



# **Optimization-Based Motion Planning and Model Predictive Control for Autonomous Driving**

With Experimental Evaluation on a Heavy-Duty Construction Truck

PEDRO F. LIMA

Doctoral Thesis  
Stockholm, Sweden 2018

TRITA-EECS-AVL-2018:52  
ISBN 978-91-7729-888-5

KTH Royal Institute of Technology  
School of Electrical Engineering and Computer Science  
Department of Automatic Control  
SE-100 44 Stockholm  
SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framläggges till offentlig granskning för avläggande av doktorsexamen i elektro- och systemteknik måndagen den 17 september 2018 klockan 10.00 i F3, Kungliga Tekniska högskolan, Lindstedtsvägen 26, Stockholm, Sweden.

© Pedro F. Lima, September 2018. All rights reserved.

Tryck: Universitetsservice US AB

## Abstract

This thesis addresses motion planning and control of autonomous heavy-duty industrial vehicles, such as trucks and buses, using optimization-based techniques. Autonomous driving is a rapidly expanding technology that promises to play an important role in the near future, since it aims at more energy efficient, more convenient, and safer transport systems.

Autonomous vehicles are expected to have their first big impact in closed environments, such as mining areas and construction sites. In these sites, autonomous vehicle enabling infrastructure expansion and law adaption are significantly simpler than in an urban environment. Also, most of the challenging problems that must be addressed in urban environments are absent.

First, we propose a clothoid-based path sparsification algorithm to describe a reference path. This approach relies on a sparseness regularization technique such that a minimal number of clothoids is used to describe the reference path. The number of clothoids used to describe the path depends on the maximum deviation allowed between the reference and the clothoid-based path.

Second, we introduce a novel framework, in which path planning problems are posed in a convex optimization format, even when considering the vehicle dimension constraints, which maximizes the path planning performance in very constrained environments. In the case of off-road applications, solving the problem successively improves the planned trajectories. In the case of on-road applications, the planning can be seen as an optimal refinement of the road centerline, and, in the case of vehicles with prominent overhangs, the algorithm minimizes the amount of vehicle overhang that exits the driving lane.

Third, we present a progress maximization (i.e., traveling time minimization) model predictive controller for autonomous vehicles. The prediction model combines simple kinematics with vehicle dynamics constraints. The proposed controller optimizes the vehicle lateral and longitudinal motion simultaneously and its effectiveness is demonstrated, in simulation, including in the presence of obstacles.

Fourth, we design a smooth and accurate model predictive controller tailored for industrial vehicles, where the main goal is to reduce the vehicle “wear and tear” during its operation. Driving smoothness is encouraged by minimizing the first- and second-order spatial derivatives of the vehicle curvature. The controller effectiveness is shown both in simulation and experimentally in a Scania construction truck. We showed that the proposed controller has an extremely promising performance in real experiments with an average deviation from the path of 6 cm and never exceeding 30 cm in challenging tracks, at both high and low speeds.

Fifth, we propose a novel terminal cost and a terminal state set in order to guarantee closed-loop stability when designing and implementing a linear time-varying model predictive controller for autonomous path following. The terminal state set is the maximum positive invariant set over all the linear time-invariant models in a multi-plant description. The terminal cost is the upper bound on the infinite cost-to-go incurred by applying a linear-quadratic regulator control law to any of the possible models in the multi-model representation. The controller successfully stabilizes an autonomous Scania construction truck even when other model predictive controllers, with none or milder terminal cost and terminal state set constraints, cannot do it.

## Sammanfattning

Denna avhandling studerar banplanering och reglering av autonoma tunga industriella fordon, som lastbilar och bussar, genom att använda optimeringsbaserade metoder. Autonom körsättning är en snabbt växande teknologi som förutspås spela en stor roll i framtidens samhälle. Målen är att utveckla transportsystem med högre energieffektivitet, bekvämlighet och säkerhet.

Autonoma fordon förutspås ha sina första större industriella tillämpningar inom avgränsade miljöer, som gruv- och byggnadsområden. Det är mycket enklare att anpassa de autonoma fordonen till rådande infrastruktur och lagar inom sådana områden jämfört med stadsmiljöer. Dessutom saknas här de mest utmanande problem med säkerhet som måste lösas i stadsmiljöer.

Först så föreslår vi en klotoid-baserad algoritm för att beskriva referensbanan på en glesare form. Detta angreppssätt förlitar sig på en förglesningsteknik så att ett minimalt antal klotoider används för att beskriva referensbanan. Antalet klotoider som används beror på den maximalt tillåtna avvikelsen mellan referensen och den klotoid-baserade banan.

Sedan introducerar vi ett nytt ramverk där problemet med banplanering formuleras på konvex form som, även med begränsningar på fordonets dimensioner, maximerar banplaneringens prestanda i väldigt begränsade miljöer. Vid offroad-körsättning förbättras den planerade trajektorian genom att successivt lösa problemet. Vid körsättning på väg kan planeringen ses som en optimal förfiningsprocess av vägens mittlinje och i fallet med stora överhäng minimerar algoritmen mängden överhäng som hamnar utanför körbanan.

Vi presenterar sedan en framfarts-maximerande (i.e., körtids-minimerande) modell-prediktiv regulator för autonoma fordon. Prediktionsmodellen kombinerar enkel kinematik med begränsningar på fordonsdynamiken. Den föreslagna regulatorn optimerar fordonets laterala och longitudinella rörelse samtidigt och dess effektivitet demonstreras i simuleringar där även hinder förekommer.

Vi konstruerar en jämn och korrekt modell-prediktiv regulator skräddarsydd för industriella fordon där huvudmålet är att begränsa slitage under driftens. Jämn körsättning uppmuntras genom att minimera första och andra rumsderivatan av kurvaturen. Regulatornens effektivitet visas både i simulering och experimentellt i ett av Scanias anläggningsfordon. Den föreslagna regulatorn visar extremt lovande prestanda i verkliga experiment med avvikelse från banan som i genomsnitt var 6 cm och aldrig överskred 30 cm på en utmanande bana i både hög och låg fart.

Vi föreslår en ny slutkostnad och sluttillstånds-mängd för att garantera stabilitet vid återkoppling när vi implementerar en linjär tidsvarierande modell-prediktiv regulator för att autonomt följa en bana. Sluttillstånds-mängden är den maximala positivt invarianta mängden av alla linjära tids-invarianta modeller i en multi-system beskrivning. Slutkostnaden är den övre begränsningen på det oändliga cost-to-go-värdet som uppkommit genom att tillämpa en styrlag från en linjärkvadratisk regulator till någon av de möjliga modellerna i flermodells-representationen. Regulatorn stabiliseras framgångsrikt ett av Scanias autonoma anläggningsfordon även när andra modell-prediktiva regulatorer, med inga eller mildare slutkostnad och sluttillstånds-mängd, inte klarar det.

*To my beloved parents and girlfriend*



# Acknowledgements

There are many who have contributed to the work presented in this thesis. First, I would like to thank my main advisor Bo Wahlberg, for giving me the chance to work at the Automatic Control department and for his guidance and unending knowledge. A heartfelt thanks goes to Jonas Mårtensson for his advice, support, good mood, enthusiasm, and dedication.

I would like to acknowledge all the people that I have collaborated with during the past five years. In particular, I would like to thank Marco, Mattias, Mogens, Gonçalo, and Rui. I am grateful to Vinnova (FFI) and to Scania CV AB for financing the iQMatic project and this research.

A special thanks goes to Marco for helping me finding my way in the beginning of this journey and for all the priceless advice at every stage of my research. I would like to show my gratitude to Mattias and Soheil for all the hours we spent together making the experimental part of this thesis a success. The work presented in this thesis would not have been possible without you.

I would like to acknowledge some special people that have always been with me in the past few years. Thanks to André for knowing everything. Thanks to Antonio for always making me see the bright side when I complain. Thanks to Demia for sharing my love for cats. Thanks to Gonçalo and Rui for the relaxing fikas and interesting discussions. Thanks to Manne challenging me to run more often. Thanks to José for making me think outside of the box. Thanks to Miguel for the writing correctness obsession. Thanks to Pedro for always being available and patient. Thanks to Riccardo for introducing me to TikZ idiosyncrasies. Thanks to Sadegh for being a great office neighbor. Thanks to Sebastian for the interesting discussions. Thanks to Mikael and Valerio for your advice and endless knowledge about model predictive control. Thanks to Yuchao for helping me with the theoretical proofs. I want to thank to the Portuguese crew living in Stockholm for making life easier and funnier. Your friendship and support was fundamental for the completion of this work.

I am also thankful to all the people at Scania CV AB for believing in me and for all the great work. In particular, I would like to thank Lars Hjorth and Assad Alam for giving me the chance to work at Scania. Thanks to Henrik, Magnus, Marcello, and Linus for making my times at Scania well spent and amusing.

I would like to thank Burak, Gonçalo, Henrik, José, Marcello, Miguel, Riccardo, Rui, Sebastian, Valerio, and Yuchao for proofreading parts of this thesis, and Manne for helping me with the Swedish abstract.

Last but certainly not least, there are no words to express my gratitude to my family and my girlfriend Madalena. Your love, care, dedication and patience were key to overcome the difficulties of this journey.

*Pedro Lima*  
Stockholm, September 2018.

# Contents

<b>Contents</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Applications . . . . .	4
1.3 Autonomous Vehicles: Historical Overview and Future Challenges . . . . .	7
1.4 Thesis Outline and Contributions . . . . .	13
<b>2 Background</b>	<b>19</b>
2.1 Autonomous Vehicle Modules . . . . .	19
2.2 Optimization Fundamentals . . . . .	34
2.3 Optimal Control . . . . .	38
2.4 Summary . . . . .	41
<b>3 Vehicle Modeling</b>	<b>43</b>
3.1 Dynamic Vehicle Model . . . . .	43
3.2 Kinematic Vehicle Model . . . . .	48
3.3 Model Comparison and Discussion . . . . .	51
3.4 Summary . . . . .	53
<b>4 Clothoid-Based Path Sparsification</b>	<b>55</b>
4.1 Clothoids . . . . .	55
4.2 Problem Formulation . . . . .	57
4.3 Simulation Results . . . . .	62
4.4 Summary . . . . .	67
<b>5 Optimization-Based Path Planning</b>	<b>69</b>
5.1 Problem Formulation . . . . .	70
5.2 Simulation Results . . . . .	78
5.3 Summary . . . . .	87
<b>6 Progress Maximization Model Predictive Controller</b>	<b>89</b>
6.1 Problem Formulation . . . . .	89

6.2	Simulation Results . . . . .	97
6.3	Summary . . . . .	106
<b>7</b>	<b>Smooth and Accurate Model Predictive Controller</b>	<b>107</b>
7.1	Problem Formulation . . . . .	108
7.2	Experimental and Simulation Results . . . . .	111
7.3	Summary . . . . .	123
<b>8</b>	<b>Stability of Linear Time-Varying Model Predictive Controller</b>	<b>125</b>
8.1	Problem Formulation . . . . .	126
8.2	Model Predictive Control Design for Autonomous Driving . . . . .	128
8.3	Closed-loop Stability and Feasibility . . . . .	138
8.4	Experimental and Simulation Results . . . . .	140
8.5	Summary . . . . .	144
<b>9</b>	<b>Conclusions and Future Work</b>	<b>147</b>
9.1	Conclusions . . . . .	147
9.2	Future Work . . . . .	150
	<b>Acronyms</b>	<b>153</b>
	<b>Bibliography</b>	<b>155</b>

# Chapter 1

## Introduction

Autonomous vehicles have emerged to fulfill the need of more energy efficient, more convenient, and safer transport systems. In the last decade, many major vehicle manufacturers, suppliers, and technology companies have started projects and collaborations with the objective of developing fully self-driving vehicles. Autonomous driving is a rapidly expanding technology that promises to play an important role in the near future.

In this thesis, we use optimization-based techniques for path planning and smooth control of autonomous heavy-duty industrial vehicles, such as trucks and buses (see Figure 1.1). These vehicles are characterized by their large dimensions, mass, and slow dynamics. In industrial sites (e.g., mining areas), self-driving vehicles are supposed to run uninterruptedly during long periods of time. Moreover, maneuvering spaces can be quite narrow and the vehicle operation can effectively change the landscape of the area. Consequently, the motion planner must account for the vehicle dimensions and limitations, while the motion controller must target smooth driving and accurate path tracking. Aggressive steering results in a higher “wear and tear”, shortening the lifetime of the vehicles and, as a consequence, increasing the costs for operators. Moreover, fast steering wheel angle changes can lead to aggressive and unpredictable driving, increased chance of vehicle rollover, and damage to tires and steering components. Increasing efficiency and productivity is of utmost importance in these production sites. To that end, we address the challenges of controlling the vehicle close to its handling limits and in highly nonlinear tire operating regimes.

The rest of this chapter is organized as follows. In Section 1.1, we motivate the need for autonomous vehicles. In Section 1.2, we describe three main areas where autonomous vehicle are increasingly present. In Section 1.3, we briefly present the history of autonomous vehicles and discuss the associated challenges with their development. Finally, in Section 1.4, we provide the outline and contributions of the thesis.



Figure 1.1: Scania bus and trucks used as experimental and research platforms (courtesy of Scania CV AB).

## 1.1 Motivation

The ability of moving people and goods efficiently, conveniently, reliably, and fast is fundamental to meet the increasing demand for economic growth. Research in the field of intelligent transportation systems has been extremely active in the past decade. Both academia and industry make use of the most recent technological advances, namely in the fields of automation, connectivity, and electrification, to address the demand for a more sustainable, energy-efficient, and safer transportation system (see Figure 1.2).

### 1.1.1 Economic Motivation

The world population will exceed more than 10 billion people in the next 50 years, meaning that the need of goods will grow at least at the same rate as the population growth (United Nations Population Division, 2017). Traffic congestions cost about 1% of the European Union (EU) gross domestic product (GDP) every year (European Commission, 2011), greatly affecting freight transport (Torrey and Ford, 2017).

In the EU, about three quarters of all surface freight is transported by road (International Transport Forum, 2017b) accounting for 5% of its GDP (European Commission, 2011). The number of personal cars per capita is reaching record high numbers: for example, according to Eurostat (2017) there is one car for every two persons in the EU. This causes many challenges to the road transport sector that need to be addressed in order to satisfy the growing transport demand.

The demand for mineral resources will continue to grow, following the economic and population growth trends, specially in developing countries (Dubiński, 2013). Also, mineral deposits are becoming more difficult to mine over time, implying more capital and labor to extract them. Deposits are increasingly deeper and are less pure, incurring in an increased cost for extracting and processing the



(a) Three truck platoon demonstration in Södertälje, Sweden.



(b) An electrical bus used for public transportation in Södertälje, Sweden.



(c) Snow clearing with two trucks in Trysil, Norway.



(d) Container loading on a truck at a harbour in Stockholm, Sweden.



(e) A truck in an underground mine in Finland.



(f) Sugarcane harvesting in São Paulo, Brazil.

Figure 1.2: Examples of applications, in which vehicle automation is beneficial (courtesy of Scania CV AB).

material into a saleable product. Although modern mines are technologically advanced, mining companies are often located in remote and unattractive locations, making it hard to recruit skilled workers. To maintain their competitiveness, mining companies have long started to automate mining sites. Automated systems can be monitored from outside the mine, minimizing the number of human workers subject to hazardous mining environments, which allow to reduce or entirely eliminate health and safety risks associated with mining operations.

### 1.1.2 Safety Motivation

Global statistics show that cars are, by far, the most dangerous mode of transportation (Ernst and Young, 2015). Road traffic injuries, 94% of which are caused by human errors (European Commission, 2011), are predicted to become the third most common cause of disability by 2020. Although they have been reduced by 50% in the last decade (European Commission, 2017), they are the main cause of death among those aged between 15 and 29 years (World Health Organization, 2015). Moreover, heavy-duty vehicles are more likely to be involved in a fatal accident than passenger cars (Securing America's Future Energy, 2017), and they account for almost one in five road traffic fatalities (European Road Safety Observatory, 2016). Finally, regarding the mining industry, the challenges associated with accessing mineral deposits further raise health and safety issues.

### 1.1.3 Environmental Motivation

The environmental impact of the road transportation sector needs to be reduced in order to respect the goals defined in the Paris agreement (United Nations, 2015). For example, in 2015, 21% of the total human-related CO<sub>2</sub> emissions were due to road transportation alone (European Commission, 2017). The percentage of new cars that are not dependent of fossil fuels are still staggering low, since only one out of approximately 700 cars in Europe is electric (European Environment Agency, 2016). In the case of heavy-duty vehicles, these numbers are even less promising (International Transport Forum, 2017b). Industrial automated systems (e.g., autonomous mining trucks) are more precise and efficient, increasing the equipment uptime. This results in a more efficient energy consumption and less waste, leading to a more environmental friendly mining operation (Fisher and Schnittger, 2012).

## 1.2 Applications

The road to autonomous driving has been slowly paved since the 40s, with the gradual introduction of Advanced Driving Assistant Systems (ADAS) (e.g., anti-lock braking system, active front steering, electronic stability control, adaptive cruise control, lane departure warning system, and automatic parking). These systems play a major role in supporting the driver both in critical and tedious situations, reducing the number of traffic accidents and fatalities (Ross, 2014). The gradual

improvement of ADAS has made autonomous driving increasingly present in long-haulage, urban, and industrial applications.

### 1.2.1 Long-Haulage Applications

Platooning is one of the first viable applications of heavy-duty vehicle automation. Vehicle platooning consists in driving several vehicles close to each other in a convoy. This leads to decreased aerodynamical forces acting on the vehicles, which results in a more efficient fuel consumption and less greenhouse gas emissions (Alam et al., 2015a,b). Densely packing the vehicles results in an increased transportation capacity of the road network. Naturally, these effects are more noticeable when vehicles drive with inter-vehicle distances that are not considered safe for human drivers. Hence, it is crucial to have reliable vehicle-to-vehicle communication and accurate automatic control.

### 1.2.2 Urban Applications

In urban environments, autonomous vehicles will revolutionize city design and car ownership models. Today, personal cars are only productive 4% of the time, as 1% is spent in traffic and 95% parked (Forbes, 2014). Self-driving vehicles, together with car-sharing services, are expected to improve vehicle utilization rate up to 75% and to reduce the number of vehicles up to 90% (The Economist, 2015). This would lead to shorter and cheaper trips, less accidents and fatalities, and cleaner and more livable cities. Also, public transportation in urban cities is increasingly important to address traffic and mobility challenges. However, the survey performed by the European Commission (2013) shows that only 16% of European citizens use public transportation as their main mode of transport. The main reasons for the low usage of public transport are expensive tickets and irregular travel times. In the case of an autonomous bus fleet, the expenses related to the driver staff would vanish, and running times would probably be more predictable. Autonomous vehicles are also expected to achieve better fuel consumption, leading to reduced expenses. These factors would greatly reduce the final costs, making public transportation a more attractive option, reducing the number of cars.

### 1.2.3 Industrial Applications

Autonomous driving technologies can be extremely beneficial not only for on-road driving applications, but also for off-road applications, such as gravel pits, mining areas, construction sites, and loading terminals. Production sites are rapidly moving to remote locations and workers are experiencing more complicated working conditions, being subject to dangerous environments and tedious tasks (Bellamy and Pravica, 2011; Brown, 2012; Times, 2018). It is estimated that up to 69% of the accidents involving haul trucks in mines are caused by operator fatigue (Mining Magazine, 2013). Additionally, many companies have recently faced high



Figure 1.3: iQmatic project ambition (courtesy of Scania CV AB). In a mining site, autonomous trucks are able to accomplish complex tasks, such as the loading and unloading of debris. Remotely, in a command central, human operators are responsible for supervising all the actions and intervene if necessary.

administrative costs due to increased regulation over environmental compliance. Autonomous vehicle enabling infrastructure expansion and law adaption are significantly simpler than in an urban environment. Also, most of the challenging problems that must be addressed in urban environments are absent in special, enclosed areas. In mining environments, it is predicted that autonomous vehicles can increase productivity up to 20%, while decreasing fuel consumption up to 15%, maintenance costs by 8% (Cosbey et al., 2016), and a 40% tire life improvement (MIT Technology Review, 2016). For instance, studies show that, when compared with manual driven trucks, autonomous trucks travel 16% less distance and sit idle up to 3 hours less (Parreira, 2013). By eliminating the human-in-the-loop factor, the number of human fatalities is also expected to decrease (Bellamy and Pravica, 2011; Times, 2018). By reducing fuel use, greenhouse gas emissions, and operating costs, autonomous haul trucks contribute to a more environmentally friendly mining industry.

### iQmatic Project

The work presented in this thesis was mostly motivated by the goals of the project iQmatic, which was led by Scania CV AB, one of the most important truck manufacturers in Europe (16.5% market share). The project, which finished in the end of 2017, was funded by the Swedish government and automotive industry within the

strategic vehicle research and innovation (FFI) program. The project partners were Saab AB, Autoliv, Linköping University, and KTH Royal Institute of Technology. An illustration of the project ambition is shown in Figure 1.3. The idea was to develop a fully autonomous truck for mining applications by 2018. At the deploying site, a fleet of autonomous vehicles would navigate and cooperate to complete a set of missions given by a command center, where human operators would supervise the entire fleet of autonomous vehicles and intervene in emergency and deadlock situations.

### 1.3 Autonomous Vehicles: Historical Overview and Future Challenges

An autonomous vehicle can be defined as “a motor vehicle that uses artificial intelligence, sensors, and global positioning system coordinates to drive itself without the active intervention of a human operator” (Nevada State Law, 2011). The society of automotive engineers (SAE, 2016) has classified six levels of driving automation in the standard J3016:

0. **No automation:** The driver is in complete control of the primary controls (vehicle brake, steering, and throttle) at all times.
1. **Driver assistance:** The vehicle is capable of assisting the driver to regain control of the vehicle or stop faster than possible by acting alone. Examples are the electronic stability control, lane-keeping assist, and adaptive cruise controller.
2. **Partial assistance:** The vehicle is capable of automating two primary functions. For instance, using simultaneously lane-keeping assist and the adaptive cruise controller. The driver, which should be ready to intervene if necessary, is responsible for monitoring the surroundings, traffic, weather, and road conditions.
3. **Conditional assistance:** The vehicle is able to have full control of all safety-critical functions in specific traffic and environmental conditions. However, the driver is expected to be available for occasional control. The driver must be ready to intervene if the vehicle requests it.
4. **High automation:** The vehicle can be in complete control of the primary controls (vehicle brake, steering, and throttle). For example, in severe weather cases, the vehicle might decide to handover the control to the driver.
5. **Full automation:** Given a destination, the vehicle drives by itself and is in complete control of the primary controls (vehicle brake, steering, and throttle) at all times. The driver is not expected to intervene at all.

### 1.3.1 A Brief History

In 1939, the automated highway system proposed at “Futurama”, a General Motors exhibition, was, arguably, the first idea towards autonomous driving (Federal Highway Administration, 2007). After almost 80 years, it is clear that the automated highway system did not succeed, but the seed was planted and, in 1949, the first steps towards autonomous driving were being made with the introduction of the cruise controller (Ross, 2014). However, it would take almost 30 years more to create the first autonomous vehicle. In 1977, the vehicle created by the Tsukuba mechanical engineering lab in Japan, was able to drive autonomously at speeds up to 30 km/h on a dedicated test course (Tsugawa et al., 1979).

In the 80s, Bundeswehr University Munich and Daimler-Benz demonstrated autonomous driving at the European project PROMETHEUS (Dickmanns and Graefe, 1988). First, in 1987, the vehicle VaMoRs drove roughly 20 km at speeds up to 90 km/h (Dickmanns et al., 1994). In 1995, the vehicle VAMP drove autonomously in a three-lane French Autoroute with speeds up to 130 km/h, where the vehicle was able to change lane and perform lane and vehicle tracking. One year later, a Mercedes S-Class drove 95% fully autonomously from Munich, Germany to Odense, Denmark (Dickmanns, 1997; Maurer et al., 1996). In the same year, the vehicle built on the ARGO project, started by the Università di Parma within the PROMETHEUS project, drove autonomously more than 2000 km on the Italian highway network (Broggi, 1999; Broggi et al., 1999).

In the 90s, Carnegie Mellon University presented its NavLab vehicles that operated in structured environments (Thorpe et al., 1988). In 1995, NavLab 5, drove from Pittsburgh to Los Angeles on a 98.2% fully autonomous trip that is remembered today as “No Hands Across America” (Pomerleau and Jochem, 1996). In parallel, the idea of automated highway systems was being developed within the California Partners for Advanced Transportation Technology (PATH) program (Hedrick et al., 1994; Shladover et al., 1991; Varaiya, 1993). Started in 1986, the main goal of the PATH program was to show technical feasibility rather than product development and, in 1991, it was one of the first to demonstrate platooning at highway speeds using wireless communication (Chang et al., 1991). In 2000, five automated vehicles at Demo 2000 in Japan were capable of stop and go, platooning, merging, and obstacle avoidance (Kato et al., 2002).

More recently, in 2004, in the first DARPA Grand Challenge, none of the participants was able to finish the proposed route. However, one year later, in 2005, the second DARPA Grand Challenge (Buehler et al., 2007; Kogan and Murray, 2006) was won by Stanford vehicle Stanley (Thrun et al., 2006). It autonomously completed an unrehearsed 230 km long course through the Mojave Desert in less than 7 hours. In 2007, DARPA organized a new competition, the Urban Challenge (Buehler et al., 2009; Chen et al., 2008; Kammel et al., 2008; Montemerlo et al., 2008; Urmson et al., 2007). At the time, the event was considered a breakthrough moment, because the participants had to deal with other moving traffic and obstacles while respecting traffic rules.

