y, \dot{y}, ..., y^{(m)}, u, \dot{u}, ..., u^{(n)}) = 0 \tag{C.1}$$

If there exist $\nu \in]0; m[$ such that $\frac{\partial \mathscr{E}}{\partial y^{(\nu)}} \neq 0$, the implicit function theorem yields then locally to:

$$y^{(\nu)} = \mathscr{E}(t, y, \dot{y}, ..., y^{(\nu-1)}, y^{(\nu+1)}, ..., y^{(m)}, u, \dot{u}, ..., u^{(n)}) \tag{C.2}$$

In practice, ν is always equal to 1 or 2.

C.2 ALIEN filters

ALIEN filters were introduced in [Fliess and Sira-Ramírez, 2003]. They are based on operational calculus [Yoshida, 1984].

Given a polynomial signal $y(t) = a_0 + a_1 t + \frac{a_2}{2} t^2 + ...$ in the time domain, we can derive an ALIEN filter at any order n using the following technique:

1. Transform the signal in the operational domain using Laplace transform:

$$\mathrm{Y}(s) = \frac{a_0}{s} + \frac{a_1}{s^2} + \frac{a_2}{s^3} + ... \tag{C.3}$$

2. Get rid of the initial conditions by multiplying by s^{m_1} and deriving m_2 times with respect to s.

3. Multiply by s^{-p} for noise attenuation, in order to have all the terms in $\mathrm{Y}(s)$ and its derivatives $\frac{d^n \mathrm{Y}}{ds^n}$ to be in the form $\frac{1}{s^\alpha} \frac{d^n \mathrm{Y}}{ds^n}$ with $\alpha \geq 1$.

4. Apply the inverse Laplace transform given by Equations (C.4) and (C.5) in order to get the result in the time domain.

$$\frac{c}{s^\alpha}, \alpha \geq 1, c \in \mathbb{C} \quad \Rightarrow \quad c \frac{t^{\alpha-1}}{(\alpha-1)!}, t > 0 \tag{C.4}$$

$$\frac{1}{s^\alpha} \frac{d^n \mathrm{Y}}{ds^n}, \alpha \geq 1 \quad \Rightarrow \quad \frac{(-1)^n}{(\alpha-1)!} \int_0^{\mathrm{T}} (\mathrm{T} - \tau)^{\alpha-1} \tau^n y(\tau) \, \mathrm{d}\tau \tag{C.5}$$

Application 3. *Signal at order* 0*:*
Starting with $y(t) = a_0 + a_1 t$*, apply the Laplace transform:*

$$\mathrm{Y}(s) = \frac{a_0}{s} + \frac{a_1}{s^2} \tag{C.6}$$

Then we apply step 2 for denoising the signal, multiplying first by s^2 *in Equation (C.7a) then deriving twice in Equations (C.7b) and (C.7c):*

$$s^2\mathrm{Y}(s) = sa_0 + a_1 \tag{C.7a}$$

$$\frac{d}{ds}(s^2\mathrm{Y}(s)) = a_0 \tag{C.7b}$$

$$\frac{d}{ds}(s^2\mathrm{Y}(s)) = 2s\mathrm{Y}(s) + s^2\frac{d\mathrm{Y}}{ds} \tag{C.7c}$$

We end up with Equation (C.8)

$$a_0 = 2s\mathrm{Y}(s) + s^2\frac{d\mathrm{Y}}{ds} \tag{C.8}$$

Thus, after doing the inverse Laplace transform:

$$a_0 = \frac{2}{\mathrm{T}^2}\int_0^{\mathrm{T}} (2\mathrm{T} - 3\tau)\, y(\tau)\, d\tau \tag{C.9}$$