In 2008, the Japanese “Energy-ITS” program achieved promising results, showing that a 14% fuel consumption reduction is possible in a three truck platoon driving at 80 km/h with a 10 meters inter-vehicular distance (Tsugawa et al., 2011). In 2010, studies under the SARTRE project concluded that this fuel reduction could potentially reach 20% and the road fatalities could be reduced by 10% while having a smoother and increased traffic flow (Robinson et al., 2010). Between 2013 and 2016, the EU project COMPANION aimed at developing on-board and off-board systems for creation, coordination, and operation of heavy-duty vehicle platoons (Eilers et al., 2015). Cooperative driving and vehicle platooning have also been demonstrated in the Grand Cooperative Driving Challenge (GCDC) 2011 (van Nunen et al., 2012) and in GCDC 2016 (Englund et al., 2016).

Since the early 2000s, Caterpillar Inc. and Komatsu Ltd. have been pioneers in automatic hauling systems in mining environments. Fleets of autonomous 240-ton trucks, which are remotely monitored, run uninterruptedly in mining sites in Chile, Canada, and Australia. Also, mining equipment suppliers, such as Sandvik and Atlas Copco, have developed automated loading and hauling systems for underground mining for mining sites in Chile, South Africa, and Canada.

In the past decade, in collaboration with academia, the majority of vehicle manufacturers, technology giants, and suppliers have allocated large amounts of resources to create new algorithms and techniques necessary to develop the first commercial fully autonomous vehicle.

In 2010, the VisLab Intercontinental Autonomous Challenge consisted in driving autonomously 13000 km between Parma, Italy and Shanghai, China (Broggi et al., 2012). The journey was completed by two vehicles in 100 days, crossing nine countries and, in some remote areas, no map was available. The first vehicle, which conducted experimental tests on sensing, decision, and control on selected sections of the trip, needed to be regularly intervened by a safety driver. The second vehicle was fully autonomous and followed the reference waypoints given by the first vehicle. Also in 2010, the Hyundai Autonomous Challenge focused in path following, obstacle detection, obstacle avoidance, and driving in tunnels (Cerri et al., 2011). Two years later, the challenge included signal detection, overtaking, cross-walk stops, and passenger detection (Jo et al., 2014, 2015). The AI and Intelligent Vehicle Future Challenge has been held in China between 2009 and 2017 (Li et al., 2016) with the support of the National Natural Science Foundation of China. From year to year, the difficulty of the challenge has evolved together with the technology maturation, and the participants have been required to drive autonomously in constrained roads, urban roads, freeways, and, recently, a combination of all. The main focus lies on the vehicle environment perception based only on the vehicle own sensors, to reduce the dependency on GPS localization.

In 2011, the Stadtpilot project (Nothdurft et al., 2011) achieved successful urban autonomous driving in German roads. In 2014, a Mercedes S-Class equipped only with standard sensors performed autonomously the Bertha-Benz historic route between Mannheim and Pforzheim in Germany (Ziegler et al., 2014b). The route consisted of different types of difficulties, where the vehicle needed to handle au-

tonomously narrow passages in urban areas, traffic lights, pedestrian crossings, intersections, and roundabouts in real traffic. Broggi et al. (2015) claim that the Public Road Urban Driverless (PROUD) car test was the first test in open public urban roads with a driverless vehicle that had to cope with roundabouts, junctions, pedestrian crossings, freeway junctions, traffic lights, and regular traffic. Finally, an autonomous racing Audi TTS has demonstrated its ability both in a normal race track (Funke et al., 2012; Kritayakirana and Gerdes, 2012b) and driving autonomously to the Pikes Peak, Colorado, USA (Wired, 2010).

Today, Waymo seems to lead the development of autonomous vehicles with reportedly more than 8 million kilometers driven autonomously, and many more in simulation. This data is used to feed its machine learning algorithms to better understand different real-world situations and what to do in each case. Tesla Inc. has introduced the “Autopilot” technology, which still requires the driver to be ready to intervene if requested. Volvo is currently leading the “Drive Me” project, which includes 100 self-driving vehicles being tested in the outskirts of Gothenburg, Sweden, using Volvo new technology called “Intellisafe Autopilot” (Volvo Cars AB, a,b). In fact, many major vehicle manufacturers have ongoing projects in the field of automated driving (e.g., Ford Motor Company; General Motors Company; Volkswagen AG).

Timeline 1.1 summarizes the main events in the history of autonomous vehicles.

### 1.3.2 Future Challenges

Despite the current trend, autonomous vehicles still face enormous challenges on different areas, such as technology maturity, infrastructure investment, consumer acceptance, legislation, and business model (KPMG, 2012).

#### Technology Maturity

Until today, ADAS have slowly been replacing the human driver, combining the information perceived by a set of sensors ranging from stereo cameras to long- and short-range RADARs in order to respond accordingly to the environmental dynamics. However, the available technology has not proven itself to be completely reliable, in the sense that the vehicle is still not able to accurately perceive its surroundings as humans do (McKinsey, 2017; Vox, 2016). For instance, while typical traffic situations can be repeated and learned using artificial intelligence techniques, there are always unforeseen scenarios that are difficult to handle in real-time with current approaches (The Guardian, 2018a,b).

A considerable shortcoming of the sensor-based approach is the high cost of integrating the available sensor technology in a commercial vehicle. Although the cost is currently decreasing, the entry price of an autonomous vehicle may be a barrier to the general public (Verge, 2018).

### **Infrastructure Investment**

Vehicles are increasingly connected in order to cooperate with each other and to easily gather more information. Vehicles can communicate in real-time with each other, through Vehicle-to-Vehicle (V2V) communication (Cadillac Motor Car Division, 2017), and with the infrastructure, through Vehicle-to-Infrastructure (V2I) communication (Audi AG, 2016). The EU project SAFESPOT has used V2V and V2I communication to create dynamic cooperative networks that share information between the vehicles and the road infrastructure in order to reduce traffic congestion and accidents (Vivo, 2007). The standard IEEE 802.11p defines the wireless Medium Access Control (MAC) and physical layer specifications to enable high-speed data exchange between vehicles and infrastructure (IEEE-Standards Association, 2010; Jiang and Delgrossi, 2008). Developments in 5G technologies might also support V2V and V2I communication in the future (European Telecommunications Standards Institute, 2012). Dedicated Short-Range Communication (DSRC) is currently the leading wireless medium for V2V communication (Society of Automotive Engineers, 2017). Decentralizing the vehicle perception would decrease the need of standalone solutions, complex sensing devices, and artificial intelligence technology while it would additionally increase the redundancy of the system. A drawback is the need of investment in both vehicles and infrastructures to be able to use these systems in practice.

### **Consumer Acceptance**

Public acceptance is one of the greatest barriers to autonomous vehicles. According to a study performed by Pew Research Center (2017), 87% of Americans would favor a requirement that all driverless vehicles have a human in the driver's seat who can take control of the vehicle in the event of an emergency, and 56% would not want to ride in a driverless vehicle. Many drivers may take an increased chance of having an accident in exchange for maintaining their ability to avoid it.

For taxi, bus, or truck drivers, the idea of having autonomous vehicles may not be an appealing technology (The Guardian, 2017). Automation replaces repetitive, unhealthy, and dangerous jobs that humans should not do. On the one hand, not just the people that make a living driving will lose their jobs but probably some maintenance jobs will disappear and traffic police will no longer be necessary. International Transport Forum (2017a) predicts that almost 70% of professional trucking jobs in the USA and Europe could be eliminated by autonomous technology. On the other hand, this will be compensated with the creation of intelligence- and information-based jobs (Wired, 2017). It is predicted that jobs in the development, observation, servicing, and maintenance of remotely controlled autonomous equipment, as well as in data processing and systems and process analysis will be created (Cosbey et al., 2016), as happened in the automation of the Port of Rotterdam (2016). It is important that current employees receive proper training and education to learn how to use new technologies and perform capably in their new roles.

## Legislation

When autonomous vehicles start to be available to the general public, there will be a transition period, where driverless vehicles and human drivers have to coexist. In case of accidents between autonomous and non-autonomous vehicles, there will be liability issues (Harvard Business Review, 2017). This will affect the way insurance companies will look at the transportation business, as well as how decision makers will influence the vehicle manufacturers and vehicle owners.

## Business Model

Finally, from the business point of view, autonomous vehicles are estimated to burst new opportunities (The Economist, 2015). For example, parking spaces will be much tighter (Banzhaf et al., 2017), vehicle sharing will substitute vehicle ownership as we know today, and autonomous fleets of mining trucks will depend on the efficiency of vehicle dispatcher algorithms.

### *Timeline 1.1: Autonomous vehicles historical highlights*

---

- 1939 - • Automated highway system proposed at “Futurama”
  - 1949 - • Introduction of the cruise controller
  - 1977 - • First autonomous vehicle (Tsukuba Mech. Eng. Lab)
  - 1986 - • Launch of the European project PROMETHEUS
  - 1986 - • Launch of the American project PATH
  - 1987 - • Autonomous vehicle VaMoRs (PROMETHEUS)
  - 1991 - • Highway platooning demonstration (PATH)
  - 1995 - • Autonomous vehicle VAMP (PROMETHEUS)
  - 1995 - • Autonomous vehicle NavLab 5 (CMU)
  - 1996 - • Launch of the ARGO project
  - 2000 - • Demo 2000, Japan
  - 2004 - • First DARPA Grand Challenge
  - 2005 - • Second DARPA Grand Challenge
  - 2007 - • DARPA Urban Challenge
  - 2008 - • “Energy-ITS” platooning program, Japan
  - 2009 - • Annual AI and Intelligent Vehicle Future Challenge, China
  - 2010 - • Launch of the European project SARTRE
  - 2010 - • VisLab Intercontinental Autonomous Challenge
  - 2011 - • Grand Cooperative Driving Challenge 2011
  - 2011 - • Stadtpilot project
  - 2013 - • COMPANION project
  - 2015 - • Public Road Urban Driverless (PROUD) project
  - 2016 - • Grand Cooperative Driving Challenge 2016
  - Today - • Waymo, Tesla Motors “Autopilot”, Volvo “IntelliSafe Autopilot”
-

## 1.4 Thesis Outline and Contributions

In this section, we outline the thesis organization and its contributions. The chapters titles marked with an asterisk contain experimental evaluation.

### 1.4.1 Chapter 2: Background

This chapter provides the background on autonomous driving from the system architecture point of view. First, we provide an overview of the literature related to the main modules of an autonomous vehicle. A route planner produces a high-level plan with long-term goals. The perception and localization modules are responsible for fusing sensor data to localize the vehicle and represent its surrounding environment in a consistent manner. The situation awareness module chooses the appropriate vehicle action depending on the scene understanding of the vehicle surroundings. This action is then used by a motion planner, which is responsible for planning feasible and obstacle-free paths or trajectories to be followed by a motion controller. In the end, we introduce the main mathematical tools used throughout the thesis.

### 1.4.2 Chapter 3: Modeling

This chapter introduces two different vehicle models. First, we introduce a dynamic vehicle model that consists of a 4-axle bicycle model describing the lateral dynamics of a Scania construction truck. The model parameters are identified and validated using real data. The dynamic vehicle model is used for simulation analysis and also contains a longitudinal model consisting of a cruise controller and vehicle longitudinal dynamics provided by Scania CV AB. Second, we explain the kinematic vehicle model that is used to predict the vehicle motion. We detail the concept of road-aligned frame (or Frenet-Serret frame), where the vehicle motion is described with respect to a given path. Moreover, time- and space-based modeling are distinguished.

### 1.4.3 Chapter 4: Clothoid-Based Path Sparsification

This chapter addresses the path sparsification problem (i.e., describing a path with few waypoints). We introduce an optimization algorithm that computes the minimum number of clothoids necessary to describe a reference path. This approach relies on a reweighed  $\ell_1$ -norm approximation of the  $\ell_0$ -norm. Each point of the clothoid-based path has embedded the clothoid segment properties, resulting in a computationally inexpensive path description. The number of clothoids used to describe the path depends on the parameter that defines the maximum deviation allowed between the reference and the clothoid-based path. The smaller the maximum allowed deviation, the larger the number of clothoid segments. The algorithm

performance is evaluated in simulation for different maximum allowed deviations. The method is evaluated in simulation using real recorded paths.

The chapter is primarily based on the publication:

- Lima, P. F., Trincavelli, M., Mårtensson, J., and Wahlberg, B. Clothoid-based model predictive control for autonomous driving. In *Proceedings of the IEEE European Control Conference*, pages 2983 – 2990, 2015b.

#### 1.4.4 Chapter 5: Optimization-Based Path Planning

This chapter introduces an optimization-based path planning framework. The development of a Linear or Quadratic Programming (LP or QP) algorithm aims the efficient incorporation of actuator, obstacle avoidance, and, foremost, vehicle dimension constraints. The main benefit of the proposed approach is the ability of posing the problem in a convex optimization format, even when considering vehicle dimensions, which improves the path planning performance in very constrained environments. The vehicle is modeled in the space-domain and in a road-aligned frame along the reference path to exclude time and speed from the equations of the vehicle dynamics. This approach also allows the formulation of driving corridor bounds as simple state bounds. In the case of off-road path planning, solving the problem successively, using the solution of the previous iteration as the reference path for linearization and discretization of the vehicle model and constraints, improves the planned trajectories. In the case of on-road planning, there is no need for successively solving the optimization problem, as the planning can be seen as a refinement of the road centerline. The proposed framework is extended for path planning of large vehicles, such as buses. These vehicles are typically forced to perform maneuvers, in which parts of the vehicle overhang go outside of the assigned driving lane. The main challenge is that long vehicles become extremely distorted in the road-aligned frame. Therefore, in the case of vehicles with prominent overhangs, the optimization objective aims to minimize the amount of vehicle overhang that exits the lane while ensuring that the paths are collision free with other obstacles.

The chapter is primarily based on the publications:

- Plessen, M. G., Lima, P. F., Mårtensson, J., Bemporad, A., and Wahlberg, B. Trajectory planning under vehicle dimension constraints using sequential linear programming. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 108–113, 2017.
- Lima, P. F., Oliveira, R., Mårtensson, J., and Wahlberg, B. Minimizing long vehicles overhang exceeding the drivable surface via convex path optimization. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 1374–1381, 2017c.

### 1.4.5 Chapter 6: Progress Maximization Model Predictive Control

This chapter addresses the problem of maximizing the progress (i.e., minimizing the traveling time) along a given path (e.g., the centerline of a road). In order to achieve real-time implementable solutions, we favor simple and computational efficient solutions to highly complex, detailed, and computationally demanding solutions. The problem is formulated as an MPC, where the vehicle is modeled in the space-domain and in a road-aligned frame, allowing the formulation of driving corridor constraints (e.g., lane boundaries) as state bounds. The vehicle model combines a kinematic model with vehicle dynamics constraints. These constraints represent the vehicle handling limits, modeled using a ‘g-g’ diagram, which limits the vehicle acceleration. Moreover, the tire slip is restricted ensuring that the vehicle tires always operate in their linear force region by limiting the lateral acceleration, which in turn limits the vehicle longitudinal velocity. The MPC is formulated as a convex optimization problem, by linearizing the vehicle model and constraints and approximating the ‘g-g’ diagram by an inscribed convex polytope. The proposed controller optimizes the vehicle lateral and longitudinal motion simultaneously and its effectiveness is demonstrated, in simulation, including in the presence of obstacles.

The chapter is primarily based on the publication:

- Lima, P. F., Collares Pereira, G., Mårtensson, J., and Wahlberg, B. Progress maximization model predictive controller. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, 2018a. To appear.

### 1.4.6 Chapter 7: Smooth and Accurate Model Predictive Control\*

This chapter presents a Smooth and Accurate MPC (SA-MPC) design for lateral control of an autonomous truck. The driving smoothness is encouraged by minimizing the first- and second-order spatial derivatives of the vehicle curvature such that the obtained curvature is encouraged to be piecewise affine. In clothoids, known by their comfort properties, the curvature varies linearly with the path arc-length. Hence, the SA-MPC mimics the smoothness and comfort characteristics of clothoids in the vehicle lateral control. The vehicle motion predictions are constrained to stay inside a driving corridor (e.g., lane boundaries). We analyze how the different parameters influence the performance of the controller using illustrative examples. The controller effectiveness is benchmarked against a standard MPC approach and a pure-pursuit controller, both in simulation and experimentally. The experimental tests are performed in a Scania construction truck at Scania test tracks facilities near Södertälje, Sweden. The main goal is to compare the controllers path tracking accuracy and their driving smoothness.

The chapter is primarily based on the publications:

- Lima, P. F., Trincavelli, M., Nilsson, M., Mårtensson, J., and Wahlberg, B. Experimental evaluation of economic model predictive control for an au-

tonomous truck. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 710–715, 2016c.

- Lima, P. F., Nilsson, M., Trincavelli, M., Mårtensson, J., and Wahlberg, B. Spatial model predictive control for smooth and accurate steering of an autonomous truck. *IEEE Transactions on Intelligent Vehicles*, 2(4):238–250, 2017b.

#### 1.4.7 Chapter 8: Stability Analysis of Linear Time-Varying Model Predictive Control\*

This chapter proposes a novel terminal cost and terminal state set used in the design of Linear Time-Varying MPC (LTV-MPC) for lateral control of an autonomous vehicle. The vehicle model, a nonlinear kinematic vehicle model, is successively linearized around the reference path, yielding an LTV model. The vehicle is modeled in the spatial domain and in a road-aligned coordinate frame along the reference path to exclude time and speed from the dynamics equations. The work makes use of a multi-plant description, in which the LTV model is described using several Linear Time-Invariant (LTI) models. We propose to use the maximum positive invariant set over all the LTI models in the multi-plant description as the terminal state set. The terminal cost is proposed to be the upper bound on the infinite cost-to-go incurred by applying a Linear-Quadratic Regulator (LQR) control law to any of the possible models in the multi-model representation. For our application, an upper bound can be obtained by positively scaling one of the Riccati matrices resulting from the infinite-horizon cost-to-go calculation, which considers that the vehicle model is contained inside a convex uncertainty polytope. By including the terminal cost and terminal state set in the controller formulation, closed-loop asymptotic stability of the LTV-MPC scheme is proved using Lyapunov arguments. The effectiveness of the proposed MPC design is evaluated in simulation and in real experiments with a Scania construction truck in a scenario that resembles an emergency maneuver, where the vehicle avoids a fictitious obstacle. The controller successfully stabilizes an autonomous Scania construction truck even when other controllers, with no or milder terminal cost and terminal state set, cannot stabilize the vehicle.

The chapter is primarily based on the publications:

- Lima, P. F., Mårtensson, J., and Wahlberg, B. Stability conditions for linear time-varying model predictive control in autonomous driving. In *Proceedings of IEEE Conference on Decision and Control*, pages 2775–2782, 2017a.
- Lima, P. F., Collares Pereira, G., Mårtensson, J., and Wahlberg, B. Experimental validation of model predictive control stability for autonomous driving. *Control Engineering Practice*, 2018b. To appear.

### 1.4.8 Chapter 9: Conclusions and Future Work

This chapter summarizes the work presented and emphasizes potential future research directions.

### 1.4.9 Other Publications by the Author

The following publications by the author had a significant influence on some of the contributions, but are not covered in the thesis.

- Collares Pereira, G., Svensson, L., Lima, P. F., and Mårtensson, J. Lateral model predictive control for over-actuated autonomous vehicle. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 310–316, 2017.
- Collares Pereira, G., Lima, P. F., Wahlberg, B., Pettersson, H., and Mårtensson, J. Linear time-varying robust model predictive control for discrete-time non-linear systems. In *Proceedings of the IEEE Conference on Decision and Control*, 2018. To appear.
- Lima, P. F., Trincavelli, M., Mårtensson, J., and Wahlberg, B. Clothoid-based speed profiler and control for autonomous driving. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 2194 – 2199, 2015a.
- Oliveira, R., Lima, P. F., Cirillo, M., Mårtensson, J., and Wahlberg, B. Curvature and sharpness continuous Dubins-like paths. In *Proceedings of the IEEE European Control Conference*, 2018b. To appear.

The order of the author names reflects the workload, where the first had the most important contribution. In all the above publications, the thesis author participated actively in the discussions and derivations of the theory and results, as well as in the paper writing.

### 1.4.10 Patents

A Swedish patent application has been filed in May 2017 having application number: 1750546-2. The application discloses an innovative method and a system for determining oversteering or understeering of a vehicle. The application was further filed in Brazil and in Germany.



## Chapter 2

# Background

This chapter establishes the required background of this thesis. First, in Section 2.1, we present the fundamental modules of an autonomous vehicle system, namely route planning, perception, localization, situation awareness, motion planning, and motion control. We briefly introduce related work about route planning, perception, localization, and situation awareness, while we describe in more detail related work about motion planning and control. Then, in Sections 2.2 and 2.3, we provide the technical background of this thesis. In Section 2.2, we introduce the main concepts and definitions of discrete optimization theory, such as feasibility, optimality, convexity, and active constraints, as well as linear and quadratic programming. Moreover, we review the main concepts regarding control theory, such as the notions of controllable, reachable, and invariant set. In Section 2.3, we introduce the concepts of optimal control used in this thesis, namely LQR and MPC. Finally, in Section 2.4 we summarize the chapter.

### 2.1 Autonomous Vehicle Modules

In this section, we describe the main modules in an autonomous vehicle system. An autonomous vehicle is composed by a number of systems, each responsible for a different task. An overview of a typical autonomous vehicle system architecture is depicted in Figure 2.1. We highlight six modules: route planning, perception, localization, situation awareness, motion planning, and motion control.

#### 2.1.1 Route Planning

A route planner produces a high-level plan with long-term goals (e.g., navigate between two locations using the shortest, the fastest, or the most fuel efficient path given the road network). The plan specifies which roads to take, preferred lanes to keep, and a recommended speed. This is similar to what commercially available GPS navigation devices already provide. Typically, the road network is

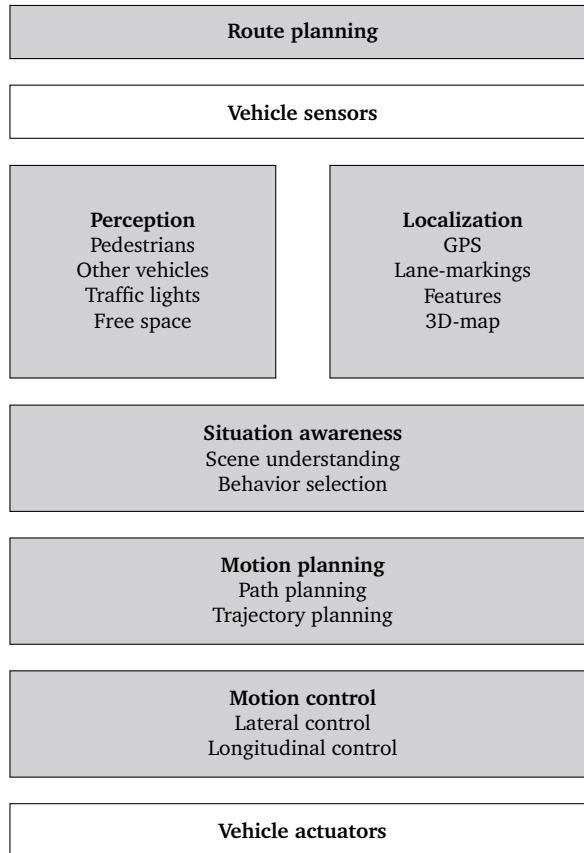


Figure 2.1: Overview of an autonomous vehicle system architecture. Inspired and adapted from Evestedt (2016).

represented as a directed graph, where the edge weights correspond to the cost of traversing a road segment. However, these graphs can contain millions of edges making classical shortest path algorithms, such as Dijkstra (1959) or A\* (Nilsson, 1969) too slow. Route planning becomes even more involved when taking into account other requirements, such as platoon coordination (Liang, 2016; van de Hoef, 2018). Efficient route planning in transportation networks is a significant research area and, for a thorough literature review of this topic, we refer to a survey by Bast et al. (2016).

### 2.1.2 Perception

The perception module is responsible for detecting, classifying, and tracking different objects present in the environment around the vehicle (see Figure 2.2). This



Figure 2.2: 360-map of the vehicle surroundings. The roof-mounted LIDAR sensor is used to generate a high-resolution map of the vehicle surroundings. Offline navigation maps contain static infrastructure information, such as traffic signs, crosswalks, and traffic lights, which enables software to quickly identify moving objects, such as pedestrians and cyclists (courtesy of Google).

module receives data generated from a set of sensors and fuses them into a consistent and suitable representation of the world for autonomous driving. In an autonomous vehicle, the typical sensor platform consists of several Radio Detection and Ranging (RADAR) sensors, Light Detection and Ranging (LIDAR) sensors, Global Positioning System (GPS) antennas, Inertial Measurement Units (IMUs), and monocular and stereo cameras. Having multiple sensors, it is possible to combine multiple streams of sensor data, allowing for a better understanding of the surrounding environment. Furthermore, by accurately modeling the uncertainty of the individual sensors, one can develop sensor fusion algorithms that combine their measurements and obtain an understanding of the environment that is far more accurate than what would be possible if the measurements were used separately.

The environment surrounding the vehicle is typically populated by lane markings, traffic signs, other vehicles, bicyclists, pedestrians, and other static or dynamic obstacles. Besides dealing with a highly dynamic environment, sensor noise, measurement errors, and visibility conditions hinder the perception problem. The computation of an occupancy grid map (Elfes, 1989) is a common approach to address the obstacle detection problem, where each grid cell stores the probability for the existence of an obstacle (Kammel et al., 2008; Montemerlo et al., 2008; Thrun et al., 2006). Besides deciding whether there is an obstacle or not, these cells may be used to characterize the environment in other ways. For example,

in determining the terrain type, the obstacle type, and helping self-localizing the vehicle (Wurm et al., 2010). When the goal is to obtain a 360-degree map of the environment surrounding the vehicle, the biggest challenges are loop closure (i.e., correctly asserting that the vehicle has returned to a previously visited location) and the ability to mark cells as unoccupied if the obstacle is not static (Levinson et al., 2007). Since occupancy maps are not suitable for keeping track of dynamic objects, Bayesian filters (Perrollaz et al., 2010), Kalman filters (Vatavu and Nedevschi, 2012), and particle filters (Catalin and Nedevschi, 2008) are commonly used to perform object tracking.

For in-depth discussions about perception for autonomous driving refer to recent surveys by Van Brummelen et al. (2018) and Bresson et al. (2017).

### 2.1.3 Localization

An accurate vehicle localization is crucial for a precise, safe, and comfortable vehicle motion. The localization module is responsible for accurately estimating the state (e.g., position, orientation, and speed) of the vehicle, locally with respect to its surroundings and globally with respect to a map.

Although GPS sensors are commonly used for vehicle localization, they are usually unreliable near tall buildings, or do not work at all underground. Therefore, more accurate state estimations are typically obtained including knowledge about the vehicle motion in the measurement data and fusing all available sensor information with offline maps (Broggi et al., 2015; Ward and Folkesson, 2016; Wolcott and Eustice, 2014; Ziegler et al., 2014b). For example, offline maps can contain the location of certain image features (Cummins and Newman, 2008), or some model of the world shape (Magnusson, 2009). Ziegler et al. (2014b) combine a feature-based localization and lane-marking-based localization. The first system detects landmarks around the ego position of the vehicle and manipulates them to match the static map. The latter is used mainly in paved road environments, where lane markings are common. The observed lane markings are also compared with the expected lane markings in the map (Schreiber et al., 2013).

Ward and Folkesson (2016) use RADAR sensors matching for vehicle localization. Thrun et al. (2006) use an unscented Kalman filter, which incorporates observations from GPS, compass, IMU, and wheel encoders. This has proven to be a successful approach, even when GPS failures occur.

### 2.1.4 Situation Awareness

The situation awareness module is responsible for determining the vehicle action (e.g., lane-keeping, overtaking, yielding, parking, off-road driving, standing still) and provide a scene understanding of the vehicle surroundings. It does so based on the processed sensor data from the perception module.

In recent years, scene understanding has been addressed using machine learning techniques, such as Convolutional Neural Networks (CNNs) (Long et al., 2015;



Figure 2.3: Scene understanding using CNNs (Cordts et al., 2016).

Schneider et al., 2016) (see Figure 2.3). Typically, an object is classified according to the way it moves (Darms et al., 2008) or the way it looks (Enzweiler and Gavrila, 2011; Lindner et al., 2004).

In the presence of other (non-autonomous) vehicles, the ability to predict the intentions of surrounding vehicles' drivers is crucial for the autonomous vehicle to behave in a proper way. In simple situations, other vehicles future trajectories can be predicted under the assumption of constant speed (Ferguson et al., 2008; Werling et al., 2008; Ziegler et al., 2014b). However, in more complicated scenarios, such as an intersection, determining the intention of other vehicles is more challenging. Some authors addressed this problem using probabilistic and machine learning methods (Liebner et al., 2013; Schlechtriemen et al., 2015; Ward and Folkesson, 2015). The prediction task becomes even more complex if one takes into account the influence that the autonomous vehicle decisions exert on other traffic participants (Cunningham et al., 2015; Eggert et al., 2015; Evestedt et al., 2016; Wei et al., 2013). For an in-depth discussion about driver intention prediction and estimation topics refer to a survey by Lefèvre et al. (2014).

The autonomous vehicle behavior is usually generated by hierarchical (concurrent) finite state machines (Kammel et al., 2008; Urmson et al., 2008; Ziegler et al., 2014b), which allows for a clear and comprehensible top-down modeling of reactive systems. Concurrency allows setting up multiple state charts that react to the same events in parallel. For example, a generic behavior (e.g., “drive”) can be separated in several sub-states (e.g., “drive start”, “drive stop”, and “lane change”). More recently, behavior trees have been the focus to model these behavior generation algorithms (Marzinotto et al., 2014). Behavior trees were originally used in video games (Lim et al., 2010) to control non-player characters. Behavior trees overcome the limitations of the state machines, because their implementation and formulation is usually simpler, more scalable, and modular. To address driver intention prediction under uncertainty, several works propose using Markov Decision

Processes (MDPs) and generalizations (Bandyopadhyay et al., 2013; Brechtel et al., 2011, 2014; Galceran et al., 2015; Ulbrich and Maurer, 2013).

### 2.1.5 Motion Planning

The motion planning module is responsible for computing a safe, comfortable, and dynamically feasible path or trajectory. While a path only includes kinematic considerations, a trajectory is also parametrized in time. A plan is computed from a vehicle pose (i.e., position and heading) to a goal pose while satisfying some constraints, such as driving inside assigned road lane boundaries, respecting the speed limits, and avoiding obstacles.

A path planner produces a detailed path plan to be executed within the next few seconds. Unlike the route planner paths, which respect constraints of the road network, the motion planner paths must take into account constraints that are more vehicle- and environment-specific, such as respect actuator limitations and avoid obstacles. Motion planning is usually handled by different algorithms depending on whether the environment is structured (e.g., urban roads or highways) or unstructured (e.g., mining sites).

Structured environments are usually associated with roads and parking lots, where there is a predefined traffic flow that indicates how the vehicle should drive and maneuver. In these environments, the planner computes a feasible path or trajectory such that, for example, the vehicle changes its lane, overtakes another vehicle, keeps a specific distance to the vehicle in front, or turns left at an intersection. In most scenarios, the motion planner has access to a reference path, usually the road centerline, and planning only involves maneuvers in the vicinity of that reference path.

Unstructured environments correspond to areas lacking clearly identified driving paths, and which require the motion planner to find ways to maneuver in it. Moreover, the planner needs to cope with environments whose structure can change over time and require computation of new paths in real-time, sometimes using exploratory algorithms to reach the destination while maneuvering in highly constrained environments and using the full kinematic capabilities of the vehicle.

## Motion Planning Methods

In general, motion planning for nonholonomic vehicles has been addressed through sampling-based, interpolation-based, and optimization-based methods (González et al., 2016; Katrakazas et al., 2015; LaValle, 2006; Paden et al., 2016).

### Sampling-Based Methods

Sampling-based methods are based on incrementally growing a tree rooted in the current configuration (e.g., position, heading, curvature, velocity) by sampling the free configuration space. The free configuration space corresponds to the set of

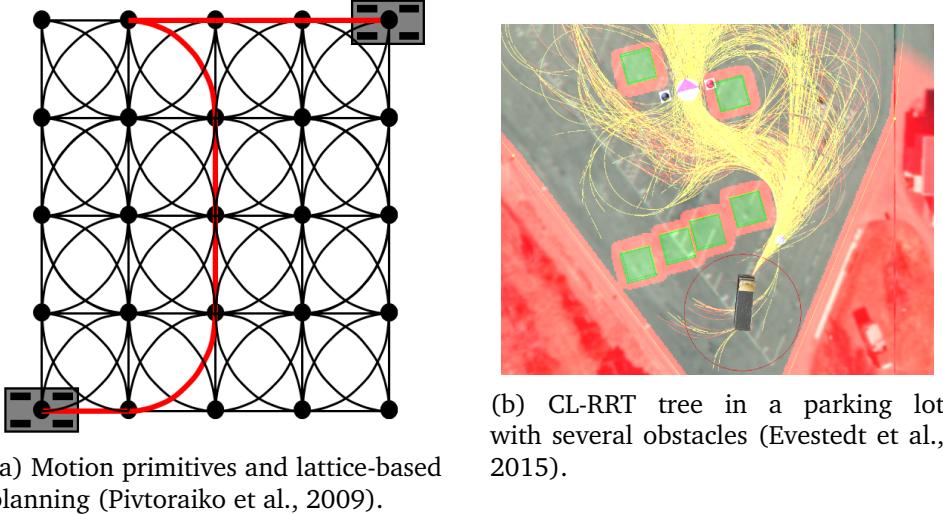


Figure 2.4: Examples of planning methods for autonomous vehicles.

vehicle configurations that do not result in collision with obstacles. The tree nodes only contain the reachable configurations from the initial configuration. The reachability of the free configuration space is explored using steering and collision checking routines. The steering routine returns a path segment starting from a given configuration going towards a goal configuration (but not necessarily reaching it) while ensuring the differential constraints are satisfied. The collision checking routine returns true if the path segment lies entirely in the free configuration space.

The major challenge is how to construct a discretization that approximates well the connectivity of the free configuration space. A common approach is to discretize the configuration space of the vehicle resorting to offline pre-computed motion primitives that allow the state space to be explored afterwards by classical graph-search algorithms (Daniel et al., 2010; Ferguson and Stentz, 2007; Hart et al., 1968). When the motion primitives generate a graph that covers the free configuration space uniformly they are called lattice-generating motion primitives (see Figure 2.4a). Lattice-based planners are common in practical applications, in particular autonomous driving, as they yield computationally fast algorithms (Pivtoraiko et al., 2009). The winners of the DARPA Urban Challenge use a lattice-based planner together with D\* for autonomous driving (Urmson et al., 2008). Montemerlo et al. (2008) propose a Hybrid A\* algorithm that works with continuous vehicle motion states, instead of using discrete nodes, to generate smooth paths in an unknown environment. Afterwards, a nonlinear optimization is used to improve the solution quality. Cirillo (2016) proposes a lattice-based planner for autonomous heavy-duty vehicles using a Time-Bounded A\* algorithm (Björnsson et al., 2009) when performing the graph search, which ensures real-time operation. That work

has been extended by Ljungqvist et al. (2017) for the task of reversing a general 2-trailer system, maintaining real-time performance.

The performance of lattice-based motion planners rely on state-space discretization, which means that these approaches may be unable to find a feasible path or produce paths that are oscillatory. Moreover, the precision with which the obtained paths approach the intended goal state depends directly on the granularity of the lattice. Therefore, smoothing methods have been applied to deal with these drawbacks (Oliveira et al., 2018a).

Another common approach is to use probabilistically complete algorithms (i.e., algorithms that find a solution, if one exists, with probability approaching one with increasing computation time). These algorithms are based on an exploratory behavior, which is achieved by, iteratively sampling a collision free point in a random location of the environment. The closest tree node is used as a starting point to simulate a path or trajectory that targets that sample. The trajectory is checked for collision and the simulated trajectory end point is added to the tree. With a given probability, the sampled point will correspond to the goal configuration. This will make the tree tend to grow towards the goal region. Eventually, a node reaches the goal configuration or a neighborhood region of it. The resulting path or trajectory is recovered by tracing backwards the tree links from the node in the goal region to the initial configuration.

Rapidly-Exploring Random Trees (RRT), which was originally proposed by LaValle (1998), is able to quickly find collision-free paths that are, by design, compliant with the vehicle nonholonomic constraints (see Figure 2.4b). However, RRTs tend to generate sub-optimal paths, which can lead to undesirable oscillatory behaviors (Karaman et al., 2011). RRT\* approaches address the sub-optimality problem, by continuously optimizing the path (Karaman and Frazzoli, 2011) and have been used for systems with nonholonomic constraints, such as road vehicles (hwan Jeon et al., 2011). For autonomous driving, Kuwata et al. (2009) proposes a Closed-Loop RRT (CL-RRT), which makes use of a low-level controller that, together with a system model, plans over the closed-loop dynamics. The need for fast replanning motivated the extension by Evestedt et al. (2015), where a sampling recovery technique using brake profile regeneration is proposed. In an obstacle avoidance scenario, when a given trajectory is deemed unsafe, re-simulating the closed-loop system using a deterministic brake profile (i.e., using the deceleration needed to stop the vehicle safely without colliding with the obstacle) increases the probability of generating safe trajectories.

### **Interpolation-Based Methods**

Interpolation-based methods use a given set of waypoints (e.g., generated with sampling-based methods) to generate a smooth path that respects the vehicle and environment constraints.

Dubins (1957) and Reeds and Shepp (1990) use only circular arcs and straight lines to construct the shortest path between any two given points for a car-like ve-

hicle. This type of paths do not have a continuous curvature function. Therefore, to follow them, a nonholonomic vehicle must stop and reorient its steering wheels. Otherwise, path following is not possible without major errors in position and in orientation at the curvature function discontinuities, since instantaneous changes in steering mechanisms are physically impossible. This problem triggered the emergence of smooth path planning (e.g., spline curves, clothoids, or polynomial curves) (Kanayama and Hartman, 1989).

Clothoids have the disadvantage of not having a closed-form expression, but the advantage of having the simplest curvature function that provides the smoothest transition curves making them easy to track (Brezak and Petrović, 2014; Funke and Gerdès, 2016; Funke et al., 2012; Walton and Meek, 2005). A spline curve is a piecewise polynomial curve that consists of other curves, such as polynomial curves or clothoids. Polynomial curves are commonly used to meet the interpolation constraints at each waypoint (i.e., position, heading, curvature). These type of curves has been used to plan not only lane change (Glaser et al., 2010) and overtaking maneuvers (Petrov and Nashashibi, 2014), but also combined with lattice-based planners (Xu et al., 2012) for on-road planning. Bernstein polynomials are a particular type of polynomials and the core of Bézier curves. These curves can be used to approximate clothoids (Sánchez-Reyes and Chacón, 2003; Wang et al., 2001) and for fast planning (Choi et al., 2008; Montés et al., 2007, 2008).

### Optimization-Based Methods

Optimization-based path planning has recently gained attention, due to the ever-increasing computing power and wide choice of numerical optimization tools. These approaches are attractive due to the possibility of explicitly capturing the vehicle model, in the form of kinematic, dynamic and actuator constraints, of taking into account vehicle geometry, and of easily defining optimization objectives. Furthermore, constrained optimization is often characterized by solution trajectories that are smoother than other approaches (Schwarting et al., 2018). The main challenge with this type of approaches is to guarantee real-time performance and convergence. Typically, that is not possible when formulating the problem in a nonlinear and non-convex framework (Frasch et al., 2013; Qian et al., 2016; Welling and Liccardo, 2012). Ziegler et al. (2014a) formulate the on-road trajectory planner as a continuous nonlinear optimization problem with nonlinear inequality constraints, where the vehicle shape is decomposed into several circles of fixed radius. The main goal is to design comfortable trajectories that avoid static and dynamic obstacles. The proposed method is successfully evaluated in practice in a typical urban scenario.

A path planning problem can also be formulated as a convex optimization problem by linearizing the vehicle dynamics and constraints (Bevan et al., 2010; Carvalho et al., 2013). Bevan et al. (2010) take into account the vehicle dimensions explicitly in the path planning layer, enforcing the vehicle to be within the space-varying road boundaries. However, these boundaries are expressed in a global

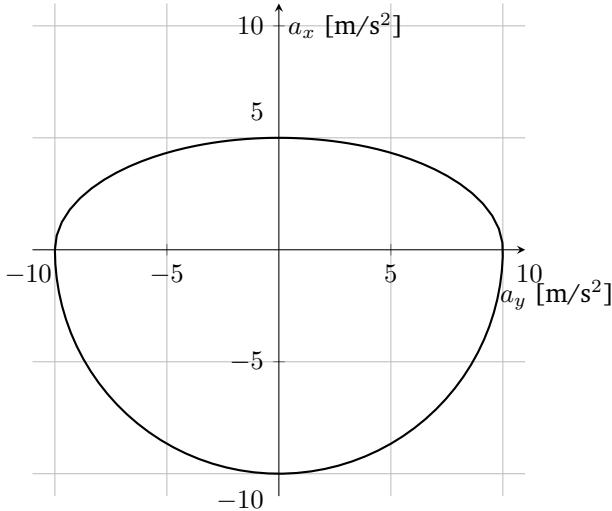


Figure 2.5: Example of a ‘g-g’ diagram.

frame, which makes the problem hard to extend to curved-road scenarios. Nilsson et al. (2015) address the problem of highway planning using MPC (i.e., solving an optimization-based path planning problem in receding-horizon), where obstacle avoidance constraints are affine, which allows the problem to be cast as a QP.

The generation of optimal trajectories such that the vehicle is close to its handling limits is addressed by Gerdts et al. (2009), where a time-based optimization seeks to, in a receding-horizon fashion, maximize the distance covered by the vehicle. A less computationally expensive approach is proposed by Liniger et al. (2015), where motion primitives consisting of arc circles driven with constant speed are projected onto the road centerline such that the one maximizing the progress is chosen and fed to the low level controller.

Velenis et al. (2007) propose a path planner that tracks a friction limit circle, typically modeled by a ‘g-g’ diagram (see Figure 2.5). This diagram is a simplified mathematical model, which neglects longitudinal and lateral load transfer and aerodynamic downforce effects, and represents the maximum force a tire can produce (Rice, 1973). Since acceleration and force are related through Newton’s second law, the vehicle handling limits are often characterized by its acceleration limits. The boundary of the circle represents the maximum allowed tire forces. Note that under forward acceleration, other limiting factors besides grip come into play (e.g., engine power in higher gears, aerodynamic drag). Therefore, ‘g-g’ diagrams are typically distorted on the upper part to account for those limitations. Tracking the limit circle leads the vehicle to use trail-braking (i.e., releasing the brake while steering towards the turn) and throttle-on-exit (i.e., pressing the throttle while steering outwards from the turn) techniques.

### 2.1.6 Motion Control

The motion control module is crucial in the design of an autonomous vehicle as it is responsible for stabilizing the vehicle and guiding it along a given reference path or trajectory. In the area of path following, the control design is concerned with steering the vehicle towards a path. Trajectory tracking is more restrictive than path following, since it is also concerned with following a time parametrized reference position. A more specific and significantly more complicated problem is vehicle stabilization around an equilibrium configuration (e.g., parking). For nonholonomic vehicles, it is well known that the vehicle model does not meet the Brockett's necessary condition for feedback stabilization (Brockett, 1983). Consequently, it is not possible to design a smooth (or even continuous) time-invariant state feedback controller that locally stabilizes a specified closed-loop equilibrium (i.e., to a specified vehicle pose). However, approaches including non-smooth feedback (Bloch et al., 1992; de Wit and Sørdalen, 1992) and time-varying feedback (Samson, 1991, 1995) have been proven to stabilize nonholonomic systems to an equilibrium.

The motion controller receives an obstacle-free and feasible reference path or trajectory from the motion planner. It also receives the vehicle states estimated by the perception and localization module. It uses this information to appropriately actuate on the steering wheel, throttle, and brakes of the vehicle. The problems of path following and trajectory tracking have been addressed in quite distinct ways and we review the literature on classical control and MPC methods (Guanetti et al., 2018; Paden et al., 2016).

#### Pure-Pursuit Control

One of the first proposed solutions to address the path following problem is pure-pursuit control (Amidi and Thorpe, 1991; Coulter, 1992; Wallace et al., 1985). The control law is based on geometric reasoning and is inspired by the way humans drive. Humans tend to focus on a spot some distance in front of the car and head towards that spot. In the pure-pursuit controller, the goal position is chosen on the reference path using an user-defined look-ahead distance. Then, as illustrated in Figure 2.6, geometric relations are used to determine the vehicle steering angle required to move the vehicle from its current position to the goal position. The steering angle is computed directly from the curvature of the arc-circle connecting both the vehicle current position and the goal position. The pure-pursuit controller is often applied due to its versatility and ease of implementation. For instance, two vehicles in the DARPA Grand Challenge (Buehler et al., 2007) and three vehicles in the DARPA Urban Challenge (Buehler et al., 2009) used pure-pursuit control. However, the controller needs to be fine-tuned for each vehicle, which can be hard to accomplish and is costly. Furthermore, pure-pursuit control, by design, can never obtain null deviation in curvy roads, resulting in a major drawback of the method.

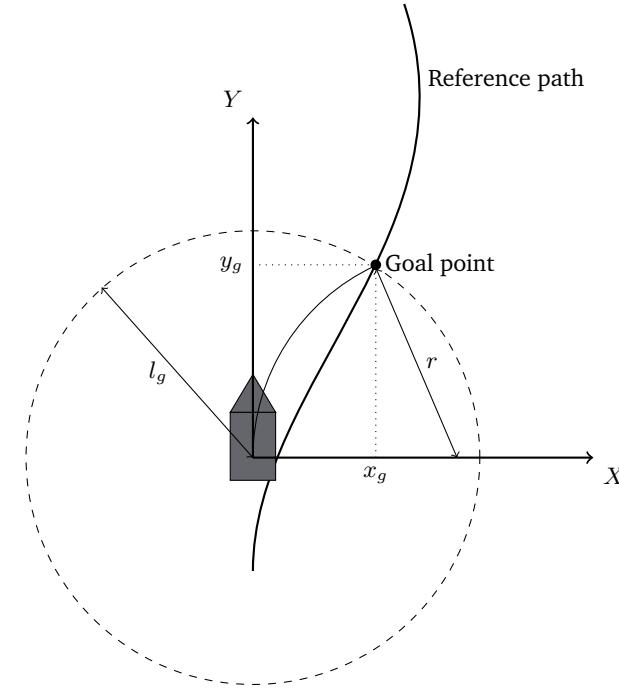


Figure 2.6: The pure-pursuit controller algorithm illustration. From the goal position  $(x_g, y_g)$  and the look-ahead distance  $l_g$ , it is possible to calculate  $r$ , which is the radius of the arc-circle necessary to steer the vehicle to the goal point. Adapted and inspired from Coulter (1992).

### Classical Control Design

Classical control approaches to path following and trajectory tracking problems are computationally inexpensive and the resulting closed-loop properties, such as stability, can be proved. However, typically, nonlinear time-varying models and constraints are hard to handle and the computed inputs are not optimal with respect to a desired cost function.

The first works on trajectory tracking and path following control design for non-holonomic ground vehicles have been presented by Samson et. al (see also de Wit et al. (1993) and the references therein). Trajectory tracking control is tackled using linear control design (De Luca et al., 1998), nonlinear control methods (Samson and Ait-Abderrahim, 1991), or dynamic feedback linearization (d'Andréa Novel et al., 1995; Oriolo et al., 2002). Path following control approaches usually consist of modeling the vehicle with respect to the reference path using the Frenet-Serret frame concept (Bishop, 1975), and then employing linear feedback design using the Taylor linearization of the corresponding model (Micaelli and Samson, 1993).

Other common automatic control approaches that are applied to path following are Lyapunov-based control (Kanayama et al., 1990) and back-stepping control design (Jiang and Nijmeijer, 1997).

### Model Predictive Control

MPC was originally developed in the 70s in order to improve the performance of petrochemical industries (Richalet et al., 1978). At the time, the low computation power available limited the use of MPC to processes with relatively slow dynamics. With the increase of computational power and optimization solvers efficiency, MPC has become a quite popular approach to address the problem of vehicle control, since it handles nonlinear time-varying models and constraints in a systematic manner (Bemporad, 2006; Borrelli et al., 2017; Garcia et al., 1989; Mayne et al., 2000). We provide an overview of the main areas where MPC has been applied for autonomous driving, namely active front steering, path following, vehicle handling, and stability control.

#### Active Front Steering

A particular area where MPC has been used is Active Front Steering (AFS). AFS modifies the vehicle steering angle without changing the steering wheel position, assisting the driver to drive around difficult cornering (e.g., on slippery roads) and improving vehicle stability. Di Cairano et al. (2013) combine differential braking and AFS to improve yaw stability using an MPC controller, which is designed to track a driver-requested yaw rate using offline optimal solutions computed using multiparametric programming (Bemporad et al., 2002). Another approach is proposed by Yoon et al. (2009), where state and input constraints are, instead, penalized in the objective function using a dynamic vehicle model. The use of complex models for predicting the vehicle behavior makes the convergence of the optimization solver slow. Also, adding constraints to the MPC, such as imposing comfort measures, increases the problem complexity and the computational burden. To overcome these problems, an LTV-MPC for AFS is proposed by Falcone et al. (2007a,b, 2008b), where the discrete-time linear system is based on successive online linearizations of the nonlinear vehicle model. Moreover, the stability conditions for LTV-MPC controllers are presented.

#### Path Following

Borrelli et al. (2005) presents a Nonlinear MPC (NMPC) that stabilizes a vehicle along a desired path while obeying to its physical constraints. Falcone et al. (2008c) use a similar strategy to combine AFS with braking such that a completely autonomous vehicle using a NMPC is capable of avoiding obstacles. Obstacle avoidance is also addressed by Rosolia et al. (2017), which propose a new optimization

algorithm, in which obstacles are considered by the trajectory generation formulated as NMPC. Since a nonlinear vehicle model is considered, the resulting MPC controller has a considerable computational complexity, which makes it difficult to be implemented in real-time. Kim et al. (2014) derive an approximate vehicle steering model and use the center of mass of a linear dynamic model (at constant speed) for path following.