Therefore, we obtain Equations (C.10-C.12) for respectively a signal at order 0, 1 and 2:

$$y(t) = a_0 = \frac{2}{\mathrm{T}^2}\int_0^{\mathrm{T}} (2\mathrm{T} - 3\tau)\, y(\tau)\, d\tau \tag{C.10}$$

$$y'(t) = a_1 = -\frac{6}{\mathrm{T}^3}\int_0^{\mathrm{T}} (\mathrm{T} - 2\tau)\, y(\tau)\, d\tau \tag{C.11}$$

$$y''(t) = a_2 = \frac{60}{\mathrm{T}^5}\int_0^{\mathrm{T}} (\mathrm{T}^2 - 6\mathrm{T}\tau + 6\tau^2)\, y(\tau)\, d\tau \tag{C.12}$$

Application 4. *Estimating* $\widehat{\mathrm{F}}$ *(for* $\nu = 1$*):*
In order to obtain the ALIEN filter for estimating the dynamic $\widehat{\mathrm{F}}$ *of a first-order system given by Equation (C.13), the same procedure is applied starting from* $\dot{y} = \mathrm{F} + \alpha u$ *and assuming that* F *is a piecewise constant function.*

$$\widehat{\mathrm{F}} = -\frac{6}{\mathrm{T}^3}\int_0^{\mathrm{T}} [(\mathrm{T} - 2\tau)\, y(\tau) + \alpha\tau(\mathrm{T} - \tau)\, u(\tau))]\, \mathrm{d}\tau \quad \textit{if } \nu = 1 \tag{C.13}$$

After applying the Laplace transform, we obtain Equation (C.14)

$$s\mathrm{Y}(s) = \frac{\mathrm{F}_s}{s} + \alpha\mathrm{U}(s) + y(0) \tag{C.14}$$

Then, we get rid of the initial condition $y(0)$ *by multiplying by* $\frac{d}{ds}$ *on both sides. In order to obtain an equation only composed of terms of the form given by Equations (C.4) and (C.5) for which the inverse Laplace transforms are known, multiply both side by* s^{-2}*. We end up with Equation (C.15).*

$$\frac{\mathrm{Y}(s)}{s^2} + \frac{1}{s}\frac{d\mathrm{Y}}{ds} = -\frac{\mathrm{F}_s}{s^4} + \alpha\frac{1}{s^2}\frac{d\mathrm{U}}{ds} \tag{C.15}$$

Applying Equations (C.4) and (C.5) to Equation (C.15) leads to Equation (C.13).

Similarly, for a second-order system, starting from $\ddot{y} = \mathrm{F} + \alpha u$ one obtain Equation (C.16):

$$\widehat{\mathrm{F}} = -\frac{5!}{2\mathrm{T}^5}\int_0^{\mathrm{T}} [(-\mathrm{T}^2 + 6\mathrm{T}\tau - 6\tau^2)\, y(\tau) + \frac{\alpha}{2}\tau^2(\mathrm{T} - \tau)^2 u(\tau))]\, \mathrm{d}\tau \quad \text{if } \nu = 2 \tag{C.16}$$

C.3 Numerical quadrature of ALIEN filters using the trapezoidal rule

Equation (C.13) is an analog equation which needs to be approximated numerically. A common approach used to discretize an integral is the *trapezoidal rule*. Let's define $N_{buf} = \lfloor T/T_s \rfloor + 1$. Set $Y_k = y(t_k)$ and $U_k = u(t_k)$ with $t_k = kT_s$, for $k \in [\![0; N_{buf} - 1]\!]$. The trapezoidal rule applied to Equation (C.13) gives Equation (C.17).