In Graf Plessen et al. (2017), the authors formulate an LTV-MPC that uses a spatial road-aligned vehicle model (Gao et al., 2012) to track a piecewise affine reference trajectory. This reference is computed by a geometric corridor planner and is that one that provides the cumulated minimal absolute heading variation, with respect to a given path and safe avoidance of all obstacles. The distance of the obstacles to the road centerline is used to define a feasible driving corridor, which is then used as state constraints in the MPC. Using a similar approach, Turri et al. (2013) present an LTV-MPC to address lane-keeping and obstacle avoidance problems for a passenger car driving on low curvature and low friction road. Instead, Carvalho et al. (2014, 2015) demonstrate obstacle avoidance effectiveness using a stochastic MPC that uses an environment model to estimate and predict the state of surrounding vehicles.

Kong et al. (2015) compare the use of kinematic and dynamic vehicle models for MPC control design. The authors show that the kinematic model has better forecast errors when discretized at 200 ms, compared to 100 ms at speeds up to 15 m/s. Therefore, the authors motivate the MPC design based on a kinematic vehicle model, rather than a dynamic one, since it is still considerably accurate.

### Vehicle Handling

Recent advances in the estimation of vehicle sideslip and tire-road friction allow the development of envelope controllers to help the driver to overcome challenging cornering situations (Beal and Gerdts, 2013; Brown et al., 2017; Erlien et al., 2014, 2016). Envelope controllers, usually posed as MPCs, make use of a state space region called the handling envelope, in which the dynamics of the vehicle are stable. These controllers track the drivers intent while making sure that the vehicle does not enter in a unstable mode.

NMPC (Borrelli et al., 2005) and LTV-MPC (Falcone et al., 2008b; Katriniok and Abel, 2011) have also been successfully used to control a vehicle on the limits of handling. Katriniok and Abel (2011) show that successive linearizations of the non-linear model over the prediction horizon improve the accuracy of the LTV prediction model and consequently the performance of the controller. The proposed controller is compared to another LTV-MPC that uses linearizations that remain unchanged over the prediction horizon. Liniger et al. (2015) demonstrate the effectiveness of two different MPC formulations to control 1:43 scale model RC cars around a race track while avoiding obstacles. The dynamics of the scale model cars are probably not representative of the full-scaled vehicles dynamics, making it less applicable to the latter. Timings and Cole (2013) formulate an MPC for progress maximization

along a given path. Although the presented strategy shows promising results, it is not real-time implementable and the overall procedure is overcomplicated, leading to a demanding implementation to ensure the validity of the results.

On the low-level control side, the slip controller makes sure that the vehicle does not exceed its handling limits. Kritayakirana and Gerdès (2012a) use the vehicle center of percussion (i.e., the point where the effects of rotation and translation from the rear tire force cancel each other out) to design a simple linear steering controller. Moreover, Kritayakirana and Gerdès (2012b) as well as Funke et al. (2012) make use of a ‘g-g’ diagram to design a feedforward and lane-keeping-based feedback controller to track a predefined racing line on the handling limits of the vehicle. Another control approach is to distribute the force among the four tires of the vehicle using MPC and convex optimization for trajectory tracking (Park and Gerdès, 2015).

### Closed-Loop Stability

The closed-loop stability properties of MPCs have been extensively studied from a theoretical point of view. However, the practical analysis is not common in the literature, since many of proposed control designs cannot be implemented in real-time. When experimental evaluation is considered, the majority of the works either leave the stability concerns out (Thrun et al., 2006; Urmson et al., 2007), or the vehicle looks stable due to careful tuning of controller (Liniger et al., 2015; Turri et al., 2013) or due to the inclusion of vehicle dynamics constraints (Beal and Gerdès, 2013; Falcone et al., 2007a, 2008a; Funke et al., 2017; Katriniok et al., 2013). The scenarios presented in these works range from lane-keeping and obstacle avoidance (Funke et al., 2017; Turri et al., 2013) to racing applications (Beal and Gerdès, 2013; Liniger et al., 2015). Also, the experiments typically consider low-friction roads (Falcone et al., 2007a; Turri et al., 2013) or vehicle handling limits (Funke et al., 2017; Katriniok et al., 2013).

In autonomous vehicles, motion control stability is commonly ensured by the use of constraints in the MPC formulation that bound the tire slip angle. Consequently, the vehicle motion is bounded within the region of the state space that does not contain unstable vehicle dynamics. Falcone et al. (2008b) propose stability conditions for an LTV-MPC scheme used in AFS systems. An additional convex constraint bounding a quadratic function of the control effort and the predicted states is computed to ensure stability. However, this requires the MPC to be cast as a Sequential QP (SQP) that has typically higher computational burden than a QP. Additionally, considering the model as time-invariant by linearizing around the current set point and assuming that the terminal state set is a singleton, reduces the complexity of the overall design but also affects the feasibility region of the controller (i.e., region where there exists a sequence of control inputs, for which the constraints are obeyed).

### 2.1.7 The Shared Responsibility of Planning and Control

The separation between the motion planning and control modules is tenuous, at best, and the co-development of motion planning and control methods is the key in order to ensure semantically valid interactions between the two modules (Paden et al., 2016). Motion planning and control modules that fit to each other can decrease the computational burden of the overall system. However, an efficient motion planner needs to interact with a computationally intensive feedback controller, such as MPC, but a computationally light, simple, and less robust controller would require using a more detailed model at motion planning stage. Finally, if the motion planner produces a reference that does not meet the assumptions used to design the feedback controller, or vice-versa, then the performance of the system may be unsatisfactory.

In the literature, there have been some attempts to assign different levels of responsibility to each module. Gao et al. (2012) and Gu et al. (2013) propose that the motion planning is computationally more expensive and is performed less frequently, and the motion controller uses simplified vehicle models and is computationally more efficient. Contrariwise, Falcone et al. (2008a) propose a two-layer MPC approach, where a high-level MPC (i.e., the motion planning) computes a desired trajectory online using simplified models, which is then provided to a low-level MPC (i.e., the motion controller) that computes the vehicle inputs using a detailed nonlinear vehicle model. Götte et al. (2016) merge the planning and the control into a single approach called Nonlinear Model Predictive Planning and Control (NMPPC), where the hard constraints related to the road boundaries and obstacle avoidance are relaxed and substituted by convex barrier functions in the vicinity of those. However, in these, vehicle dimensions are typically discarded, either assuming point mass vehicle modeling or resorting to coarse approximations of the vehicle limits.

## 2.2 Optimization Fundamentals

In this section, we recall the main concepts and definitions of optimization theory. The concepts of feasibility, optimality, and convexity introduced in this section are widely used in the technical chapters of this thesis. The material presented in this section follows closely the presentation by Borrelli et al. (2017).

### 2.2.1 Optimization Problems

An optimization problem is generically formulated as

$$\min J(z) \quad (2.1a)$$

$$\text{s.t. } g_i(z) \leq 0 \quad \text{for } i = 1, \dots, q, \quad (2.1b)$$

$$h_j(z) = 0 \quad \text{for } j = 1, \dots, p, \quad (2.1c)$$

$$z \in \mathcal{Z} \subseteq \mathbb{R}^n, \quad (2.1d)$$

where the vector  $z$  is the decision variable and  $\mathcal{Z}$  is the optimization problem domain, which, in this case, is a subset of the  $n$ -dimensional vector space  $\mathbb{R}^n$ . The cost function  $J : \mathcal{Z} \rightarrow \mathbb{R}$  assigns to each decision  $z$  a cost  $J(z) \in \mathbb{R}$ . The functions  $g_i : \mathcal{Z} \rightarrow \mathbb{R}$ ,  $h_j : \mathcal{Z} \rightarrow \mathbb{R}$  are called inequality constraints and equality constraints, respectively. The domain of  $\mathcal{Z}$  is the intersection of the domains of the cost function and constraint functions. If a point  $\bar{z} \in \mathbb{R}^n$  satisfies all inequality and equality constraints it is a feasible point.

We are interested in finding an optimal solution of (2.1). In mathematical terms, this means that we seek  $z^*$  such that

$$\begin{aligned} J(z) &\geq J(z^*) = J^*, \quad \forall z \in \mathcal{Z}, \\ \text{with } z^* \in \mathcal{Z}, \quad g_i(z^*) &\leq 0, \quad i = 1, \dots, q, \quad \text{and } h_j(z^*) = 0, \quad j = 1, \dots, p. \end{aligned} \tag{2.2}$$

The number  $J^*$  is also called the optimal value of problem (2.1).

In this thesis, we deal with optimization problems of the same form as problem (2.1).

**Remark.** *The equality constraints are sometimes excluded from the optimization problem formulation by rewriting the cost function and inequality constraints. The elimination of equality constraints can make a solver less efficient due to the destruction of useful structural properties of the problem, such as sparsity.*

**Definition 2.1** (Convex set). *A set  $\mathcal{S} \in \mathbb{R}^n$  is convex if*

$$\lambda z_1 + (1 - \lambda)z_2 \in \mathcal{S}, \quad \forall z_1 \in \mathcal{S}, \quad \forall z_2 \in \mathcal{S}, \quad \text{and } \lambda \in [0, 1].$$

**Definition 2.2** (Convex function). *A function  $f : \mathcal{S} \rightarrow \mathbb{R}$  is convex if  $\mathcal{S}$  is convex and*

$$f(\lambda z_1 + (1 - \lambda)z_2) \leq \lambda f(z_1) + (1 - \lambda)f(z_2), \quad \forall z_1 \in \mathcal{S}, \quad \forall z_2 \in \mathcal{S}, \quad \text{and } \lambda \in [0, 1].$$

*A function  $f : \mathcal{S} \rightarrow \mathbb{R}$  is concave if  $\mathcal{S}$  is convex and  $-f$  is convex.*

The optimization problem (2.1) is said to be convex if the cost function  $J$  is convex on  $\mathcal{Z}$  and  $\mathcal{Z}$  is a convex set. Examples of convex functions are linear and quadratic functions. The optimal solution of a convex optimization problem, if it exists, is a global optimum (Borrelli et al., 2017, Theorem 1.1).

**Definition 2.3** (Polyhedron). *A polyhedron  $\mathcal{P}$  in  $\mathbb{R}^n$  is defined by the intersection of a finite set of closed half-spaces in  $\mathbb{R}^n$ :*

$$\mathcal{P} = \{z \in \mathbb{R}^n : Gz \leq g, \text{ with } G \in \mathbb{R}^{q \times n}, \quad g \in \mathbb{R}^q\},$$

where  $Gz \leq g$  represents a system of inequalities.

**Definition 2.4** (Polytope). *A polytope is a bounded polyhedron. A polytope is a set and convexity is defined according to Definition 2.1.*

### 2.2.2 Numerical Methods for Optimization

Usually, an analytical solution to problem (2.1) does not exist. To solve it, we typically resort to iterative algorithms (also called solvers) that, starting from an initial guess  $z_0$ , generate points  $z_k$  such that the sequence  $\{J(z_k)\}_{k=0}^{\infty}$  converges to the optimal solution  $z^*$  as  $k \rightarrow \infty$ .

In general, there are two categories of numerical methods for solving (2.1), namely, first-order methods that make use of the first-order information of the objective function (e.g., gradient and gradient projection methods) and second-order methods that, in addition, make use of second-order information (e.g., Newton's method). Among the most widely used numerical methods for convex optimization problems are interior point and active set methods. Interior point methods solve convex optimization problems that include inequality constraints by applying Newton's method to a sequence of equality constrained problems, using a so-called barrier function. Active set methods identify the set of active constraints at the optimal solution such that the problem can be solved efficiently.

### 2.2.3 Linear and Quadratic Programming

When the cost and the constraints of the optimization problem (2.1) are affine, then the problem is called a linear program (LP):

$$\min c^T z \quad (2.3a)$$

$$\text{s.t. } Gz \leq g, \quad (2.3b)$$

$$Wz = w, \quad (2.3c)$$

where  $G \in \mathbb{R}^{q \times n}$ ,  $g \in \mathbb{R}^q$ ,  $W \in \mathbb{G}^{p \times n}$ , and  $w \in \mathbb{R}^p$ .

When the cost is a convex quadratic function and the constraints are affine, the optimization problem (2.1) is called a quadratic program (QP):

$$\min \frac{1}{2} z^T H z + h^T z \quad (2.4a)$$

$$\text{s.t. } Gz \leq g, \quad (2.4b)$$

$$Wz = w, \quad (2.4c)$$

where  $G \in \mathbb{R}^{q \times n}$ ,  $g \in \mathbb{R}^q$ ,  $W \in \mathbb{R}^{p \times n}$ ,  $w \in \mathbb{R}^p$ ,  $\mathbb{R}^{n \times n} \ni H = H^T > 0$ , and  $h \in \mathbb{R}^n$ .

### 2.2.4 Controllability, Reachability, and Invariance

Consider a discrete-time system subject to external inputs:

$$z(t+1) = f(z(t), u(t)), \quad (2.5)$$

where  $z(t) \in \mathbb{R}^n$  and  $u(t) \in \mathbb{R}^m$  are the state and input vectors, respectively. Both signals are subject to state and input constraints

$$z(t) \in \mathcal{Z} \subseteq \mathbb{R}^n, \quad u(t) \in \mathcal{U} \subseteq \mathbb{R}^m, \quad \forall t \geq 0, \quad (2.6)$$

where the sets  $\mathcal{Z} \subseteq \mathbb{R}^n$  and  $\mathcal{U} \subseteq \mathbb{R}^m$  are polytopes. The function  $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  describes the system dynamics.

When system (2.5) is subject to the feedback control law  $u(t) = F(t)z(t)$ , where  $F(t)$  is a time-varying feedback control gain, the discrete-time autonomous system becomes

$$z(t+1) = f(z(t), F(t)). \quad (2.7)$$

**Definition 2.5** (One-step controllable set). *For system (2.5), the one-step controllable set to the set  $\mathcal{S}$  is denoted as*

$$\text{Pre}(\mathcal{S}) = \{z \in \mathbb{R}^n : \exists u \in \mathcal{U}, f(z, u) \in \mathcal{S}\}. \quad (2.8)$$

$\text{Pre}(\mathcal{S})$  is the set of states that evolve into the target set  $\mathcal{S}$  in one time step.

**Definition 2.6** ( $N$ -step controllable set). *For a given target set  $\mathcal{S} \subseteq \mathcal{Z}$ , the  $N$ -step controllable set  $\mathcal{K}_N(\mathcal{S})$  of system (2.5) subject to the constraints (2.6) is defined recursively as*

$$\mathcal{K}_j(\mathcal{S}) = \text{Pre}(\mathcal{K}_{j-1}(\mathcal{S})) \cap \mathcal{Z}, \quad \mathcal{K}_0(\mathcal{S}) = \mathcal{S}, \quad j = \{1, \dots, N\}. \quad (2.9)$$

All states of system (2.7) belonging to the  $N$ -step controllable set  $\mathcal{K}_N(\mathcal{S})$  evolve to the target set  $\mathcal{S}$  in  $N$  steps, while satisfying the constraints.

**Definition 2.7** (Positive invariant set). *A set  $\mathcal{O} \subseteq \mathcal{Z}$  is said to be a positive invariant set for system (2.7) subject to the constraints in (2.6), if*

$$z(0) \in \mathcal{O} \Rightarrow z(t) \in \mathcal{O}, \quad \forall t \geq 0.$$

For a given feedback controller  $u(t) = F(t)z(t)$ ,  $\mathcal{O}$  is the set of initial states, for which the future trajectory never violates the system constraints.

**Definition 2.8** (Maximal positive invariant set). *The set  $\mathcal{O}^\infty \subseteq \mathcal{Z}$  is the maximal positive invariant set of the autonomous system (2.7) subject to the constraints in (2.6) if  $\mathcal{O}^\infty$  is invariant and contains all the invariant sets contained in  $\mathcal{Z}$ .*

**Theorem 2.1** (Geometric condition for invariance). *A set  $\mathcal{O} \subseteq \mathcal{Z}$  is a positive invariant set for the autonomous system (2.7) subject to the constraints in (2.6) if and only if*

$$\mathcal{O} \subseteq \text{Pre}(\mathcal{O}). \quad (2.10)$$

*A proof can be found in Borrelli et al. (2017, Theorem 10.1).*

In other words, if the system (2.7) is inside  $\mathcal{O}$ , it remains there in the future while respecting the constraints (2.6).

## 2.3 Optimal Control

In this section, we introduce the basic definitions and concepts regarding optimal control that are used in the rest of this thesis. Particularly, we focus on LQR and MPC.

### 2.3.1 Linear-Quadratic Regulator

Consider the LTI system

$$z(t+1) = Az(t) + Bu(t), \quad \forall t \geq 0, \quad (2.11)$$

where  $z(t) \in \mathbb{R}^n$  and  $u(t) \in \mathbb{R}^m$  are the state and input vectors.

Let the LQR control problem be defined as

$$\min_{U_{t,\infty}} J_\infty(z(t), U_{t,\infty}) = \sum_{k=0}^{\infty} z_k^T Q z_k + u_k^T R u_k \quad (2.12a)$$

$$\text{s.t. } z_{k+1} = Az_k + Bu_k, \quad k = 0, \dots, \infty, \quad (2.12b)$$

$$z_0 = z(t), \quad (2.12c)$$

where  $z_k$  denotes the state vector at discrete-time  $k$  obtained by applying an input sequence  $\{u_j\}_{j=0}^{k-1}$  to the model

$$z_{k+1} = Az_k + Bu_k,$$

starting from the state  $z_0 = z(t)$ . We assume that the state penalty matrix is semi-positive definite  $Q = Q^T \geq 0$  and the input penalty matrix is positive definite  $R = R^T > 0$ . Let  $U_{t,\infty} = \{u_k\}_{k=0}^{\infty}$  denote a sequence of inputs.

We can use dynamic programming to solve (2.12) in a recursive manner. This approach yields a feedback control law expressing, at each time step, the control law as function of the state at that time. The derivation of the feedback control law is based on solving recursively the discrete-time Riccati equation

$$P_t = A^T P_{t+1} A + Q - A^T P_{t+1} B (B^T P_{t+1} B + R)^{-1} B^T P_{t+1} A, \quad (2.13)$$

backwards (i.e., for  $t \rightarrow -\infty$  starting with  $P_0 = Q$ ). Assuming that it converges to a solution  $P_\infty$  (Goodwin and Sin, 2014, Appendix E), which satisfies the Algebraic Riccati Equation (ARE)

$$P_\infty = A^T P_\infty A + Q - A^T P_\infty B (B^T P_\infty B + R)^{-1} B^T P_\infty A, \quad (2.14)$$

then, the optimal feedback control law is

$$u^*(t) = F_\infty z(t) = -(B^T P_\infty B + R)^{-1} B^T P_\infty A z(t), \quad (2.15)$$

and the optimal infinite-horizon cost-to-go is

$$J_\infty^*(z(t)) = z^T(t)P_\infty z(t). \quad (2.16)$$

The conditions that guarantee that the discrete-time Riccati equation (2.13) converges and yields a stabilizing feedback control law are summarized in the following theorem.

**Theorem 2.1** (LQR stability). *If  $(A, B)$  is a stabilizable pair and  $(A, Q^{1/2})$  is an observable pair, then the discrete-time Riccati equation (2.13) with  $P_0 \geq 0$  converges to the unique positive definite solution  $P_\infty$  of the ARE (2.14) and all the eigenvalues of  $(A + BF_\infty)$  lie strictly inside the unit circle.*

A proof can be found in (Lewis and Syrmos, 1995, Theorem 2.4-2).

The first condition is necessary for  $J_\infty^*(z(t))$  (and  $P_\infty$ ) to be finite. The second condition requires that the output<sup>1</sup>  $Q^{1/2}z$  captures all system modes. Hence, convergence of the output  $Q^{1/2}z$  implies convergence of the state to zero.

Thus, for linear systems the optimal control law can be computed efficiently, is guaranteed to give an asymptotically stable closed-loop, and is simple to evaluate and implement online. However, when the optimization problem (2.12) includes state and control constraints, the optimal control feedback law is difficult to find. Hence, rather than relying on dynamic programming and aiming for optimal policies, we solve the optimization problem (2.12) in a receding-horizon fashion to find control laws that guarantee stability and have near-optimal performance.

### 2.3.2 Model Predictive Control

Consider the infinite-horizon constrained optimal control problem to be solved at time instant  $t$

$$\min_{U_{t,\infty}} J_\infty(z(t), U_{t,\infty}) = \sum_{k=0}^{\infty} z_{k|t}^T Q z_{k|t} + u_{k|t}^T R u_{k|t} \quad (2.17a)$$

$$\text{s.t. } z_{k+1|t} = Az_{k|t} + Bu_{k|t}, \quad k = 0, \dots, \infty, \quad (2.17b)$$

$$z_{0|t} = z(t), \quad (2.17c)$$

$$z_{k|t} \in \mathcal{Z}, \quad k = 0, \dots, \infty, \quad (2.17d)$$

$$u_{k|t} \in \mathcal{U}, \quad k = 0, \dots, \infty, \quad (2.17e)$$

where  $\mathcal{Z}$  and  $\mathcal{U}$  are convex polytopes and represent state and input constraints, respectively. The notation  $z_{k|t}$  denotes the state vector at time  $k$  obtained by applying an input sequence  $\{u_{j|t}\}_{j=0}^{k-1}$  to the model

$$z_{k+1|t} = Az_{k|t} + Bu_{k|t},$$

---

<sup>1</sup>Let the objective function be rewritten as  $z^T Q z = (z^T Q^{1/2})(Q^{1/2}z)$ . Thus, the “output”  $Q^{1/2}z$  is penalized in the objective function.

starting from the state  $z_{0|t} = z(t)$ .

When the system is linear and the objective is quadratic, the optimal cost-to-go (2.16) is quadratic and can be computed efficiently offline. However, if we add input or state constraints, the cost-to-go is typically not quadratic and consequently, the optimal control problem (2.17) is solved using iterative optimization methods, such as those introduced in Section 2.2.2 and in a receding-horizon fashion. This is why the problem (2.17) is known as Receding-Horizon Control (RHC) problem or MPC problem. The basic idea behind RHC (or MPC) is to compute a sequence of optimal control inputs over a fixed prediction horizon  $N$ , but then apply only an input subset (typically, the first input) of the optimal sequence to the system. This procedure is repeated, at the next sampling instant, over a shifted horizon using the updated state of the system. Solving repeatedly the optimization problem using the updated states of the system provides feedback, making it robust to disturbances and model uncertainties. The biggest advantage of MPC over classical control approaches is its ability to cope with both input and state constraints, which represent limitations on actuators or limitations that arise from physical, economic, or safety constraints.

Problem (2.17) is a convex optimization problem with an infinite number of variables. In order to be able to solve this problem, we rewrite it into the equivalent finite-horizon problem

$$\min_{U_{t,\infty}} J_\infty(z(t), U_{t,N}) = z_N^T Q_f z_N + \sum_{k=0}^{N-1} z_{k|t}^T Q z_{k|t} + u_{k|t}^T R u_{k|t} \quad (2.18a)$$

$$\text{s.t. } z_{k+1|t} = A z_{k|t} + B u_{k|t}, \quad k = 0, \dots, N-1, \quad (2.18b)$$

$$z_{0|t} = z(t), \quad (2.18c)$$

$$z_{k|t} \in \mathcal{Z}, \quad k = 1, \dots, N-1, \quad (2.18d)$$

$$u_{k|t} \in \mathcal{U}, \quad k = 0, \dots, N-1, \quad (2.18e)$$

$$z_{N|t} \in \mathcal{Z}_f, \quad (2.18f)$$

where the terminal cost represents the cost-to-go of the unconstrained infinite-horizon control problem, the terminal state penalty matrix is semi-positive definite  $Q_f = Q_f^T \geq 0$ , and  $\mathcal{Z}_f$  is a convex polytope that represents the terminal state constraint.

The formulation (2.18) is typically referred to as linear MPC, due to the linear prediction model, polytopic constraints, and quadratic cost function. When the prediction model is LTV, i.e.,

$$z_{k+1|t} = A_{k|t} z_{k|t} + B_{k|t} u_{k|t},$$

we refer to it as LTV-MPC.

In MPC, the optimization problem can become infeasible (i.e., there does not exist a sequence of control inputs for which the constraints are satisfied). Even if

the optimization problem is always feasible, the computed optimal control inputs may not lead to an asymptotically stable closed-loop system.

Conditions on how the terminal weight  $Q_f$  and the terminal constraint set  $\mathcal{Z}_f$  should be chosen such that closed-loop stability and feasibility are ensured, have been extensively studied in the past three decades (Mayne et al., 2000). Although no closed form expression can be found to compute the constrained infinite-horizon cost-to-go function, we are often able to provide a good estimate of it in a restricted region of the state space, the so-called terminal region  $\mathcal{Z}_f$ . The main idea is to find a feedback controller law that stabilizes the unconstrained system inside an invariant terminal region  $\mathcal{Z}_f$ . Then, the closed-loop is stable if the cost function accounts for the infinite-horizon cost. The finite-horizon cost, when the controller is constrained, is added to the infinite-horizon cost, when the controller is unconstrained (i.e., when the system enters  $\mathcal{Z}_f$ ). Then, for an unconstrained linear system, the convergence to the origin is exponential. The most common choice of terminal weight  $Q_f$  is  $P_\infty$ , which is the solution to the ARE (2.14), and the terminal constraint set  $\mathcal{Z}_f$  is chosen as the maximal positive invariant set  $\mathcal{O}_\infty$  (see Definition 2.8) for the closed-loop system

$$z(t+1) = (A + BF_\infty)z(t), \quad (2.19)$$

where  $F_\infty$  is defined in (2.15). The domain of attraction of the controller with  $Q_f = P_\infty$  and  $\mathcal{Z}_f = \mathcal{O}_\infty$  is  $\mathcal{K}_N(\mathcal{O}_\infty)$  (see Definition 2.6).

## 2.4 Summary

The fundamental modules of an autonomous vehicle system are route planning, perception, localization, situation awareness, motion planning, and motion control. A route planner produces a high-level plan with long-term goals. The perception and localization modules fuse the information collected by the different sensors, estimate the vehicle position with respect to the world environment, and understand the vehicle surroundings. Using this information, the situation awareness and the motion planning modules are responsible for appropriately choosing the vehicle action and how to reach a predefined goal configuration, respectively. Finally, the motion control module is responsible for accurately executing the planned path or trajectory.

In this chapter, we first provided an overview of previous works that address some of the problems present in each one of the aforementioned modules. Secondly, we reviewed the main concepts and definitions of optimization theory, such as feasibility, optimality, and convexity. Lastly, we introduced the main concepts of optimal control, which helped explaining MPC, which is the base of the remaining chapters of this thesis.



## Chapter 3

# Vehicle Modeling

Model-based control is highly affected by the quality of the models provided. On the one hand, accurate models are typically computationally expensive and provide accurate predictions. On the other hand, simple models are less computationally demanding, but provide less accurate predictions.

In this chapter, we present a dynamic and a kinematic vehicle model. For simulation analysis, we develop a dynamic vehicle model based on a Scania G480 construction truck shown in Figure 3.1. The vehicle lateral dynamics are modeled as a 4-axles bicycle model with two steering axles, and the longitudinal dynamics include the model of a cruise controller (provided by Scania CV AB). For motion planning and control design, we use a nonholonomic kinematic vehicle model. We detail how the vehicle model is described in a road-aligned coordinate frame, and discuss the difference between time- and space-based modeling. Finally, we detail how the continuous model is linearized and discretized to obtain a linear representation of the vehicle dynamics in discrete-time or space.

The outline of this chapter is as follows. In Section 3.1, we describe a 4-axles bicycle model used for simulation analysis. In Section 3.2, we present the kinematic vehicle model used for control design. In Section 3.3, we compare the behavior of both models when driving at different speeds and subject to different curvature requests. In Section 3.4, we summarize the presented models.

### 3.1 Dynamic Vehicle Model

The lateral dynamics of the simulation model is described by a 4-axles nonlinear bicycle model with two steering axles in the front and two traction axles in the back. It is based on a modified Scania G480 construction truck shown in Figure 3.1.

The notation used in the 4-axles bicycle model is shown in Figure 3.2. The vehicle lateral dynamics are modeled by the following set of differential equations describing the lateral force and the momentum (Rajamani, 2011)



Figure 3.1: Modified Scania G480 construction truck used as experimental and research platform (courtesy of Scania CV AB).

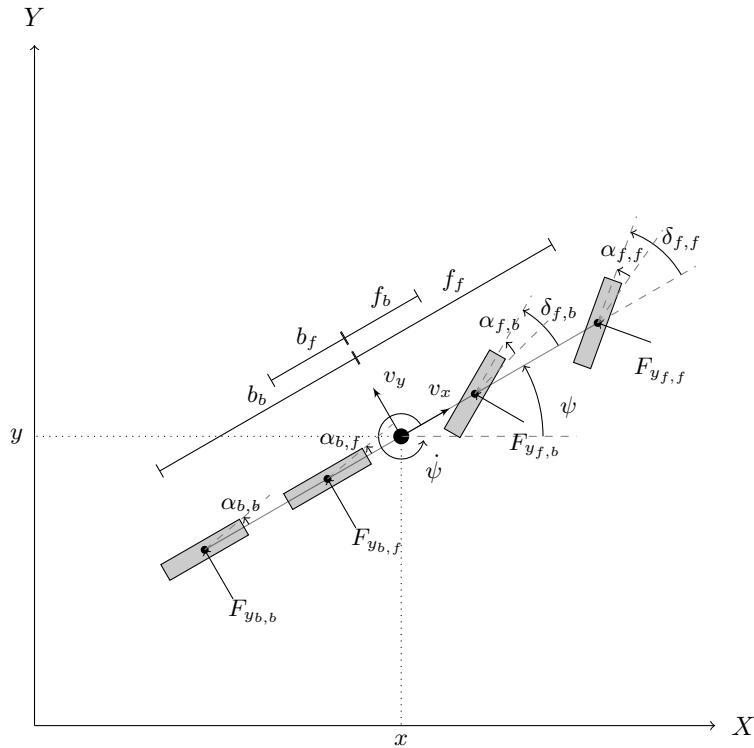


Figure 3.2: Dynamic vehicle model illustrated with a 4-axles bicycle model.

$$m(\dot{v}_y + \dot{\psi}v_x) = F_{y_{b,b}} + F_{y_{b,f}} + F_{y_{f,b}} \cos \delta_{f,b} + F_{y_{f,f}} \cos \delta_{f,f}, \quad (3.1a)$$

$$J_z \ddot{\psi} = f_f F_{y_{f,f}} \cos \delta_{f,f} + f_b F_{y_{f,b}} \cos \delta_{f,b} - b_f F_{y_{b,f}} - b_b F_{y_{b,b}}, \quad (3.1b)$$

where  $v_x$  and  $v_y$  denote the longitudinal and the lateral component of the vehicle velocity, respectively, and  $\dot{\psi}$  denotes the yaw rate. The constants  $m$  and  $J_z$  denote the vehicle mass and moment of inertia about the yaw axis, respectively, and  $f_{b/f}$  and  $b_{b/f}$  represent the distances from the center of gravity to the front and rear axles, respectively. Finally,  $\delta_{f/b}$  are the steering angles of the front axles, which are mechanically coupled and  $\delta_{f,b} = \beta \delta_{f,f}$ , where  $\beta$  defines the ratio between both steering angles. The estimation of  $\beta$  is made using real data containing the wheel angles from both front axles. Since the model focuses on the lateral dynamics,  $v_x$  is considered known and unperturbed. In this model, the side forces are assumed to be linear functions of the slip angles

$$F_{y_{f,f}} = -C_{f,f} \alpha_{f,f}, \quad (3.2a)$$

$$F_{y_{f,b}} = -C_{f,b} \alpha_{f,b}, \quad (3.2b)$$

$$F_{y_{b,f}} = -C_{b,f} \alpha_{b,f}, \quad (3.2c)$$

$$F_{y_{b,b}} = -C_{b,b} \alpha_{b,b}, \quad (3.2d)$$

where the constants  $C_{b,f/b}$  and  $C_{f,f/b}$  are the cornering stiffnesses of the rear and the front axles, respectively. The axles cornering stiffnesses have been computed by linearizing the Pacejka (2005) formula for no longitudinal slip, no camber, small slip angles, and using the steady-state vertical loads on each axle obtained through ADAMS (2015) simulation. ADAMS is a simulation environment that is specialized in detailed vehicle design. The slip angles are given by

$$\alpha_{f,f} = \arctan \left( \frac{v_y + \dot{\psi} f_f}{v_x} \right) - \delta_{f,f}, \quad (3.3a)$$

$$\alpha_{f,b} = \arctan \left( \frac{v_y + \dot{\psi} f_b}{v_x} \right) - \delta_{f,b}, \quad (3.3b)$$

$$\alpha_{b,f} = \arctan \left( \frac{v_y - \dot{\psi} b_f}{v_x} \right), \quad (3.3c)$$

$$\alpha_{b,b} = \arctan \left( \frac{v_y - \dot{\psi} b_b}{v_x} \right). \quad (3.3d)$$

The vehicle lateral dynamics are controlled by a curvature request. The relation between vehicle curvature  $\kappa$  (i.e., ratio between yaw rate and longitudinal speed at each instant), and its steering wheel angle  $\delta_{SW}$ , assuming steady-state cornering

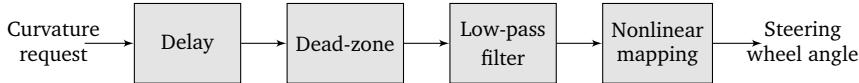


Figure 3.3: Curvature to steering wheel angle dynamics.

(i.e., driving a path with constant radius and speed) and that the vehicle has no slip (i.e.,  $v_x \gg v_y$ ) can be shown to be

$$\frac{\delta_{SW}}{i_s} = \delta_{f,f} = \frac{\arctan((L + K_{us}v_x^2)\kappa)}{C_{eq}}, \quad C_{eq} = \frac{C_{f,f} + C_{f,b}\beta}{C_{f,f} + C_{f,b}}, \quad (3.4)$$

where  $K_{us}$  is the understeering gradient of the vehicle, which depends on the vehicle mass and on its center of mass,  $L$  is the wheelbase of the truck<sup>1</sup>,  $i_s$  is the ratio between steering wheel and front wheel angle, and  $C_{eq}$  is a scaling factor. Moreover, we model the steering wheel angle to curvature dynamics with a constant delay, a low-pass filter, and a dead-zone, as shown in Figure 3.3.

The dynamic vehicle model is validated using experimental data, where the truck, shown in Figure 3.1, is subject to different curvature request functions, such as sinusoids and square waves, at low speeds (between 5 m/s and 10 m/s). Then, the dynamic vehicle model is subject to the same steering wheel angle and the same longitudinal speed as the recorded measurements. Figure 3.4 shows the comparison between the dynamic vehicle model and the experimental data, and it is clear that the dynamic vehicle model correctly captures the vehicle dynamics.

The longitudinal dynamics model are described in two steps and it was developed by Scania CV AB. First, the longitudinal model receives a speed request or an acceleration request and outputs an engine torque or a brake demand. It consists of three controllers, namely a cruise controller that outputs engine torque, a downhill speed controller that outputs auxiliary (or retarder) torque, and an external brake interface that takes acceleration as input and outputs wheel brake torque. The cruise controller is a Proportional-Integral (PI) controller that takes into account the vehicle characteristics, such as mass and maximum engine torque, as well as the road profile, to produce an engine torque that makes the vehicle track the speed request. The parameters used in the simulation are the same as the ones commercially available in the truck cruise controller. Secondly, the powertrain illustrated in Figure 3.5 is modeled. The engine receives a torque request and determines the amount of diesel that needs to be combined with air in a high pressurized chamber. The clutch is responsible for decoupling the engine from the drivetrain to enable gear shifts. The gear box connects the clutch to the propeller shaft and determines the conversion ratio between the engine torque and the propeller shaft torque. Finally, these torques are converted to wheel torque in the final drive and drive shafts.

---

<sup>1</sup>In a 4-axle bicycle model, the wheelbase is the distance between the vehicle center of rotation and equivalent front axle, which combines the two front axles.

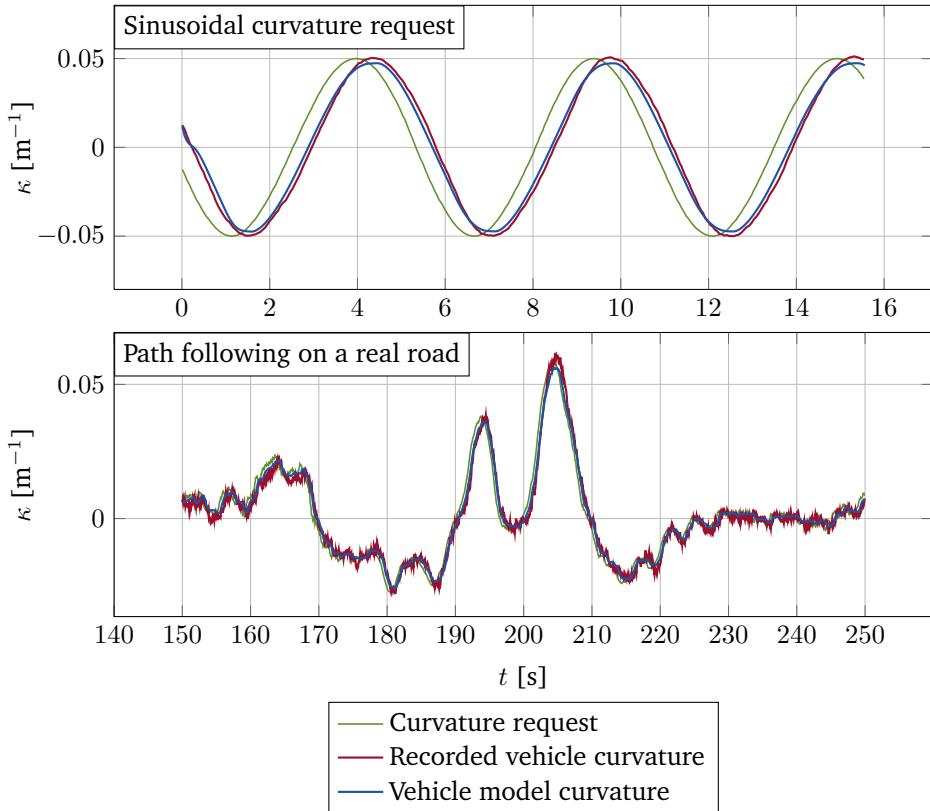


Figure 3.4: Two examples of recorded experiments with the truck. On top, a sinusoidal wave is used as curvature request. Below, the vehicle drives on a real road.

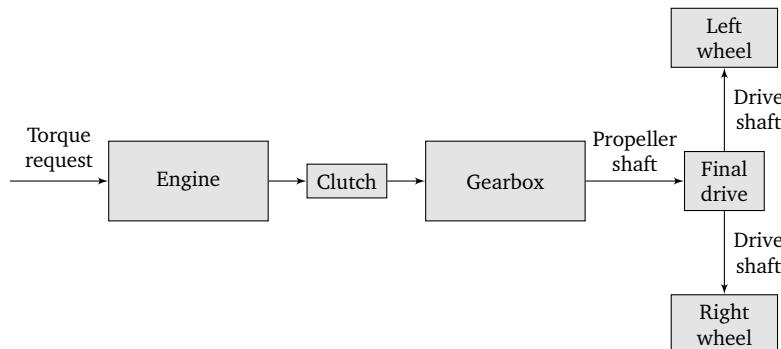


Figure 3.5: Illustration of the powertrain model.

For a detailed description of each one of these elements of the powertrain, please refer to (Alam, 2014).

### 3.2 Kinematic Vehicle Model

The movement of a nonholonomic vehicle (e.g., car, truck, and bus) at low speeds (i.e., when the lateral dynamics have little influence) can approximately be described by its time-domain kinematic equations (De Luca et al., 1998) given by

$$\begin{aligned}\dot{x} &= \frac{dx}{dt} = v_x \cos(\psi), \\ \dot{y} &= \frac{dy}{dt} = v_x \sin(\psi), \\ \dot{\psi} &= \frac{d\psi}{dt} = \frac{v_x}{L} \tan(\delta),\end{aligned}\tag{3.5}$$

where  $x$  and  $y$  are the coordinates of the vehicle in the global coordinate system,  $\psi$  is the yaw angle,  $L$  is the wheelbase,  $v_x$  is the longitudinal component of the vehicle velocity<sup>2</sup>, and  $\delta$  is the steering angle of the front wheels. We can relate the vehicle curvature  $\kappa$  with the vehicle steering angle  $\delta$  by  $\kappa = \frac{\tan(\delta)}{L}$ .

Figure 3.6 illustrates the relation between the global frame and the road-aligned frame (also known as Frenet-Serret frame or curvilinear frame). In the road-aligned frame, the variable  $s$  is introduced representing the projected vehicle position along the path. We model the lateral displacement  $e_y$  and heading displacement  $e_\psi$  between the vehicle and the road, as a function of distance traveled along the path. According to Figure 3.6, we can geometrically derive the relations

$$\begin{aligned}v_s &= (\rho_s - e_y)\dot{\psi}_s = v_x \cos(e_\psi), \\ \dot{s} &= \rho_s \dot{\psi}_s = \frac{\rho_s}{\rho_s - e_y} v_x \cos(e_\psi),\end{aligned}\tag{3.6}$$

where  $v_s$  is the projected vehicle velocity along the direction of the path,  $\rho_s$  is the radius of curvature of the road (assumed to be bigger than  $e_y$ ),  $\psi_s$  is the path heading angle, and  $\dot{s}$  is the vehicle velocity along the path. It is also possible to derive the relations

$$\begin{aligned}\dot{e}_y &= v_x \sin(e_\psi), \\ \dot{e}_\psi &= \dot{\psi} - \dot{\psi}_s.\end{aligned}\tag{3.7}$$

Throughout this thesis, we typically use a space-based kinematic vehicle model in order to eliminate the time-dependency present in (3.7). To that end, we use the chain rule to express the spatial derivative as function of the time derivative,

---

<sup>2</sup>The kinematic vehicle model assumes that the slip angles at both wheels are zero. Therefore, the lateral component of the vehicle velocity is null. The equations of motion are based purely on geometric relationships governing the system.

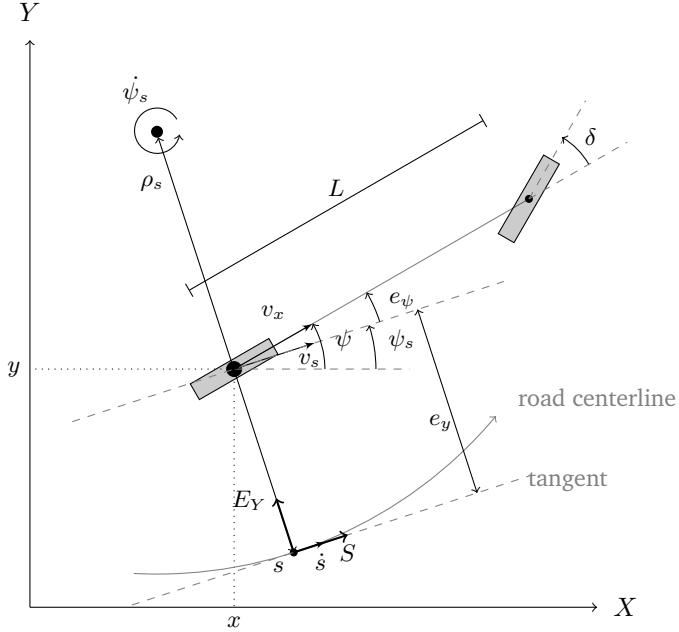


Figure 3.6: Kinematic vehicle model illustration in the global and in the road-aligned frame.

namely  $\frac{d(\cdot)}{ds} = \frac{d(\cdot)}{dt} \frac{dt}{ds} = \frac{d(\cdot)}{dt} \frac{1}{\dot{s}}$ . Hence, the space-based representation of (3.7) can be derived as

$$\begin{aligned} e'_y &= \frac{\dot{e}_y}{\dot{s}} = \frac{\rho_s - e_y}{\rho_s} \tan(e_\psi), \\ e'_\psi &= \frac{\dot{e}_\psi}{\dot{s}} = \frac{(\rho_s - e_y)}{\rho_s \cos(e_\psi)} \kappa - \dot{\psi}_s. \end{aligned} \quad (3.8)$$

In the remainder of this thesis, we consider the state vector  $z = [e_y, e_\psi]^T$ , and the input  $u = \kappa$ , such that  $z' = f(z, u)$ .

### 3.2.1 Linearization and Discretization

In this thesis, a discrete linear time-varying version of (3.8) is typically used.

Consider a reference  $(z_r, u_r)$ , which respects the dynamics given by (3.8). Then, the model (3.8) can be linearized using a first-order Taylor approximation with respect to  $(z_r, u_r)$

$$\begin{aligned} f(z, u) &\approx f(z_r, u_r) + \frac{\partial f(z, u)}{\partial z}\Big|_{z_r, u_r}(z - z_r) + \frac{\partial f(z, u)}{\partial u}\Big|_{z_r, u_r}(u - u_r) \Leftrightarrow \\ &\Leftrightarrow f(z, u) \approx f(z_r, u_r) + A_c(z - z_r) + B_c(u - u_r), \end{aligned} \quad (3.9)$$

which allows describing (3.8) using a linear model of the form

$$z' = A_c z + B_c u + h_c, \quad (3.10)$$

where  $h_c = f(z_r, u_r) - (A_c z_r + B_c u_r)$ .

To perform the discretization of a continuous-time system, it is common to assume that the control signal is constant between samples (Åström and Wittenmark, 2013). Let the discretization step be  $\Delta_s$ , then using Zero-Order Hold (ZOH) (Van Loan, 1978), it is possible to write (3.10) as a linear time-varying (or linear space-varying) model

$$z(k+1) = A_d^{\text{ZOH}}(k)z(k) + B_d^{\text{ZOH}}(k)u(k) + h_d^{\text{ZOH}}(k), \quad k \geq 0, \quad (3.11)$$

where

$$\begin{aligned} A_d^{\text{ZOH}}(k) &= \exp(A_c(k)\Delta_s), \\ B_d^{\text{ZOH}}(k) &= \left( \int_0^{\Delta_s} \exp(A_c(k)\eta) d\eta \right) B_c(k), \\ h_d^{\text{ZOH}}(k) &= \left( \int_0^{\Delta_s} \exp(A_c(k)\eta) d\eta \right) h_c(k), \end{aligned} \quad (3.12)$$

where  $A_c(k)$ ,  $B_c(k)$ , and  $h_c(k)$  are obtained by successively applying (3.9) using a discretized version of the reference  $(z_r(k), u_r(k))$ . When  $A_c(k)$  is singular, it is possible to obtain (3.12) by computing an auxiliary matrix  $M$  at every sampling instant

$$M(k) = \exp \left( \begin{bmatrix} A_c(k) & I \\ 0 & 0 \end{bmatrix} \Delta_s \right).$$

Let  $M_{12}$  be the upper-right block matrix of  $M$ , then it can be shown that

$$\int_0^{\Delta_s} \exp(A_c(k)\eta) d\eta = M_{12}(k),$$

and therefore

$$\begin{aligned} A_d^{\text{ZOH}}(k) &= \exp(A_c(k)\Delta_s), \\ B_d^{\text{ZOH}}(k) &= M_{12}(k)B_c(k), \\ h_d^{\text{ZOH}}(k) &= M_{12}(k)h_c(k). \end{aligned} \quad (3.13)$$

We use ZOH discretization of the vehicle model in Chapters 5 and 6.

Another option is to use Forward Euler (FE) discretization. According to Kong et al. (2015), a kinematic vehicle model accurately describes the movement of a nonholonomic vehicle when discretized using FE with a discretization time step of 200 ms. This model targets vehicle motion prediction when driving curvy paths at low speeds or straight paths at high speeds (i.e., when the lateral dynamics have little influence).

Let the discretization step be  $\Delta_s$ , then using FE discretization, it is also possible to write (3.10) as a linear time-varying (or linear space-varying) model

$$z(k+1) = A_d^{\text{FE}}(k)z(k) + B_d^{\text{FE}}(k)u(k) + h_d^{\text{FE}}(k), \quad k \geq 0, \quad (3.14)$$

where

$$\begin{aligned} A_d^{\text{FE}}(k) &= A_c(k)\Delta_s + I, \\ B_d^{\text{FE}}(k) &= B_c(k)\Delta_s, \\ h_d^{\text{FE}}(k) &= h_c(k)\Delta_s. \end{aligned} \quad (3.15)$$

We use FE discretization of the vehicle model in Chapters 7 and 8.

### 3.3 Model Comparison and Discussion

Figure 3.7 shows simulation results using the dynamic and the kinematic vehicle models. There, the vehicle models are run at different speeds and, at 7.5 seconds of simulation, a curvature request is given to both models for another 7.5 seconds. Table 3.1 presents the Euclidean distance between the last position of the kinematic and the dynamic model. As expected, the dynamic and the kinematic vehicle models are similar at low speeds (5 m/s), since the slip angles and the tire forces modeled in the dynamic vehicle model have neglectable influence on the vehicle motion at these speeds. However, at high speeds (15 m/s), the behavior of the models is totally distinct. Therefore, at high speeds, it is not possible to accurately predict the motion of a realistic vehicle using a kinematic vehicle model. In this case, the dynamic vehicle model presents a considerable understeering behavior, which is not modeled by the kinematic vehicle model. Finally, at moderate speeds (10 m/s), the behavior of both vehicle models is similar when the curvature is small. Nevertheless, when the curvature is considerable, the kinematic vehicle model is unable to accurately replicate the lateral dynamics of the vehicle.

In this thesis, we typically use the kinematic vehicle model to develop motion planning algorithms and design motion controllers. We assume that the lateral dynamics of the vehicle are neglectable when driving straight or with small curvature. To that end, low speed is enforced when turning to minimize the vehicle lateral slip. With these two conditions, the kinematic vehicle model is able to accurately predict the vehicle motion. Moreover, note that the kinematic vehicle model is typically used in a receding-horizon fashion to predict the vehicle motion. Therefore, the current vehicle state is frequently updated (e.g., every 0.1 seconds) providing robustness to model mismatch.