- Trapezoidal rule:

$$-\frac{6}{T^3}\int_0^T [(T-2\tau)y(\tau)+\alpha\tau(T-\tau)u(\tau))]\,d\tau \approx \tag{C.17}$$
$$-\frac{6T_s}{T^3}\sum_{k=0}^{N-2}\left[(T-2(k+\frac{1}{2})T_s)\left(\frac{Y_k+Y_{k+1}}{T_s}\right)\right]$$
$$-\frac{6T_s}{T^3}\sum_{k=0}^{N-2}\left[(k+\frac{1}{2})T_s(T-(k+\frac{1}{2})T_s)\left(\frac{\alpha_{k+1}U_{k+1}+\alpha_k U_k}{T_s}\right)\right]$$

At steady-state, $\dot{y} = 0$ and u is constant, thus $F = -\alpha u$. The analog expression of the ALIEN filter given by Equation (C.18) leads to:

$$\widehat{F}_{filt} = -\frac{6}{T^3}\int_0^T [(T-2\tau)y+\tau(T-\tau)\alpha u)]\,d\tau \quad = \quad -\alpha u \tag{C.18}$$

However, the numerical quadrature of the integral using the trapezoidal rule given by the right-hand side of Equation (C.17) leads to an error. More precisely:

$$\widehat{F}_{num} \quad = \quad -\alpha u - \frac{1}{2}\left(\frac{T_s}{T}\right)^2 \alpha u \tag{C.19}$$

C.4 Decomposition of the ALIEN filter for a first-order system

Let's define $N_{buf} = \lfloor T/T_s \rfloor + 1$, the number of stored measurement for the filtering.

Lemma 1. *Given* $N_{buf} \geq 3$, *there exist* $m \in \mathbb{N}^*$ *and* $(t_{k_0}, t_{k_1}, ..., t_{k_m}) \in (\mathbb{R}^+)^{m+1}$ *such that the three following properties hold (with* $T = (N_{buf} - 1)T_s$*):*

- *P1:* $0 = t_{k_0} < t_{k_1} < t_{k_2} < ... < t_{k_m} = T$
- *P2:* $\forall i \in [\![1; m]\!], \frac{t_{k_{i+1}} - t_{k_i}}{T_s} = \{2, 3\}$
- *P3:* $\forall i \in [\![1; m]\!], \mod(t_{k_i}, T_s) = 0$

Proof. Let's proof this lemma by mathematical induction:

- Initialization
  For $N_{buf} = 3$ and $N_{buf} = 4$, set $m = 1$.
- Heredity: Let's proof that Lemma 1 holds for $N_{buf} + 1$.
  Suppose that $\forall n \in [\![3; N_{buf}]\!]$, Lemma 1 is true. In particular, for $n = N_{buf} - 1$, there exist an $m \in \mathbb{N}^*$ and $(t_{k_0}, t_{k_1}, ..., t_{k_m}) \in (\mathbb{R}^+)^{m+1}$ that respect the properties P1, P2 and P3.

  Set $m' = m + 1$ and define $(t'_{k_0}, t'_{k_1}, ..., t'_{k_{m'}}) \in (\mathbb{R}^+)^{m'+1}$ such that $\forall i \in [\![1; m]\!], t'_{k_i} = t_{k_i}$ and $t'_{k_{m'}} = T = N_{buf}T_s$.

  We have

  - $0 = t'_{k_0} < t'_{k_1} < t'_{k_2} < ... < t'_{k_{m'}} = T$;
  - $\forall i \in [\![1; m']\!], \mod(t'_{k_i}, T_s) = 0$;

– $\forall i \in [\![1; m']\!], \frac{t'_{k_{i+1}} - t'_{k_i}}{\mathrm{T}_s} = \{2,3\}$ and $\frac{t_{k'_m} - t_{k_{m'-1}}}{\mathrm{T}_s} = 2$.

which concludes the proof.