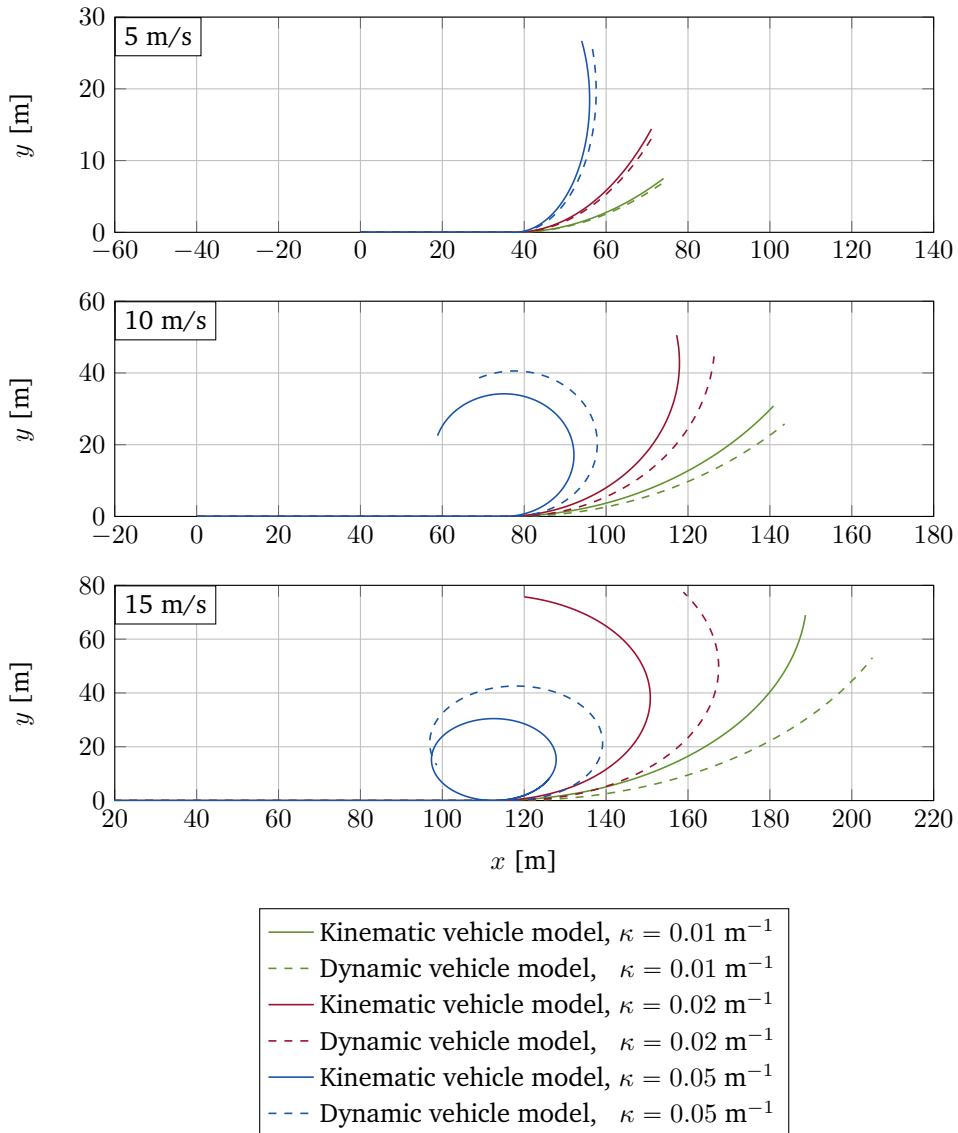


Figure 3.7: Comparison between the dynamic and the kinematic vehicle models for different velocities and curvatures.

Curvature/Velocity	5 m/s	10 m/s	15 m/s
$0.01 \text{ m}^{-1}$	0.50 m	5.70 m	22.79 m
$0.02 \text{ m}^{-1}$	1.02 m	10.87 m	38.89 m
$0.05 \text{ m}^{-1}$	2.69 m	18.93 m	27.97 m <sup>3</sup>

Table 3.1: Euclidean distance between the kinematic and the dynamic vehicle model in the end of the simulation.

### 3.4 Summary

In this chapter, we described two different vehicle models. A dynamic vehicle model, which is a 4-axle bicycle model integrated with steering column dynamics, was presented to describe the vehicle lateral dynamics. To validate the dynamic vehicle model, we compared its behavior with real vehicle data, where a truck was subject to different curvature requests. We briefly detailed the longitudinal model provided by Scania CV AB, which contains a cruise controller model and vehicle longitudinal dynamics. In addition, we presented a kinematic vehicle model, which is used as a model for motion planning and control design in the technical chapters of this thesis. In the end, we compared the behavior of both models at different speeds and subject to different curvature requests. The vehicle behavior is quite different at high speeds and considerable curvatures, due to the non-neglectable lateral dynamics. Nevertheless, the kinematic vehicle model is able to reproduce accurately the dynamic vehicle model behavior at low speeds or with small curvatures.

---

<sup>3</sup>This case is misleading. In the end of the simulation, the kinematic and the dynamic vehicle model are closer to each other than in the simulation with smaller curvature and the same speed by chance.



## Chapter 4

# Clothoid-Based Path Sparsification

Proper path representation is crucial for autonomous driving. Clothoids are used in road design because of their smoothness property, since their curvature varies linearly with the traveled distance. This provides a linear change in the vehicle steering angle when performing a turn and prevents sudden changes in lateral acceleration. Therefore, when used in transition curves (e.g., highway exit ramps), clothoids yield low values of lateral jerk. For these reasons, we propose to use clothoid segments to describe reference paths. In fact, we address the problem of describing the path using the minimum number of clothoid segments.

The rest of this chapter is organized as follows. In Section 4.1, we define clothoids mathematically, providing related literature about clothoid fitting. In Section 4.2, we formulate the problem of clothoid-based path sparsification and detail how it is solved using  $\ell_0$ -norm minimization techniques. In Section 4.3, we provide simulation results using the proposed algorithm. Section 4.4 concludes this chapter with a summary of its content.

### 4.1 Clothoids

The general parametric form of a clothoid curve is given by

$$x(s) = x_0 + \int_0^s \cos(\theta(\eta)) d\eta, \quad (4.1a)$$

$$y(s) = y_0 + \int_0^s \sin(\theta(\eta)) d\eta, \quad (4.1b)$$

$$\theta(s) = \theta_0 + \int_0^s \kappa(\eta) d\eta, \quad (4.1c)$$

$$\kappa(s) = \kappa_0 + cs, \quad (4.1d)$$

where  $(x(s), y(s))$  are the Cartesian coordinates of the curve, and  $\theta(s)$  is the angle defined by the tangent along the curve with respect to the  $x$ -axis, where  $(x_0, y_0, \theta_0)$

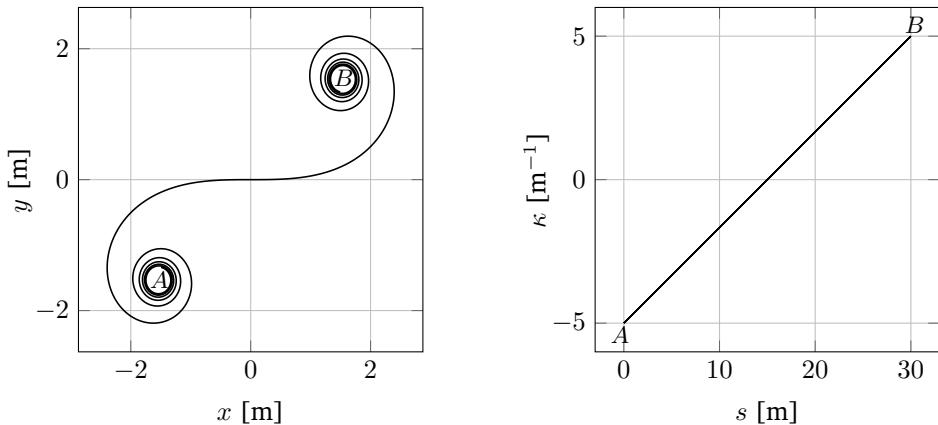


Figure 4.1: Example of a clothoid curve. The curvature grows linearly from  $-5 \text{ m}^{-1}$  to  $5 \text{ m}^{-1}$  and the clothoid length is 30 meters, when traveling from  $A$  to  $B$ .

is the initial pose. The curvature  $\kappa$  varies linearly with respect to the traveled distance  $s$  with a constant curvature change rate  $c$ , as depicted in Figure 4.1. The curve is parametrized in the space domain  $s \in \mathbb{R}^+$ , which represents the distance traveled along the path. Solving (4.1c) shows that  $\theta$  is a quadratic function of  $s$ . When plugging this into (4.1a) and (4.1b), we obtain the so-called Fresnel integrals, which do not have a closed-form expression (Heald, 1985).

Clothoids are known in the literature under various names, for instance “Euler spiral” or “Cornu spiral”. Mathematically, the underlying equation describing a clothoid is known as a Fresnel integral. The many different names are due to the historical background of this curve, which was reinvented many times with completely different purposes. There is, however, one discovery that directly connects clothoids with autonomous driving. Talbot (1904) used clothoids to design transition curves in railway tracks. Due to the curvature discontinuity between straight lines and circular arcs, clothoids are used as transition curves between these two types of segments to minimize the unpleasant lurch of trains. Today, clothoids are also the standard transition curve in road design (Kühn, 2013). In the case of ground vehicles, clothoidal road design provides a linear change in the vehicle steering angle when performing a turn, preventing sudden changes in lateral acceleration and therefore, yielding low values of lateral jerk (Marzbani et al., 2015).

Early work on path planning for nonholonomic vehicles focused on the generation of Dubins curves (i.e., paths obtained by connecting circular arcs and straight lines) (Dubins, 1957). However, these type of paths are not curvature continuous. Therefore, to follow them, a nonholonomic vehicle must stop and reorient its steering wheels; otherwise, the path tracking is not possible without position and orientation errors at the curvature discontinuities, since instantaneous changes in steering angles are physically impossible. This problem stimulated research on

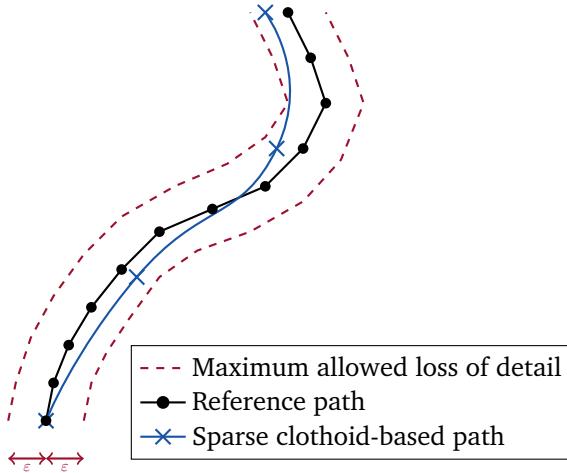


Figure 4.2: The reference path is a dense set of linearly interpolated waypoints, while the intended representation is a sparse set of clothoid segments that can deviate at most  $\epsilon$  from the original path.

smooth path planning (Kanayama and Hartman, 1989), for which clothoids are frequently chosen (Chu et al., 2012; Jo et al., 2015; Labakhua et al., 2008; Scheuer and Fraichard, 1996). Clothoids have the disadvantage of not having a closed-form expression, but the advantage of being the curve with the simplest curvature function. Besides trajectory planning, clothoids are extremely useful in a wide range of applications, such as road estimation (Kirchner and Heinrich, 1998; Manz et al., 2010), lane recognition (Goldbeck et al., 2000; Takahashi and Ninomiya, 1996), or computer-aided design (Baran et al., 2010; McCrae and Singh, 2009).

The problem of clothoid fitting has been a subject of research in recent years. Clothoid fitting addresses the interpolation between two given points using one or more clothoid segments. When the fitting is done via two points with assigned tangent directions is called  $G^1$  interpolation (Bertolazzi and Frego, 2015). When the fitting is done via two points with assigned curvature is called  $G^2$  interpolation (McCrae and Singh, 2009). This is a complex problem that is usually addressed using iterative numerical algorithms to find the solution to one or more nonlinear equations. The problem of path sparsification, presented in this chapter, is a relaxed version of the fitting problem. We seek the minimum number of clothoid segments that describe the original path within a small deviation margin.

## 4.2 Problem Formulation

In this section, we formulate the path sparsification problem, illustrated in Figure 4.2, as an optimization problem.

We deal with three different paths of waypoints represented by the tuple of Cartesian coordinates, heading angle, and curvature:

- A reference path,  $(x_{r,i}, y_{r,i}, \theta_{r,i}, \kappa_{r,i})$ ,  $i = 1, \dots, N$ ,
- A dense clothoid-based path,  $(x_{d,i}, y_{d,i}, \theta_{d,i}, \kappa_{d,i})$ ,  $i = 1, \dots, N$ ,
- A sparse clothoid-based path,  $(x_{s,i}, y_{s,i}, \theta_{s,i}, \kappa_{s,i})$ ,  $i = 1, \dots, M$ , where  $M \leq N$ .

The Euclidean distance between waypoints  $i$  and  $i + 1$  of the reference path is  $\Delta s_{r,i}$ . The dense clothoid-based path consists of clothoid segments, where the second-order differences of the curvature vector is exactly zero everywhere except at a few points. These points are called kink-points and are used to form the sparse clothoid-based path. A pair of kink-points, together with the arc-length between them, allow the computation of a clothoid segment. Therefore, if  $\mathcal{M}$  is a set containing the indices of dense clothoid-based path waypoints that have non-zero second-order differences of the curvature vector, i.e.,

$$\mathcal{M} = \{i \in \{2, \dots, N - 1\} : \kappa_{d,i-1} - 2\kappa_{d,i} + \kappa_{d,i+1} \neq 0\} \cup \{1, N\}.$$

Then, we are interested in minimizing  $M = |\mathcal{M}|$ , to describe a clothoid-based path with the minimum number of kink-points. These kink-points describe the reference path with an accuracy measured by the maximum allowed deviation  $\varepsilon$  with respect to  $(x_{r,i}, y_{r,i})$ . The first dense clothoid-based path waypoint pose (i.e.,  $(x_{d,1}, y_{d,1}, \theta_{d,1})$ ) is constrained to be equal to the first reference waypoint pose (i.e.,  $(x_{r,1}, y_{r,1}, \theta_{r,1})$ ). Also, the last dense clothoid-based waypoint heading angle (i.e.,  $\theta_{d,N}$ ) is constrained to be equal to the last reference path waypoint heading angle (i.e.,  $\theta_{r,N}$ ). The problem is summarized as follows

$$\min M \tag{4.2a}$$

$$\text{s.t. } |x_{r,i} - x_{d,i}| \leq \varepsilon, \quad i = 1, \dots, N, \tag{4.2b}$$

$$|y_{r,i} - y_{d,i}| \leq \varepsilon, \quad i = 1, \dots, N, \tag{4.2c}$$

$$x_{d,1} = x_{r,1}, \quad y_{d,1} = y_{r,1}, \tag{4.2d}$$

$$\theta_{d,1} = \theta_{r,1}, \quad \theta_{d,N} = \theta_{r,N}, \tag{4.2e}$$

$$M = |\mathcal{M}|. \tag{4.2f}$$

The dense clothoid-based path is based on the discretization of the clothoid parametric equations (4.1a) and (4.1b), which uses the fact that integrals can be approximated by Riemann sums. Thus, the position  $(x_{d,i}, y_{d,i})$  after applying

$\bar{\kappa}_{d,i} = [\kappa_{d,1}, \dots, \kappa_{d,i}]^T$  with  $i = 1, \dots, N$  is described by

$$x_{d,i+1} = x_{d,1} + \sum_{k=1}^i \cos \left( \sum_{j=1}^k \kappa_{d,j} \Delta s_{d,j} \right) \Delta s_{d,k}, \quad i = 1, \dots, N-1, \quad (4.3a)$$

$$y_{d,i+1} = y_{d,1} + \sum_{k=1}^i \sin \left( \sum_{j=1}^k \kappa_{d,j} \Delta s_{d,j} \right) \Delta s_{d,k}, \quad i = 1, \dots, N-1, \quad (4.3b)$$

$$\theta_{d,i+1} = \theta_{d,1} + \sum_{k=1}^i \kappa_{d,k} \Delta s_{d,k} \quad i = 1, \dots, N-1, \quad (4.3c)$$

where  $\Delta s_{d,i}$  is the Euclidean distance between waypoints  $i$  and  $i+1$  of the dense clothoid-based path and is assumed to be equal to  $\Delta s_{r,i}$ .

Let  $S \in \mathbb{R}^{N \times N}$  be a lower triangular matrix that performs the cumulative sum operation along a vector

$$S = \begin{bmatrix} 0 \\ \Delta s_{d,1} \\ \Delta s_{d,1} & \Delta s_{d,2} \\ \Delta s_{d,1} & \Delta s_{d,2} & \Delta s_{d,3} \\ \vdots & \vdots & \vdots & \ddots \\ \Delta s_{d,1} & \Delta s_{d,2} & \Delta s_{d,3} & \Delta s_{d,N} \end{bmatrix}. \quad (4.4)$$

Then, a more compact form of (4.3) is

$$\bar{x}_d(\bar{\kappa}_d) = S \cos(S\bar{\kappa}_d), \quad (4.5a)$$

$$\bar{y}_d(\bar{\kappa}_d) = S \sin(S\bar{\kappa}_d), \quad (4.5b)$$

$$\bar{\theta}_d(\bar{\kappa}_d) = S\bar{\kappa}_d, \quad (4.5c)$$

where  $\bar{x}_d = [x_{d,1}, \dots, x_{d,N}]^T$ ,  $\bar{y}_d = [y_{d,1}, \dots, y_{d,N}]^T$ , and  $\bar{\kappa}_d = [\kappa_{d,1}, \dots, \kappa_{d,N}]^T$ . Note that the Cartesian coordinates of the waypoints only depend on the curvature.

The problem of fitting a sparse piecewise affine (PWA) function to data involves  $\ell_0$ -norm optimization. Minimizing the  $\ell_0$ -norm of the second-order differences of the curvature vector, is equivalent to minimizing the number of kink-points. The  $\ell_0$ -norm is commonly defined as the number of non-zero elements in a vector. According to our previous definition  $M = |\mathcal{M}| = \|D_2 \bar{\kappa}_d\|_0$ .

**Remark.** Strictly speaking,  $\ell_0$ -norm is not actually a norm. It is a cardinality function which has its definition in the form of  $\ell_p$ -norm, when  $p \rightarrow 0$ .

Thus, we propose to solve the optimization problem

$$\min_{\bar{\kappa}_d} \|D_2 \bar{\kappa}_d\|_0 \quad (4.6a)$$

$$\text{s. t. } |x_{d,i} - x_{r,i}| \leq \varepsilon, \quad i = 1, \dots, N, \quad (4.6b)$$

$$|y_{d,i} - y_{r,i}| \leq \varepsilon, \quad i = 1, \dots, N, \quad (4.6c)$$

$$x_{d,1} = x_{r,1}, \quad y_{d,1} = y_{r,1}, \quad (4.6d)$$

$$\theta_{d,1} = \theta_{r,1}, \quad \theta_{d,N} = \theta_{r,N}, \quad (4.6e)$$

where  $\bar{\kappa}_d$  is the vector of curvatures to be estimated,  $\varepsilon \in \mathbb{R}^+$  is the maximum allowed error between the reference and the dense clothoid-based path, the inequalities (4.6b) and (4.6c) are performed elementwise, and  $D_2 \in \mathbb{Z}^{(N-2) \times N}$  is a matrix operator that calculates second-order differences of a vector, which is defined as

$$D_2 = \begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 & 1 \end{bmatrix}. \quad (4.7)$$

The problem (4.6) is non-deterministic polynomial-time hard (NP-hard) (Ge et al., 2011) and therefore, difficult to solve because of its computational complexity. These type of problems are usually relaxed using the  $\ell_1$ -norm, which results in convex relaxations of the original problem. The  $\ell_1$ -norm of a vector  $x \in \mathbb{R}^n$  is defined as

$$\|x\|_1 = \sum_{i=1}^n |x_i|.$$

As proposed by Candes et al. (2008), a method to solve these problems consists in applying an algorithm from the family of majorization-minimization algorithms, where the minimization of a generic function (e.g.,  $\ell_0$ -norm) is performed by iteratively minimizing a convex majorizer function (e.g.,  $\ell_1$ -norm). In our case, the problem (4.6) is rewritten as

$$\begin{aligned} \min_{\bar{\kappa}_d} & \quad \|\bar{W}_k \odot D_2 \bar{\kappa}_d\|_1 \\ \text{s.t. } & \quad |x_{d,i} - x_{r,i}| \leq \varepsilon, \quad i = 1, \dots, N, \\ & \quad |y_{d,i} - y_{r,i}| \leq \varepsilon, \quad i = 1, \dots, N, \\ & \quad x_{d,1} = x_{r,1}, \quad y_{d,1} = y_{r,1}, \\ & \quad \theta_{d,1} = \theta_{r,1}, \quad \theta_{d,N} = \theta_{r,N}, \end{aligned} \quad (4.8)$$

where  $\bar{W}_k \in \mathbb{R}^{N-2}$  is a vector containing weights that are initially (i.e., when  $k = 1$ ) set to 1 and  $\odot$  is the Hadamard product. The problem (4.8) is solved iteratively, where  $\bar{W}_k$  is updated in each iteration as

$$\bar{W}_{k+1} = \frac{1}{|\bar{W}_k \odot D_2 \bar{\kappa}_d|}, \quad k = 1, \dots, K, \quad (4.9)$$

where  $K$  is the maximum number of iterations allowed. At each iteration, the weighting vector is normalized such that the weights sum up to  $N$ . In other words, the weights are updated such that the entries of the smoothness term close to zero receive a higher weight, while the others obtain a smaller weight. This results in a reweighed  $\ell_1$ -norm that, as  $k$  increases, tends to be identical to the  $\ell_0$ -norm.

**Remark.** As discussed by Candes et al. (2008), there are several properties of this algorithm that need more research and understanding. Here, the algorithm is analyzed from a practical point of view and we do not claim any specific property, such as convergence or optimality.

Problem (4.8) is not convex, since the reconstructed path  $(\bar{x}_d(\bar{\kappa}_d), \bar{y}_d(\bar{\kappa}_d))$  is computed using (4.5), which are clearly nonlinear equations. A linear approximation is done using a first-order Taylor approximation of both cosine and sine around the reference path curvature  $\bar{\kappa}_r$ . The linearization yields

$$\bar{x}_d(\bar{\kappa}_d) \approx \bar{x}_d^{\text{lin}}(\bar{\kappa}_d) = S(\cos(S\bar{\kappa}_r) - \sin(S\bar{\kappa}_r)S(\bar{\kappa}_d - \bar{\kappa}_r)), \quad (4.10a)$$

$$\bar{x}_d(\bar{\kappa}_d) \approx \bar{y}_d^{\text{lin}}(\bar{\kappa}_d) = S(\sin(S\bar{\kappa}_r) + \cos(S\bar{\kappa}_r)S(\bar{\kappa}_d - \bar{\kappa}_r)). \quad (4.10b)$$

Finally, we formulate the original problem (4.6) as a convex problem

$$\min_{\bar{\kappa}_d} \quad ||\bar{W}_k \odot D_2 \bar{\kappa}_d||_1 \quad (4.11a)$$

$$\text{s.t.} \quad |x_{d,i}^{\text{lin}} - x_{r,i}| \leq \varepsilon, \quad i = 1, \dots, N, \quad (4.11b)$$

$$|y_{d,i}^{\text{lin}} - y_{r,i}| \leq \varepsilon, \quad i = 1, \dots, N, \quad (4.11c)$$

$$x_{d,1}^{\text{lin}} = x_{r,1}, \quad y_{d,1}^{\text{lin}} = y_{r,1}, \quad (4.11d)$$

$$\theta_{d,1} = \theta_{r,1}, \quad \theta_{d,N} = \theta_{r,N}, \quad (4.11e)$$

where, at each iteration, the clothoid equations (4.5) are linearized around the solution from the previous iteration.

At iteration  $k$ , let  $\bar{J} = |D_2 \bar{\kappa}_d^*|$ , where  $\bar{J} = [J_1, \dots, J_N]^T$ . The kink-points are the elements of the optimal curvature vector  $\bar{\kappa}_d^*$ , for which the smoothness term  $J_i$  is bigger than a small threshold  $\delta$ . These are the elements with index belonging to the set  $\mathcal{M}$ . In the end, we obtain sparse clothoid-based path kink-points of the form  $(x_{s,i}, y_{s,i}, \theta_{s,i}, \kappa_{s,i})$ ,  $i \in \mathcal{M}$ . We also save the information about the arc-length between two consecutive kink-points  $\Delta s_{s,i}$ , which is the sum of the Euclidean distances between all the intermediate waypoints of the dense clothoid-based path.

We summarize the algorithm as follows:

1. Initialize  $\bar{W}_1$  with a vector of ones;
2. Linearize the model (4.5) around  $\bar{\kappa}_r$  to obtain (4.10);
3. For  $k = 1, \dots, K$ :
  - a) Solve (4.11);
  - b) Update  $\bar{W}_{k+1}$  as (4.9);
  - c) Set  $\bar{\kappa}_r = \bar{\kappa}_d^*$  to obtain a new linear clothoid model (4.10).
4. Output  $(x_{s,i}, y_{s,i}, \theta_{s,i}, \kappa_{s,i})$  and  $\Delta s_{s,i}$ ,  $i \in \mathcal{M}$ .

Using only the kink-points, it is possible to reconstruct the clothoid-based path, where the rational approximation proposed by Heald (1985) can be used to approximate the Fresnel integrals (i.e., avoiding numerical integration) in (4.1a) and (4.1b).

### 4.3 Simulation Results

In this section, we present simulation results using the clothoid-based path sparsification algorithm described in Section 4.2.

First, an illustrative example is discussed, where a *double S-curve* composed of nine clothoids segments is sparsified. Then, we apply the algorithm to sparsify two paths obtained using real data from Scania test track facilities located in Södertälje, Sweden, south of Stockholm. The first, a *precision test track*, consists of a narrow gravel road with an approximate length of 1.5 km with several sharp curves designed to perform tests with autonomous mining trucks. The second, a *high-speed test track*, consists of a regular asphalt road with two long straights and two sharp U-turns. These paths were recorded using a real-time kinematic GPS installed in a truck.