□

Theorem 10. *For all* $\mathrm{N}_{buf} \geq 3$*, there exists a third-order numerical quadrature of Equation (6.3).*

Proof. According to Lemma 1, there exists $m \in \mathbb{N}^*$ and $(t_{k_0}, t_{k_1}, ..., t_{k_m}) \in (\mathbb{R}^+)^{m+1}$ verifying the properties P1, P2 and P3.
As $0 = t_{k_0} < t_{k_1} < t_{k_2} < ... < t_{k_m} = \mathrm{T}$, the integral $\int_0^{\mathrm{T}}$ can be expressed as:

$$\int_0^{\mathrm{T}} = \int_{t_{k_0}}^{t_{k_1}} + ... + \int_{t_{k_i}}^{t_{k_{i+1}}} + ... + \int_{t_{k_{m-1}}}^{t_{k_m}}$$

where each subintegral $\int_{t_{k_i}}^{t_{k_{i+1}}}$ is well defined and can be computed as $\forall i \in [\![0; m]\!], mod(t_{k_i}, \mathrm{T}_s) = 0$.
Finally, as $\forall i \in [\![1; m]\!], \frac{t_{k_{i+1}} - t_{k_i}}{\mathrm{T}_s} = \{2,3\}$, a third-order numerical quadrature can be obtained using Equation (6.15) if $\frac{t_{k_{i+1}} - t_{k_i}}{\mathrm{T}_s} = 2$ and Equation (6.16) if $\frac{t_{k_{i+1}} - t_{k_i}}{\mathrm{T}_s} = 3$. □

Remark 24. *A second-order system cannot systematically be decomposed into subintervals where Equation (6.17) can be applied.*

C.5 Bound on the tracking error

Let $\delta > 0$. Choosing $\mathrm{K_P}$ and α such that $|\mathrm{A}| < 1$, i.e. $\frac{2}{\mathrm{T}_s} \geq \alpha \mathrm{K_P} \geq 0$, leads to $|\mathrm{A}|^k \underset{k \to +\infty}{\longrightarrow} 0$.
Therefore,

$$\forall \epsilon > 0, \exists \mathrm{K} \in \mathbb{N}^*, \forall k \in \mathbb{N}^*, \quad k \geq \mathrm{K} \Rightarrow |\mathrm{A}|^k < \epsilon \tag{C.20}$$

Hence, Equation (6.29) leads to:

$$\forall \epsilon > 0, \exists \mathrm{K} \in \mathbb{N}^*, \forall k \geq \mathrm{K}, \quad |e_k| < \epsilon |e_0| + (1+\epsilon) \left| \frac{f_{max}}{\alpha \mathrm{K_P}} \right| \tag{C.21}$$

In particular, choosing $\epsilon = \frac{1}{\left(1 + |e_0| \left| \frac{\alpha \mathrm{K_P}}{f_{max}} \right|\right)} \delta > 0$ which is defined as $f_{max} \neq 0$ leads to the presented results. Thus:

$$\forall \delta > 0, \exists \mathrm{K} \in \mathbb{N}^*, \forall k \in \mathbb{N}^*, k \geq \mathrm{K}, \quad |e_k| \leq (1+\delta) \left| \frac{1}{\alpha \mathrm{K_P}} \right| f_{max} \tag{C.22}$$

Appendix D

Résumé en francais

Chapitre 1 : Introduction

Aujourd'hui, la révolution numérique transforme la mobilité, tout comme la révolution industrielle a pu le faire au début du XXème siècle. En particulier, la voiture qui était un objet mécanique par excellence au début des années 1980, est devenue au fil du temps de plus en plus semblable à un appareil électronique. De nombreux systèmes d'aide à la conduite, désignés par l'acronyme ADAS (pour Advanced Driver Assistance Systems), équipent déjà les voitures haut de gamme afin d'assurer une meilleure prévention des accidents, si bien qu'aujourd'hui, la voiture autonome n'est plus une utopie mais un objet réel qui pourrait faire partie du monde de demain, en particulier quand on observe que plus de 94% des accidents de la route sont dus à des erreurs humaines.