Figure 4.3 depicts the double S-curve reference path clothoid-based sparsification. The maximum number of iterations is  $K = 6$  for illustration purposes, since typically the number of kink-points stops decreasing after 2 or 3 iterations. The reference path is discretized every 1 meter. On the upper plot, we illustrate the influence of the parameter  $\varepsilon$  on the algorithm solution. On the lower plot, it can be seen that the sparse clothoid-based path does not violate the imposed  $\varepsilon$  constraint. As expected, the larger the  $\varepsilon$  the more the clothoid-based path is allowed to deviate from the reference path and fewer kink-points are necessary to describe it. Figure 4.4 brings extra information about the kink-points determination. The upper and middle plot show that smaller weights correspond to the points with  $|D_2\bar{\kappa}_d| \geq \delta$ . The lower plot shows a monotonic decrease of the number of kink-points with the number of iterations. Note that the number of kink-points depicted

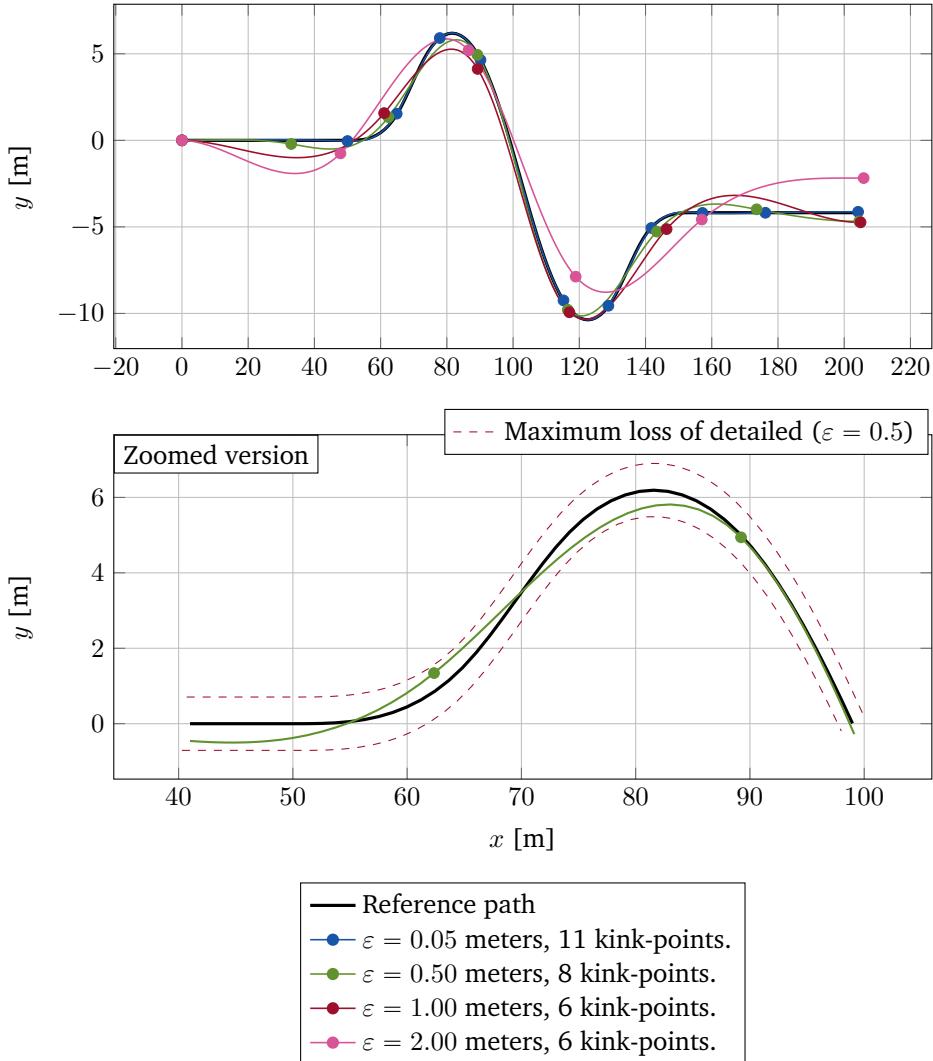


Figure 4.3: *Double S-curve*. The upper plot depicts the reference path and several clothoid-based path sparsifications for different  $\varepsilon$ . The lower plot shows a zoomed section of the upper plot for  $\varepsilon = 0.5$  meters.

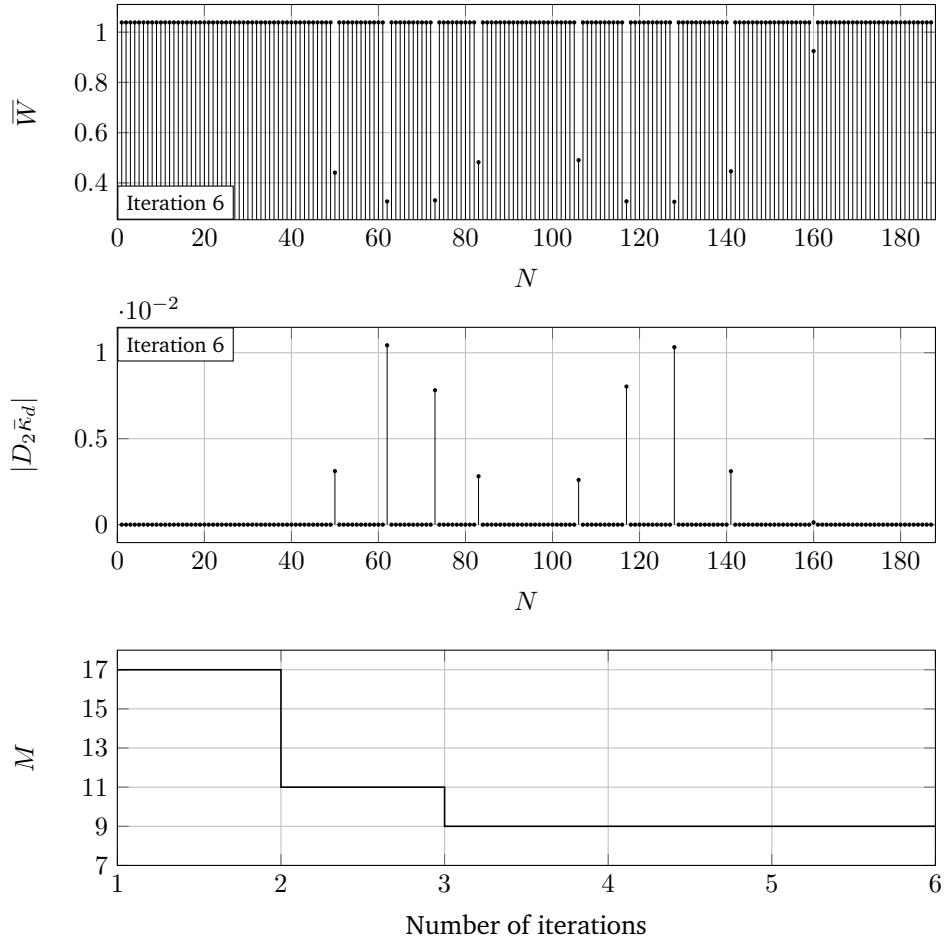


Figure 4.4: *Double S-curve*. The upper and middle plot depict the value of the vector  $\bar{W}$  and  $|D_2 \bar{\kappa}_d|$ , respectively, at the 6<sup>th</sup> iteration of (4.11). Note that when  $|D_2 \bar{\kappa}_d| \geq \delta \approx 0$  the corresponding point is a kink-point. The lower plot represents the evolution of the number of kink-points with increasing iterations of (4.11). In this simulation,  $\varepsilon = 0.05$  meters.

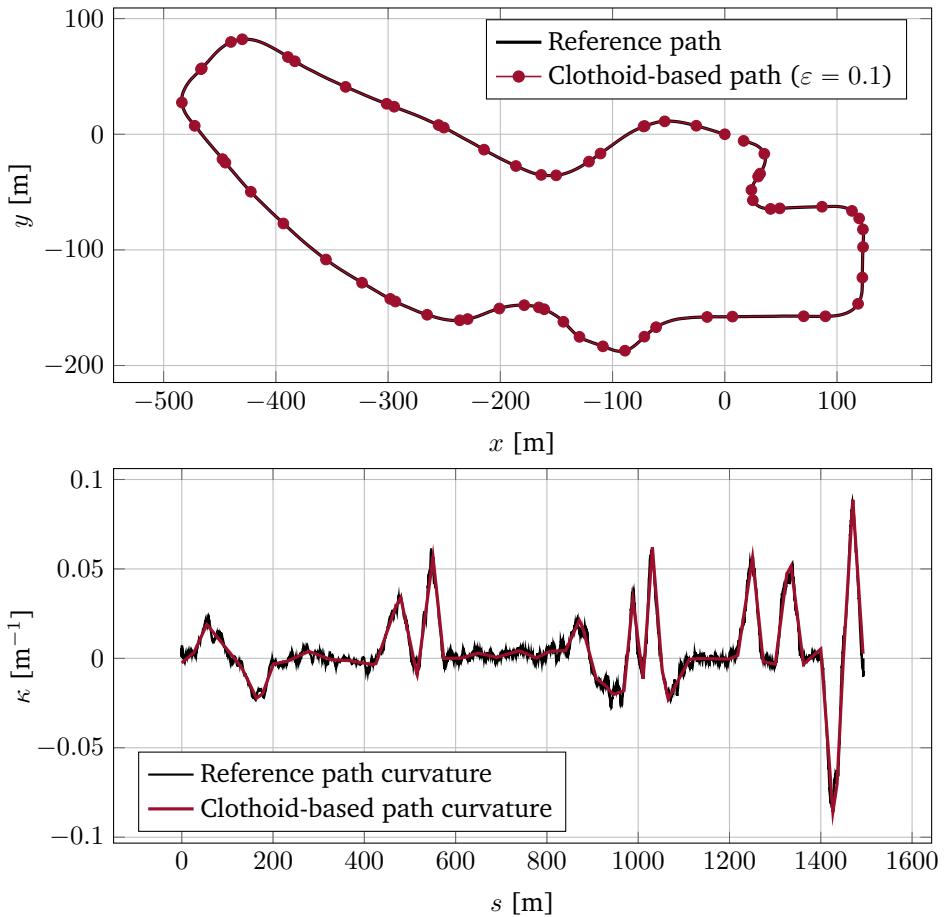


Figure 4.5: *Precision test track*. The upper plot depicts the result of the clothoid-based path sparsification. The lower plot shows the reference curvature and the PWA curvature after sparsification.

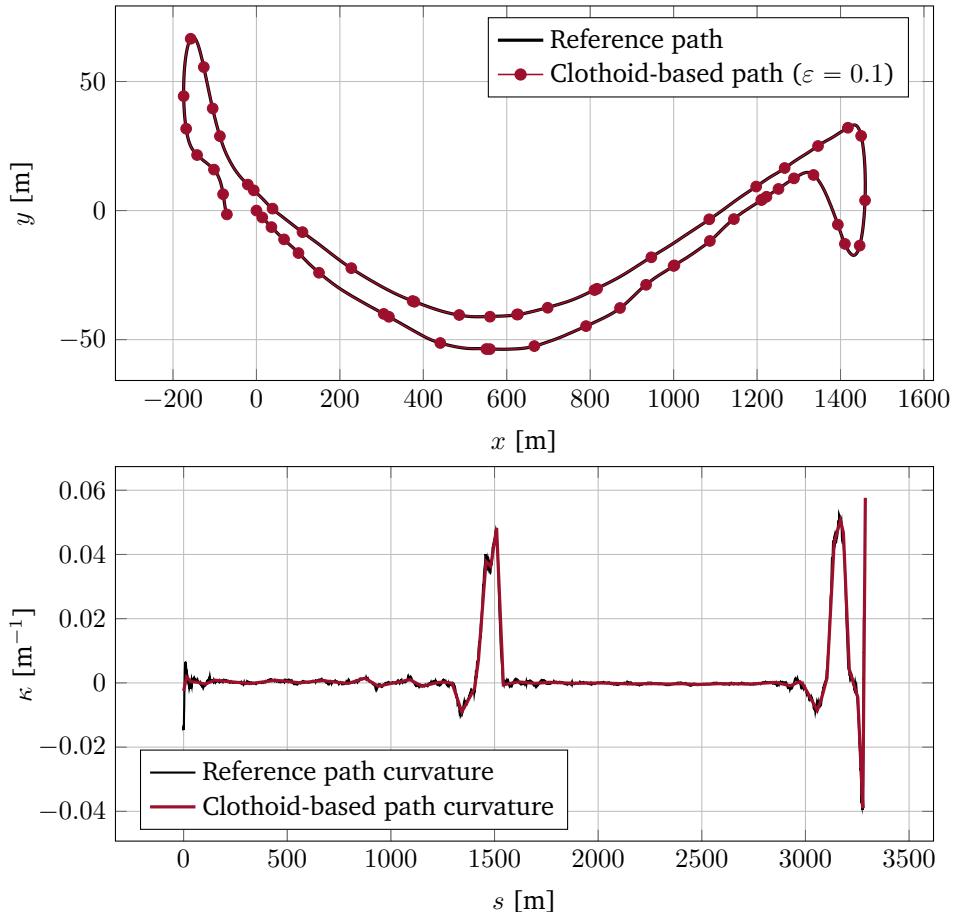


Figure 4.6: *High-speed test track*. The upper plot depicts the result of the clothoid-based path sparsification. The lower plot shows the reference curvature and the PWA curvature after sparsification.

	Precision test track	High-speed test track
Points of reference path ( $N$ )	1459	3024
Points of sparse path ( $M$ )	55	59

Table 4.1: Reference path information for  $\varepsilon = 0.1$  meters.

in Figure 4.4 does not include the first and the last point, which are also considered kink-points.

Figures 4.5 and 4.6 depict the precision and the high-speed test track sparsification using  $\varepsilon = 0.1$  meters. The maximum number of iterations allowed is  $K = 3$ , since typically the number of kink-points stops decreasing after 2 or 3 iterations. In addition to correctly describing the reference path, the sparse clothoid-based path also results in a smoother curvature function. This is because the new path is composed of clothoid segments, each of which has linear curvature (cf. Figure 4.1). Table 4.1 presents the number of points of the original and the sparsified paths. While the compression ratio (i.e.,  $M/N$ ) between the sparsified path and the original path is about 2%-4%, the sparsified path has almost no loss of detail, since the maximum deviation allowed is small.

## 4.4 Summary

In this chapter, we have presented a clothoid-based path sparsification algorithm. The path sparsification problem was formulated as an  $\ell_0$ -norm curvature regularization problem (i.e., an optimization problem that consists in finding a PWA curvature function). The  $\ell_0$ -norm is used to induce sparsity in the solution, since we want the cost function (i.e., the second-order differences of the curvature vector) to always be zero except at a few points, the so-called kink-points. The goal is to minimize the number of kink-points such that a clothoid-based path can fit the reference path while allowing for small deviations from it. However, optimization problems using the  $\ell_0$ -norm are computationally intractable and therefore, a relaxed version of the problem was solved using a majorization-minimization algorithm, where a weighted  $\ell_1$ -norm optimization problem is solved iteratively. In the end of the chapter, simulation results using real data were presented. The results showed that it is possible to sparsify the paths with a compression ratio of about 2%-3%, allowing for, at most, 10 cm deviations.



## Chapter 5

# Optimization-Based Path Planning

Industrial vehicles, such as trucks and buses, are considerably affected by their large dimensions. These characteristics seek to optimize fuel consumption and driver staff cost, but come at the expense of hardening the driving task, as it is difficult to maneuver the vehicle in tight roads or spaces. For instance, mining trucks are required to maneuver in highly constrained environments, which can dynamically change with the vehicle operation, and buses are supposed to drive in narrow urban roads while performing sharp turns. The emergence of autonomous vehicles, particularly in the environments where these type of vehicles operate, requires autonomous driving technology that can effectively replace professional drivers.

In this chapter, we propose an optimization-based motion planning framework. Although motion planning algorithms are subject of extensive research, highly constrained environments still prove to be challenging. Common on-road planners generate, in a first step, candidate trajectories to different goal poses ignoring the presence of obstacles and then select the best one. However, this can lead to situations where none of these trajectories are feasible. Furthermore, graph-search methods (e.g., A\*) can result in paths that are not smooth, in the sense that they contain unnatural swerves, which are hard to follow accurately and introduce passenger discomfort (Fassbender et al., 2016). Therefore, it is of utmost importance to design algorithms that are computationally efficient, even when considering the vehicle dimensions.

The outline of this chapter is as follows. In Section 5.1, we recall the notion of space-based road-aligned vehicle model, derive the vehicle dimensions constraints, and propose a novel optimization-based path planning framework. In Section 5.2, we provide simulation results showing the effectiveness of the proposed framework, both in unstructured and structured areas. Finally, in Section 5.3, we summarize the chapter and provide some concluding remarks.

## 5.1 Problem Formulation

The problem addressed in this chapter is illustrated in Figure 5.1. We seek the generation, at each time instant  $t$ , of a smooth space-based discrete path defined by

$$\mathcal{T}_t(s) = [e_{y,t}(s) \quad e_{\psi,t}(s)]^T, \quad (5.1)$$

with  $s \in [s_t, s_t + S]$ , where the reference path length is  $S > 0$  and  $s_t$  denotes the vehicle position along a given reference path at time  $t$ . The path (5.1) describes the lateral displacement  $e_y(s)$  and heading displacement  $e_\psi(s)$  between the vehicle and a reference path as a function of space  $s$ . The reference path can be, for example, the road centerline, the middle of a driving corridor, or an obstacle-avoiding PWA path as proposed by Graf Plessen et al. (2017). Furthermore, let the associated path in the global frame be defined by

$$\mathcal{X}_t(s) = [x_t(s) \quad y_t(s) \quad \psi_t(s)]^T. \quad (5.2)$$

The actual path length traveled by the vehicle is denoted by  $\eta(s)$ . It holds that  $\eta(s) = s$  if  $e_\psi(s) = 0$  and  $e_y(s) = 0$ ,  $\forall s$ . Otherwise,  $\eta(s) \neq s$  due to the vehicle lateral displacement from the road centerline.

### 5.1.1 Vehicle Model

Consider the space-based road-aligned model (3.8) presented in Section 3.2, which describes the lateral displacement  $e_y$  and heading displacement  $e_\psi$  between the vehicle and a given reference path. Let the state vector be  $z = [e_y \quad e_\psi]^T$ , the control input be  $u = \kappa$  and (3.8) be described by  $z' = f(z, u)$ . The reference path is discretized such that  $\{s_j\}_{j=0}^N = \{s_0, s_1, \dots, s_N\}$ , where  $N + 1$  is the number of discretization points. In the presence of obstacles, new grid points are added such that all obstacle corners within the road-aligned frame are accounted for. Consequently, the grid is, in general, non-uniformly spaced. We linearize and discretize the vehicle model around the reference path  $\{e_{y,j}^{\text{ref}}\}_{j=0}^N = 0$ ,  $\{e_{\psi,j}^{\text{ref}}\}_{j=0}^N = 0$ , and  $\{\kappa_j^{\text{ref}}\}_{j=0}^N = \{\frac{1}{\rho_{s,j}}\}_{j=0}^N$  using a first-order Taylor approximation and then apply exact discretization using zero-order hold, as explained in Section 3.2. The linearized and discretized system dynamics are given by  $z(k+1) = A_{d,k}z(k) + B_{d,k}u(k) + h_{d,k}$ .

Let  $c$  be the vehicle curvature rate, with spatial relation with the curvature given by

$$\dot{\kappa} = c \Leftrightarrow \kappa' = \frac{\dot{\kappa}}{\dot{s}} = \frac{\rho_s - e_y}{\rho_s v \cos(e_\psi)} c. \quad (5.3)$$

The curvature  $\kappa$  is directly related to the vehicle steering angle  $\delta$  by  $\kappa = \frac{\tan(\delta)}{L}$ . The input is limited by

$$\kappa_{\min} \leq \kappa(s) \leq \kappa_{\max}, \quad (5.4)$$

$$c_{\min} \leq c(s) \leq c_{\max}. \quad (5.5)$$

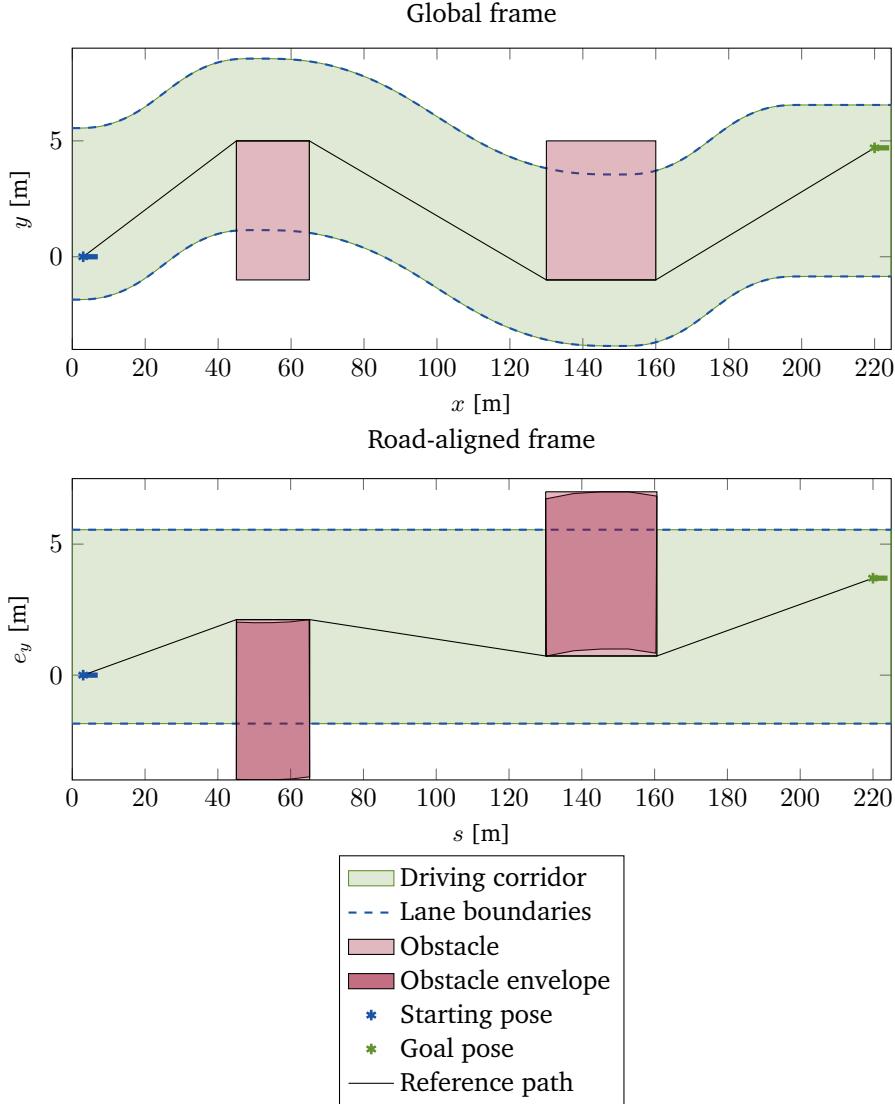


Figure 5.1: Problem illustration. The original problem, in the global frame (above), is first transformed to the road-aligned frame (below). Given a start pose (blue), a path is sought avoiding any obstacles (red), accounting for vehicle dimensions, traveling within corridor bounds, respecting physical actuator constraints, and encouraging smooth trajectories such that an end pose (green) is reached. The transformation from the global to the road-aligned frame originates distorted obstacles that are approximated by their minimal rectangle-envelope within the road-aligned frame. The reference path (black) connects the start to the end pose traveling through obstacle corners in a PWA fashion (Graf Plessen et al., 2017). This path is infeasible to track by a nonholonomic vehicle and, thus, requires smoothing using a path planner before being sent to a motion controller.

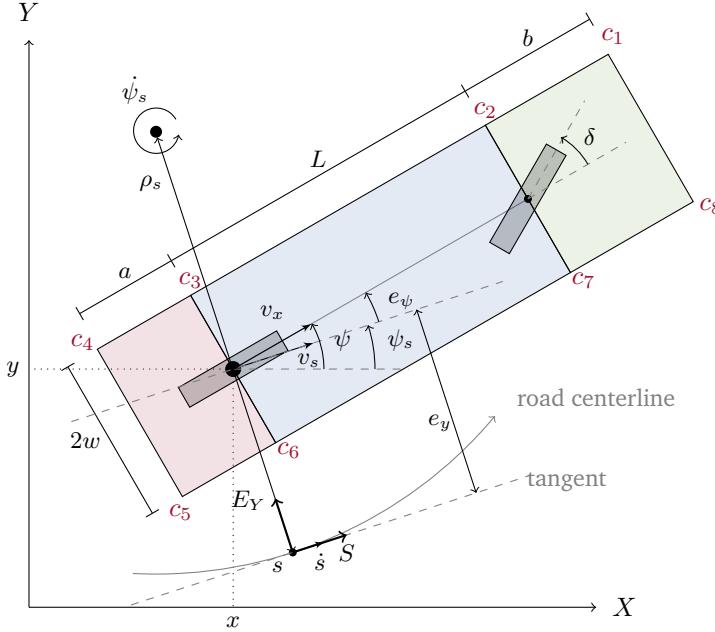


Figure 5.2: Illustration of a bicycle model including the representation of the vehicle frame and respective dimensions. The vehicle model is separated in three different parts, namely, rear overhang ( $a$ ), wheelbase ( $L$ ), and front overhang ( $b$ ).