La voiture autonome est un robot sur roues : ainsi, elle est composée des quatre primitives de la robotique, à savoir la perception, la modélisation du monde environnant, la planification et l'action. Ces tâches sont plus ou moins complexes en fonction de la situation et peuvent demander une forte ou une faible capacité cognitive. Alors que les ordinateurs surpassent les humains en performance dans le second cas de figure, le premier cas est encore dominé par les humains... mais pour combien de temps encore ? Cette thèse porte exclusivement sur les tâches de planification et d'action, que nous nommerons par la suite respectivement planification du mouvement et contrôle-commande. En particulier, nous nous intéressons à la cohérence entre les deux, c'est-à-dire l'impact qu'a la planification sur le contrôle et inversement, surtout en termes de niveau de modélisation et d'abstraction du comportement de la voiture. Ainsi, nous montrerons tout d'abord comment garantir qu'une trajectoire de référence calculée par le planificateur de mouvement est faisable à la fois cinématiquement et dynamiquement. Dans un second temps, nous comparerons les performances de contrôleurs latéraux communément utilisés dans la littérature pour savoir quel type de trajectoires de référence ils sont capables de suivre. Enfin, le dernier chapitre de cette thèse, indépendant du reste, abordera la commande sans modèle. Cette technique récente, introduite dans Fliess et Join (2013), apparaît comme prometteuse car elle demande peu de connaissances sur le système. Les questions de son implémentation sur une vraie voiture et des garanties obtenues sont abordées.

Chapitre 2 : Modélisation du véhicule

Afin de tester les algorithmes de planification du mouvement et de contrôle-commande, nous avons développé un simulateur de la dynamique du véhicule avec un haut degré de réalisme. Pour cela, nous nous sommes intéressés à la fois aux différents modèles dynamiques du châssis, à savoir le modèle bicyclette et le modèle à quatre roues, et aux différents modèles de pneumatique (Pacejka, Dugoff, linéaire, LuGre, Burckhardt-Kiencke, Fiala) pour représenter la force générée par la route sur la roue. L'implémentation du modèle à 10 degrés de liberté (10 DDL) choisi pour les simulations est ensuite détaillée. Les 10 DDL sont les vitesses longitudinales, latérales et verticales, les

vitesses de rotation des angles de roulis, tangage et lacet, et les vitesses de rotations de chacune des quatre roues.

Enfin, ce chapitre présente différents modèles simplifiés pour représenter le mouvement d'un véhicule lors des étapes de planification et de contrôle, que ce soit le point-matériel, modèle le plus simplifié, le modèle unicycle ou encore le modèle bicyclette cinématique.

Chapitre 3 : Planification du mouvement pour la voiture autonome

Pour calculer numériquement un chemin ou une trajectoire pour une voiture qui évite les obstacles, il existe principalement deux types de méthodes. La première méthode consiste à résoudre le problème dans l'espace des configurations qui correspond à l'ensemble des configurations que peut occuper la voiture (une configuration détermine l'espace occupé par la voiture de manière unique). La deuxième méthode repose sur l'espace des contrôles et utilise un modèle du comportement de la voiture pour prédire son évolution.

Résoudre le problème dans l'espace des configurations revient à capturer les composantes connexes de ce dernier afin de savoir si deux configurations appartiennent à la même composante et donc peuvent être reliées. Pour cela, on peut utiliser soit des méthodes d'échantillonnage déterministes de l'espace de configuration (ex : décomposition cellulaire, décomposition trapézoïdale, triangulation de Delaunay, graphe de visibilité), soit des méthodes d'échantillonnage probabilistes (ex : Probabilistic Roadmap Method, Rapidly-exploring Random Tree). Dans les deux cas de figure, un algorithme de recherche de chemin dans un graphe (ex : A-star, Dijkstra) est ensuite utilisé pour trouver le “meilleur" chemin.

Résoudre le problème dans l'espace des contrôles consiste le plus souvent à formuler le problème comme un problème de commande optimale telle la commande prédictive à modèle. Il s'agit alors de minimiser une fonction de coût sur un horizon glissant en respectant des contraintes sur le système.

Cependant, un véhicule possède certaines caractéristiques en termes de cinématique et dynamique afin de trouver un chemin ou une trajectoire reliant un état initial à un état final. En particulier, les contraintes non-holonomes de roulement sans glissement limitent les possibilités de déplacement du véhicule, ce qu'il faut prendre en compte dans la planification du mouvement ; cette question a été traitée de façon relativement exhaustive dans la littérature. En revanche, les aspects dynamiques, notamment liés aux dérapages et aux glissements des roues sur le sol, ont reçu peu d'attention au sein de la communauté de planification du mouvement, se limitant le plus souvent à considérer le cercle de friction. Cependant, cette situation pourrait mener à des situations imprévues notamment avec l'augmentation de la vitesse des véhicules et la baisse d'adhérence de la route due aux conditions météorologiques ou au revêtement de la chaussée.

Chapitre 4 : Garantir la faisabilité dynamique d'un modèle bicyclette cinématique

Dans ce chapitre, nous proposons d'établir une architecture de planification et de contrôle pour le véhicule autonome qui soit non seulement cohérente mais aussi efficace d'un point de vue trafic. Le mot “cohérence" réfère ici à la garantie que la trajectoire de référence à suivre soit toujours faisable à la fois cinématiquement et dynamiquement et que les contrôleurs bas-niveau soient capables de suivre l'intention du véhicule. Partant du constat qu'un humain conduit la plus part du temps en respectant le modèle bicyclette cinématique, où le dérapage des roues peut être négligé, nous nous sommes intéressés aux limites de validité de ce modèle. Pour cela, nous avons comparé ce modèle au modèle à haut degré de réalisme (10 DDL) présenté dans le chapitre 2 prenant en compte les dérapages et les glissements, ainsi que les transferts de charge. Les résultats obtenus nous montrent que le modèle bicyclette cinématique est une très bonne représentation du mouvement d'un véhicule quand l'accélération latérale est en dessous d'un certain seuil qui dépend

des conditions de la route. En revanche, au-dessus de cette valeur maximale, les erreurs de modèle deviennent très grandes. Cette contrainte sur l'accélération latérale peut être exprimée comme une contrainte dynamique sur l'angle au volant : ce dernier est limité par une valeur maximale qui dépend alors de la vitesse du véhicule. Après avoir obtenu ces résultats sur le modèle bicyclette cinématique, une architecture de planification et de contrôle pour le véhicule autonome est proposée. Elle repose sur quatre modules principaux : (i) un planificateur local de vitesse ; (ii) un gestionnaire d'obstacles ; (iii) un planificateur de trajectoire reposant sur la commande prédictive à modèle (planificateur MPC) avec un modèle bicyclette cinématique ; (iv) des contrôleurs longitudinaux et latéraux.

Le planificateur local de vitesse permet de rendre l'architecture du véhicule moins conservatrice en calculant une vitesse heuristique qui servira de référence au planificateur de trajectoire MPC. Cette vitesse heuristique dépend à la fois de la vitesse actuelle du véhicule mais aussi de la géométrie de la route à venir. Le gestionnaire d'obstacles construit une parabole autour de chaque obstacle. Le planificateur MPC calcule ensuite une trajectoire de référence qui évite les paraboles définies précédemment. La vitesse de référence que calcule le MPC est guidée par la vitesse heuristique définie précédemment. Le planificateur MPC est soumis à des contraintes liées au mouvement du véhicule : ce dernier doit non seulement respecter le modèle bicyclette cinématique mais aussi la contrainte dynamique sur l'angle au volant mentionnée précédemment. Ainsi, les trajectoires obtenues sont assurées d'être dynamiquement faisables et de plus, le choix du contrôleur devient plus facile car il suffit que ce dernier repose sur un modèle similaire. Les contrôleurs se décomposent en un contrôleur longitudinal proportionnel-intégral-dérivatif (PID) et en un contrôleur latéral comprenant un terme en boucle ouverte (angle au volant calculé par le planificateur MPC) et un terme en boucle fermée (PID sur l'erreur d'orientation du véhicule anticipée de 0.2s).

Cette architecture est ensuite testée en simulation sur un circuit difficile, alternant entre longues lignes droites et virages très serrés, afin de montrer son efficacité et sa capacité à générer des trajectoires faisables. Enfin, une adaptation aux routes mouillées et enneigées est proposée. Cette adaptation ne modifie pas les gains des planificateurs de mouvement et des contrôleurs ce qui la rend plus facile à implémenter en situation réelle.

Chapitre 5 : Contrôleurs pour voitures autonomes

L'objectif des contrôleurs pour la voiture autonome est de suivre une trajectoire ou un chemin de référence en actionnant les pédales d'accélération et de frein, ainsi que le volant de la voiture. La plupart du temps, le contrôle longitudinal qui est responsable du suivi de la vitesse et/ou de l'accélération de référence est traité indépendamment du contrôle latéral, dédié au suivi du chemin : on parle alors de contrôleurs découplés. Cependant, dans des situations plus complexes telles les manœuvres d'urgence, les couplages des dynamiques longitudinales et latérales nécessitent de considérer les deux problèmes simultanément : on parle alors de contrôleurs couplés. Après une première partie succincte sur le contrôle longitudinal, différents contrôleurs latéraux communément utilisés dans la littérature (en particulier les contrôleurs Stanley, pure-poursuite et forme chaînée reposant sur le modèle bicyclette cinématique) sont présentés et comparés en situation réelle sur le campus de l'Université de Shanghai Jiao-Tong. Les conclusions de cette expérience sont qu'il n'existe pas de "meilleur" contrôleur mais différents contrôleurs adaptés à différentes situations : par exemple, le contrôleur qui repose sur la forme chaînée a une précision inférieure à 30cm plus de 95% du temps ; en revanche, contrairement au contrôleur pure-poursuite, il n'est ni très confortable en ligne droite, ni très précis dans les virages serrés ; le contrôleur pure-poursuite a une précision inférieure à 30cm seulement 50% du temps. Enfin, la dernière section de ce chapitre présente un état de l'art des contrôleurs couplés.

Chapitre 6 : Commande sans modèle

Dans ce chapitre, indépendant du reste de la thèse, nous nous sommes intéressés à la commande sans modèle telle qu'elle a été introduite par Fliess et Join (2013) et notamment à son application à la conduite automatisée. Après une présentation de la théorie sous-jacente et des avantages que présente cette méthode, la question de son implémentation en pratique est posée. En particulier, la discrétisation des filtres ALIEN introduits par Sira-Ramirez et Fliess (2003) et qui permettent d'estimer la dynamique du véhicule est étudiée. De plus, une méthode systématique et simple pour régler le paramètre α du contrôleur est proposée. La stabilité pour les systèmes supposés affines en la variable de contrôle est analysée en cas d'estimation imparfaite (ce qui est toujours le cas en pratique). Il en ressort que pour que la stabilité du système soit garantie, il faut que l'erreur d'estimation reste bornée, avec une borne aussi petit que possible. Cela peut être obtenu d'une part par le bon choix de l'ordre de discrétisation du filtre qui permet de garantir une erreur d'estimation nulle en régime permanent et d'autre part, par le bon réglage du paramètre α, ce dernier ayant une grande influence sur la qualité du suivi (et donc sur l'erreur d'estimation) pendant les phases transitoires. Les solutions proposées sont ensuite testées en conditions réelles sur la voiture autonome de l'ENSIAME, à la fois pour la gestion du freinage et pour le suivi d'un profil de vitesse référence. Une ouverture sur le contrôle sans modèle latéral est proposée.

Chapitre 7 : Conclusion

Cette thèse défend qu'une condition sine qua non pour assurer le déploiement à grande échelle des véhicules autonomes est de pouvoir garantir à tout moment que l'action effectivement réalisée par la voiture corresponde à son intention. Cette garantie essentielle à la sécurité du véhicule devient d'autant plus importante en cas de conduite coopérative où les intentions sont échangées entre véhicules. Les sources d'incohérence entre l'action et l'intention d'un véhicule peuvent provenir majoritairement de deux facteurs : la non-faisabilité de l'intention due à l'utilisation d'un modèle trop simplifié et par conséquent non valide dans la plage considérée; le choix d'un contrôleur inapproprié pour la réalisation de l'intention.

Cette thèse ouvre des perspectives diverses. Tout d'abord, les travaux sur la validité du modèle bicyclette cinématique peuvent être étendus au modèle unicycle. Cela permettra peut-être à terme d'établir un cadre unique pour répondre aux problématiques de prise de décision et de faisabilité dynamique. Ensuite, l'intégration du machine learning pour apprendre à « simplifier » des problèmes complexes tels la prise de décision ou le couplage de la dynamique permettrait de gagner en efficacité algorithmique. Enfin, l'intégration au niveau de la planification des imperfections des contrôleurs permettrait de renforcer l'adéquation entre l'intention et l'action.

Résumé

La voiture autonome pourrait réduire le nombre de morts et de blessés sur les routes tout en améliorant l'efficacité du trafic. Cependant, afin d'assurer leur déploiement en masse sur les routes ouvertes au public, leur sécurité doit être garantie en toutes circonstances. Cette thèse traite de l'architecture de planification et de contrôle pour la conduite automatisée et défend l'idée que l'intention du véhicule doit correspondre aux actions réalisées afin de garantir la sécurité à tout moment. Pour cela, la faisabilité cinématique et dynamique de la trajectoire de référence doit être assurée. Sinon, le contrôleur, aveugle aux obstacles, n'est pas capable de la suivre, entraînant un danger pour la voiture elle-même et les autres usagers de la route. L'architecture proposée repose sur la commande à modèle prédictif fondée sur un modèle bicyclette cinématique afin de planifier des trajectoires de référence sûres. La faisabilité de la trajectoire de référence est assurée en ajoutant une contrainte dynamique sur l'angle au volant, contrainte issue de ces travaux, afin d'assurer que le modèle bicyclette cinématique reste valide. Plusieurs contrôleurs à haute-fréquence sont ensuite comparés afin de souligner leurs avantages et inconvénients. Enfin, quelques résultats préliminaires sur les contrôleurs à base de commande sans modèle et leur application au contrôle automobile sont présentés. En particulier, une méthode efficace pour ajuster les paramètres est proposée et implémentée avec succès sur la voiture expérimentale de l'ENSIAME en partenariat avec le laboratoire LAMIH de Valenciennes.

Mots Clés

Planification de trajectoire, Conduite autonome, Commande prédictive à modèle, contrôle longitudinal et latéral, dynamique du véhicule

Abstract

Autonomous vehicles are believed to reduce the number of deaths and casualties on the roads while improving the traffic efficiency. However, before their mass deployment on open public roads, their safety must be guaranteed at all time. Therefore, this thesis deals with the motion planning and control architecture for autonomous vehicles and claims that the intention of the vehicle must match with its actual actions. For that purpose, the kinematic and dynamic feasibility of the reference trajectory should be ensured. Otherwise, the controller which is blind to obstacles is unable to track it, setting the ego-vehicle and other traffic participants in jeopardy. The proposed architecture uses Model Predictive Control based on a kinematic bicycle model for planning safe reference trajectories. Its feasibility is ensured by adding a dynamic constraint on the steering angle which has been derived in this work in order to ensure the validity of the kinematic bicycle model. Several high-frequency controllers are then compared and their assets and drawbacks are highlighted. Finally, some preliminary work on model-free controllers and their application to automotive control are presented. In particular, an efficient tuning method is proposed and implemented successfully on the experimental vehicle of ENSIAME in collaboration with the laboratory LAMIH of Valenciennes.

Keywords

Motion planning, Autonomous driving, Model predictive control, Model-free control, Longitudinal and lateral control, vehicle dynamics