In this chapter, the vehicle dimensions are modeled. The relation between the global frame and the road-aligned frame is illustrated in Figure 5.2, where the parameters  $a$ ,  $b$ ,  $L$ , and  $w$  indicate front and rear overhang length, vehicle wheelbase, and vehicle half-width, respectively. The overhang is defined as the length of a vehicle, in both the front and rear directions, which exceeds its wheelbase, and it can reach several meters in the case of buses.

### 5.1.2 Driving corridor

Expressing states and control variables as a function of the reference path length  $s$  is beneficial when formulating linearly constrained optimization problems, since it is possible to describe a driving corridor using linear bounds on state variable  $e_y$ . These bounds are derived from spatially dependent road widths and coordinate transformation-distorted obstacles that can be approximated by their minimal rectangle-envelope within the road-aligned frame.

We assume that a driving corridor is given and it consists of spatially dependent state bounds

$$e_{y,\min}(s) \leq e_y(s) \leq e_{y,\max}(s), \quad (5.6)$$

$$e_{\psi,\min}(s) \leq e_{\psi}(s) \leq e_{\psi,\max}(s). \quad (5.7)$$

In the presence of obstacles (dynamic or static), a driving corridor can be determined based on a preprocessing step using heuristics (e.g., overtaking left or right). Dynamic obstacles can be accounted using a velocity-adjusted mapping to the road-aligned coordinate system according to Graf Plessen et al. (2017). Moreover, we assume forward motion, where  $-\frac{\pi}{2} < e_{\psi,\min}(s) \leq 0$  and  $0 \leq e_{\psi,\max}(s) < \frac{\pi}{2}$ .

### 5.1.3 Vehicle Dimensions

When the vehicle is short and the road has low curvature (e.g., passenger vehicles driving in highway scenarios), the vehicle can be approximated by a rectangle in the road-aligned frame. However, for large vehicles driving in large curvature roads (e.g., buses driving in urban environments), that approximation loses its validity.

In the following, we derive the vehicle dimension constraints for the vehicle wheelbase edges. The derivation of the vehicle edges for the other parts (i.e., the rear or the front overhang), when needed, is done analogously. The derivation of vehicle dimension constraints allows imposing different constraints for different vehicle chassis parts. Bus drivers often allow the vehicle overhangs to sweep outside of the road limits and over curbs in order to more easily maneuver the vehicle. For example, one can enforce the vehicle wheelbase to be inside the driving corridor at all times while allowing for the overhangs to exit it.

According to Figure 5.3, at every  $s_j$ , we define the set of  $s$ -coordinates that includes the  $s$ -coordinates of the vehicle corners in between any two grid points. Thus, we compute the  $s$ -coordinates of the vehicle corners and approximate them conservatively. Formally, we define the set  $\tilde{\mathcal{S}}_j$ , which contains all the  $s$ -coordinates between the outer-most vehicle corners, as

$$\begin{aligned} \tilde{\mathcal{S}}_j &= \{s \in \{s_k\}_{k=0}^N : \Delta s_{j,\min} \leq s \leq \Delta s_{j,\max}\} \\ &= \{s_{\tilde{k}_1}, s_{\tilde{k}_2}, \dots, s_{\tilde{k}_{\tilde{M}}}\}, \end{aligned} \quad (5.8)$$

with  $\tilde{M}$  being the number of points in the set  $\tilde{\mathcal{S}}_j$  and, for the wheelbase,

$$\begin{aligned} \Delta s_{j,\min} &= [\min(s_{j,c_3}, s_{j,c_6})], \\ \Delta s_{j,\max} &= [\max(s_{j,c_2}, s_{j,c_7})], \end{aligned}$$

where  $s_{j,c_i}$  for  $i = 2, 3, 6, 7$  represent the four vehicle wheelbase corners when the rear-axle center is at position  $s_j$ . The operators  $[\cdot]$  and  $\lfloor \cdot \rfloor$  represent the ceiling and floor function, respectively. The  $s$ -coordinate of the corners  $c_i$ ,  $i = 1, \dots, 8$  in Figure 5.2, can be expressed as

$$s_{c_i} = s + \xi_{c_i} \cos(e_{\psi}) + \zeta_{c_i} \sin(e_{\psi}), \quad (5.9)$$

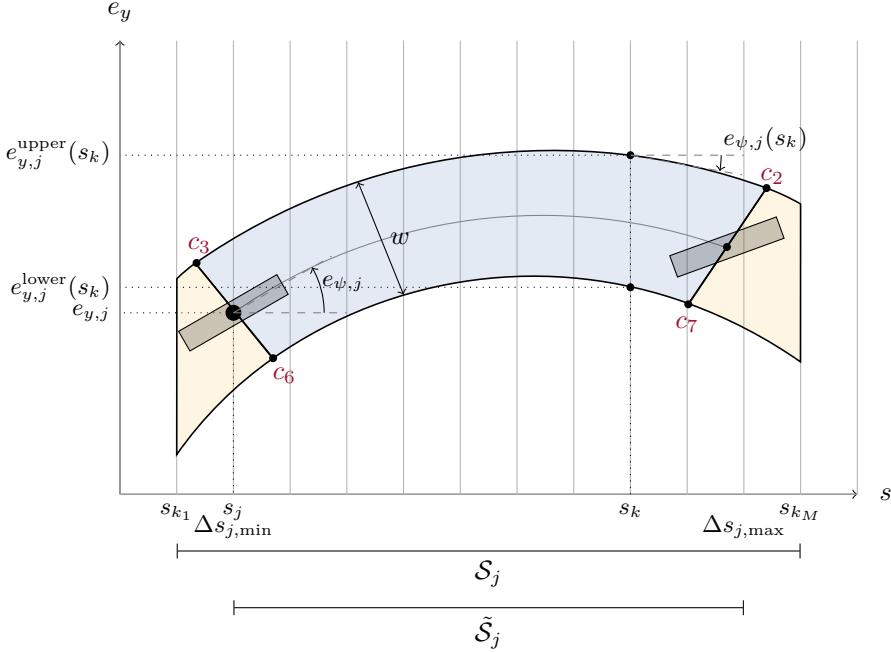


Figure 5.3: Illustration of the distorted vehicle wheelbase body (blue) and the conservative envelope (yellow).

with

$$\begin{aligned}\xi_{c_i} &= [L + b, L, 0, -a, -a, 0, L, L + b]^T, \\ \zeta_{c_i} &= [-w, -w, -w, -w, w, w, w, w]^T.\end{aligned}\quad (5.10)$$

Afterwards, we take the conservative approximation of the vehicle envelope by prepending and appending the closest  $s$ -coordinates, defining the new set  $\mathcal{S}_j$  as

$$\mathcal{S}_j = \{s_{\tilde{k}_1-1}, s_{\tilde{k}_1}, \dots, s_{\tilde{k}_{\tilde{M}}}, s_{\tilde{k}_{\tilde{M}}+1}\} = \{s_{k_1}, \dots, s_{k_M}\}, \quad (5.11)$$

where  $M \geq \tilde{M}$  is the number of points in the set  $\mathcal{S}_j$ .

Using trigonometric relationships according to Figure 5.3, the equations describing the vehicle lateral boundaries are affine in  $s_j$  and nonlinear in  $e_{\psi,j}$  as

$$\begin{aligned}e_{y,j}^{\text{lower}}(s_k) &= e_{y,j} + \tan(e_{\psi,j}(s_k))(s_k - s_j) - \frac{w}{\cos(e_{\psi,j}(s_k))}, \\ e_{y,j}^{\text{upper}}(s_k) &= e_{y,j} + \tan(e_{\psi,j}(s_k))(s_k - s_j) + \frac{w}{\cos(e_{\psi,j}(s_k))},\end{aligned}\quad (5.12)$$

where  $s_k \in \mathcal{S}_j$ . The inclusion of the  $s_k$  dependency in the term  $e_{\psi,j}(s_k)$  approximates the deformation caused by the transformation from global to road-

aligned frame. In cases where the distortion can be neglected, we assume that  $e_{\psi,j}(s_k) = e_{\psi,j}$ ,  $\forall s_k \in \mathcal{S}_j$ .

We are interested in modeling the vehicle edges through linear equations to solve an optimization problem via convex optimization techniques. Thus, we compute the first-order Taylor expansion of (5.12), yielding

$$\begin{aligned} e_{y,\text{lin},j}^{\text{lower}}(s_k) &= [1 \ g^{\text{lower}}(s_k)] z_j + h_{\text{lin},j}^{\text{lower}}(s_k), \\ e_{y,\text{lin},j}^{\text{upper}}(s_k) &= [1 \ g^{\text{upper}}(s_k)] z_j + h_{\text{lin},j}^{\text{upper}}(s_k), \end{aligned} \quad (5.13)$$

with  $g^{\text{lower}}(s_k)$ ,  $h_{\text{lin},j}^{\text{lower}}(s_k)$ ,  $g^{\text{upper}}(s_k)$ , and  $h_{\text{lin},j}^{\text{upper}}(s_k)$  parametrized by  $s_j$ ,  $e_{\psi,j}^{\text{ref}}$ , and  $e_{y,j}^{\text{ref}}$ , yielding

$$g^{\text{lower}}(s_k) = \sec^2(e_{\psi,j}^{\text{ref}}(s_k))(s_k - s_j) - w \sec(e_{\psi,j}^{\text{ref}}(s_k)) \tan(e_{\psi,j}^{\text{ref}}(s_k)), \quad (5.14)$$

$$h_{\text{lin},j}^{\text{lower}}(s_k) = \tan(e_{\psi,j}^{\text{ref}}(s_k))(s_k - s_j) - \frac{w}{\cos(e_{\psi,j}^{\text{ref}}(s_k))} - e_{\psi,j}^{\text{ref}}(s_k)g^{\text{lower}}(s_k), \quad (5.15)$$

$$g^{\text{upper}}(s_k) = \sec^2(e_{\psi,j}^{\text{ref}}(s_k))(s_k - s_j) + w \sec(e_{\psi,j}^{\text{ref}}(s_k)) \tan(e_{\psi,j}^{\text{ref}}(s_k)), \quad (5.16)$$

$$h_{\text{lin},j}^{\text{upper}}(s_k) = \tan(e_{\psi,j}^{\text{ref}}(s_k))(s_k - s_j) + \frac{w}{\cos(e_{\psi,j}^{\text{ref}}(s_k))} - e_{\psi,j}^{\text{ref}}(s_k)g^{\text{upper}}(s_k). \quad (5.17)$$

Expressions (5.14) to (5.17) evaluated at the discrete grid points (5.11) can be expressed as the set of inequalities

$$\begin{aligned} \begin{bmatrix} e_{y,\text{lin},j}^{\text{lower}}(s_{k_1}) \\ \vdots \\ e_{y,\text{lin},j}^{\text{lower}}(s_{k_M}) \end{bmatrix} &\geq \begin{bmatrix} e_{y,\text{min}}(s_{k_1}) \\ \vdots \\ e_{y,\text{min}}(s_{k_M}) \end{bmatrix}, \\ \begin{bmatrix} e_{y,\text{lin},j}^{\text{upper}}(s_{k_1}) \\ \vdots \\ e_{y,\text{lin},j}^{\text{upper}}(s_{k_M}) \end{bmatrix} &\leq \begin{bmatrix} e_{y,\text{max}}(s_{k_1}) \\ \vdots \\ e_{y,\text{max}}(s_{k_M}) \end{bmatrix}. \end{aligned} \quad (5.18)$$

Inequalities (5.18) are linear in state  $z_j$  and can be compactly summarized as

$$Q_j^{\text{lower}} z_j \geq q_j^{\text{lower}} \text{ and } Q_j^{\text{upper}} z_j \leq q_j^{\text{upper}},$$

with  $Q_j^{\text{lower}}$ ,  $Q_j^{\text{upper}} \in \mathbb{R}^{M \times 2}$ ,  $q_j^{\text{lower}}$ , and  $q_j^{\text{upper}} \in \mathbb{R}^M$  dependent on references  $e_{\psi,j}^{\text{ref}}$  and  $e_{y,j}^{\text{ref}}$ .

We have a vector of inequalities (5.18) for each of the vehicle parts. Thus, we construct

$$\begin{aligned} \bar{Q}_j^{\text{lower}} &= [Q_{j,\text{roh}}^{\text{lower}} \ Q_{j,\text{wb}}^{\text{lower}} \ Q_{j,\text{foh}}^{\text{lower}}]^T, \\ \bar{Q}_j^{\text{upper}} &= [Q_{j,\text{roh}}^{\text{upper}} \ Q_{j,\text{wb}}^{\text{upper}} \ Q_{j,\text{foh}}^{\text{upper}}]^T, \\ \bar{q}_j^{\text{lower}} &= [q_{j,\text{roh}}^{\text{lower}} \ q_{j,\text{wb}}^{\text{lower}} \ q_{j,\text{foh}}^{\text{lower}}]^T, \\ \bar{q}_j^{\text{upper}} &= [q_{j,\text{roh}}^{\text{upper}} \ q_{j,\text{wb}}^{\text{upper}} \ q_{j,\text{foh}}^{\text{upper}}]^T, \end{aligned} \quad (5.19)$$

where the subscripts roh, wb, and foh stand for rear overhang, wheelbase, and front overhang, respectively.

### 5.1.4 Objective function

The objective function has two distinct terms. The first term,  $J_{\text{smooth}}(\bar{\kappa})$ , where  $\bar{\kappa} = \{\kappa_j\}_{j=0}^{N-1}$ , is a convex function (e.g.,  $\max |\bar{\kappa}|$ ) and used to enforce smoothness of the planned path. The second term,  $J_{\text{slack}}(\bar{\sigma}_{e_y}, \bar{\lambda})$ , is another convex function and accounts for the vehicle dimension constraints violations (e.g., lane boundaries or obstacles). The parameter  $\bar{\lambda}$  is used to penalize vehicle dimension constraints violations.

As described previously, it is possible to model the vehicle in three different parts (i.e., rear and front overhang and wheelbase). When dealing with vehicles with these characteristics, we choose  $\bar{\lambda} = \text{diag}(\lambda_{\text{foh}}, \lambda_{\text{wb}}, \lambda_{\text{roh}})$ , to penalize the front overhang, between wheels, and rear overhang outside the assigned driving lane, separately. For example, we can choose  $\lambda_{\text{wb}} \gg \lambda_{\text{foh}} = \lambda_{\text{roh}}$  if we want to prevent the wheels from exiting the lane boundaries, but allow the overhang(s) to surpass the lane boundary. Otherwise, if the overhangs are neglectable it is possible to choose  $\lambda = \lambda_{\text{wb}} \gg 0$  to enforce constraint satisfaction at all times. The introduction of a slack variable vector  $\bar{\sigma}_{e_y} = [\sigma_{e_y^{\text{foh}}}, \sigma_{e_y^{\text{wb}}}, \sigma_{e_y^{\text{roh}}}]^T$ , when there are overhangs, and  $\sigma_{e_y} = \sigma_{e_y^{\text{wb}}}$ , when the overhangs are neglectable, ensures feasibility of the optimization problem.

### 5.1.5 Convex optimization problem formulation

We formulate the path optimization problem as a convex optimization based on the linearized vehicle model and vehicle dimension constraints as

$$\min_{\bar{\kappa}, \bar{\sigma}_{e_y}} J_{\text{smooth}}(\bar{\kappa}) + J_{\text{slack}}(\bar{\sigma}_{e_y}, \bar{\lambda}) \quad (5.20a)$$

$$\text{s.t. } z_{j+1} = A_{d,j}z_j + B_{d,j}u_j + h_{d,j}, \quad j = 0, \dots, N-1, \quad (5.20b)$$

$$e_{y,0} = e_{y,0}^{\text{ref}}, \quad e_{\psi,0} = e_{\psi,0}^{\text{ref}}, \quad (5.20c)$$

$$e_{y,N} = e_{y,N}^{\text{ref}}, \quad e_{\psi,N} = e_{\psi,N}^{\text{ref}}, \quad (5.20d)$$

$$\kappa_{\min} \leq \kappa_j \leq \kappa_{\max}, \quad j = 0, \dots, N-1, \quad (5.20e)$$

$$c_{\min} \leq c_j \leq c_{\max}, \quad j = 0, \dots, N-1, \quad (5.20f)$$

$$\bar{Q}_j^{\text{lower}} z_j \geq \bar{q}_j^{\text{lower}} - \bar{\sigma}_{e_y}, \quad j = 1, \dots, N, \quad (5.20g)$$

$$\bar{Q}_j^{\text{upper}} z_j \leq \bar{q}_j^{\text{upper}} + \bar{\sigma}_{e_y}, \quad j = 1, \dots, N, \quad (5.20h)$$

$$\bar{\sigma}_{e_y} \geq 0, \quad (5.20i)$$

with decision variables  $\bar{\kappa}$  and  $\bar{\sigma}_{e_y}$ . Note that  $u_j = \kappa_j$ . The initial and final state are denoted by  $z(0) = [e_{y,0}^{\text{ref}} \ e_{\psi,0}^{\text{ref}}]^T$  and  $z(N) = [e_{y,N}^{\text{ref}} \ e_{\psi,N}^{\text{ref}}]^T$ , respectively. The equal-

ity constraint (5.20b) defines the vehicle model dynamics, which are described in Section 5.1.1. The vehicle is enforced to stay inside the driving corridor, as explained in Section 5.1.2, in (5.20g) and (5.20h) using linear vehicle dimensions constraints detailed in Section 5.1.3. The objective function (5.20a) is discussed in Section 5.1.4.

**Remark.** *The resulting obstacle avoiding path may significantly deviate from the reference path or road centerline. As defined in Section 5.1, the path arc-length along the actual planned path is  $\eta$ . For example, a PWA curvature profile  $\kappa(\eta)$  would ensure clothoid-based path planning. However, a PWA  $\kappa(s)$  does not yield a clothoid path, unless  $s = \eta$ .*

### 5.1.6 Sequential Linear/Quadratic Programming

Successful obstacle avoidance maneuvers that require significant deviations from the reference path and large input commands are strongly dependent on underlying reference paths used for linearization and discretization of the vehicle model and dimensions constraints. This is of particular relevance for the first references initialization, if we use a least-heading-varying PWA path avoiding all obstacles as a reference path (Graf Plessen et al., 2017).

Choosing appropriately the cost function, problem (5.20) can be cast as an LP or QP (see Section 2.2.3). Solving (5.20) sequentially, using the solution of the previous iteration as the reference path for linearization and discretization of the vehicle model and constraints, originates an SLP or SQP.

We summarize the algorithm as follows, where  $I_{\max}$  is the maximum number of iterations:

1. Initialize:  $\{e_{y,j}^{\text{ref}}\}_{j=0}^N$ ,  $\{e_{\psi,j}^{\text{ref}}\}_{j=0}^N$ , and  $\{u_j^{\text{ref}}\}_{j=0}^{N-1}$ ;
2. For  $i = 1, \dots, I_{\max}$ :
  - a) Solve (5.20);
  - b) Update  $\{e_{y,j}^{\text{ref}}\}_{j=0}^N$ ,  $\{e_{\psi,j}^{\text{ref}}\}_{j=0}^N$ , and  $\{u_j^{\text{ref}}\}_{j=0}^{N-1}$  using solution of the iteration  $i - 1$ ;
  - c) Check termination criterion.
3. Output:  $\{e_{y,j}^{\text{ref}}\}_{j=0}^N$ ,  $\{e_{\psi,j}^{\text{ref}}\}_{j=0}^N$ , and  $\{u_j^{\text{ref}}\}_{j=0}^{N-1}$ .

The termination criterion is as follows. In every iteration  $i$ , the nonlinear vehicle dimensions constraints (5.12) are evaluated along  $\{s_j\}_{j=0}^N$ . If any obstacle or road boundary is hit an additional iteration is conducted. Moreover, two consecutive solutions are compared and if they differ significantly (i.e., if the difference is larger than a certain small threshold) an additional iteration is conducted.

## 5.2 Simulation Results

In this section, we discuss the performance of the proposed path planning optimization framework described in Section 5.1. Three different scenarios are presented to demonstrate the flexibility of the proposed approach. First, we consider a *tight maneuvering scenario*, such as a parking lot. Second, we consider a *low curvature road scenario* corresponding to a Low curvature road profile with large inter-obstacle distance. In these two scenarios, the vehicle distortion is ignored and the vehicle overhangs are neglected, since we assume the vehicle is short and that the reference path curvature is small. Also, the optimization problem is solved sequentially, as described in Section 5.1.6. For comparison, we consider a semi-analytical path planning method based on (Funke and Gerdès, 2016), where three path primitives (straight lines, arcs, and clothoids) are concatenated to plan emergency lane changes up to the vehicle friction limits. Finally, we consider an *overhang minimization scenario*, which consists in a road with tight lane boundaries and large curvature. In this scenario, we consider a long vehicle, such as a bus, with prominent overhangs and, consequently, the vehicle distortion in the road-aligned model is considered. The performance evaluation is based on the amount of vehicle overhang exiting the assigned driving lane. To be able to provide performance bounds, we derive geometrically the minimum road width necessary depending on the vehicle geometry and road curvature. In the following, the path planning framework presented in this chapter is abbreviated as OPP (Optimization-based Path Planning).

For prototyping the optimization problem, we use the MATLAB toolbox CVX (Grant and Boyd, 2014) with Gurobi (Gurobi Optimization, Inc., 2016) as a solver. All simulations are conducted on a desktop PC running Windows 7 equipped with an Intel Xeon CPU E5-2670 v2 @2.50GHz, 16.0GB of memory, and using MATLAB R2016a.

### 5.2.1 Concatenating Clothoids for Path Planning

Funke and Gerdès (2016) propose a semi-analytical path planning method based on concatenating straight lines, arcs, and clothoids to plan emergency lane changes up to the vehicle friction limits. According to González et al. (2016), interpolation-based path planning methods are among the most applied in real implementations by research groups worldwide. Clothoids, together with Bézier and polynomial curves, belong to that class. Safety margins are added to obstacle contours to account for vehicle dimensions. Here, we treat the corners of safety margin-adjusted obstacles as waypoints. Planning based on path primitives requires selecting multiple parameters, such as arc length, straight length, and symmetric point fractions. These selections can significantly affect the end result. Clothoids can be fitted in either the global or road-aligned frame. In the following, the comparative method is abbreviated as CPP (Clothoid Path Planning).

**Remark.** To maintain simplicity, we focus on clothoids, straight lines, and symmetric trajectory design when performing lane changes. We do not consider “early” or “late” steering, as discussed by Funke and Gerdes (2016).

### 5.2.2 Tight Maneuvering Scenario

Consider a planning scenario in a parking lot. In this scenario, the objective function is defined as  $J_{\text{smooth}}(\bar{\kappa}) = \max |\bar{\kappa}| + \max |D_1 \bar{\kappa}|$ . Matrix  $D_1 \in \mathbb{Z}^{(N-2) \times (N-1)}$  is a matrix operator that calculates first-order differences of a vector. The cost function minimizes the maximum absolute curvature and curvature rate along the reference path.

The vehicle dimensions and input constraints used are based on a standard passenger car. The dimensions are  $L = 4.3$  meters and  $w = 0.9$  meters. The input constraints are  $\kappa_{\max} = -\kappa_{\min} = 0.195 \text{ m}^{-1}$  and  $c_{\max} = -c_{\min} = 0.06 \text{ m}^{-1}/\text{s}$ .

Figure 5.5 depicts the outcome of OPP and CPP for this scenario. First, the CPP does not account explicitly for actuator constraints, which results in actuator constraints violation, and it connects waypoints pairwise, which enforces parallel obstacle avoidance (i.e., obstacle avoidance with the same heading as its edges). In contrast, OPP remains within actuation limits at all times. This is enabled by the fact that the OPP accounts for the entire obstacle constellation space resulting in anticipative steering. Second, note that the final steering angle, upon reaching the end pose is turned at  $-22.1^\circ$ . This is in line with the formulation of (5.20) connecting the given start and end pose (and not planning beyond these). Finally, four iterations were required to meet the termination criterion introduced in Section 5.1.6.

Note that the linearization of the vehicle model yields a pole at  $e_{\psi,j}^{\text{ref}} = \pm \frac{\pi}{2}$ ,  $\forall j$ . Accordingly, forward motion along positive  $s$  is achieved only for  $-\frac{\pi}{2} < e_{\psi,j} < \frac{\pi}{2}$ ,  $\forall j$ . As Figure 5.5 illustrates, the obstacle avoiding path is almost vertical with respect to the reference path. Despite such large deviation from the road centerline, the SQP planner is able to converge in only four iterations, which indicates good robustness of the method.

### 5.2.3 Low Curvature Road Scenario

Consider a planning scenario in a highway. In this scenario, the objective function is again defined as  $J_{\text{smooth}}(\bar{\kappa}) = \max |\bar{\kappa}| + \max |D_1 \bar{\kappa}|$  and the reference path, which is the road centerline, is discretized using  $N = 200$  points.

For a fair comparison between OPP and CPP, we add conservative safety margins to the obstacles and dismiss constraints (5.20g) and (5.20h) from (5.20). However, to ensure that the vehicle remains within the driving corridor equivalent constraints are added, but on the center of the rear axle

$$e_{y,j,\min} - \sigma_{e_y} \leq e_{y,j} \leq e_{y,j,\max} + \sigma_{e_y}, \quad j = 1, \dots, N, \quad (5.21)$$

$$e_{\psi,j,\min} - \sigma_{e_\psi} \leq e_{\psi,j} \leq e_{\psi,j,\max} + \sigma_{e_\psi}, \quad j = 1, \dots, N, \quad (5.22)$$

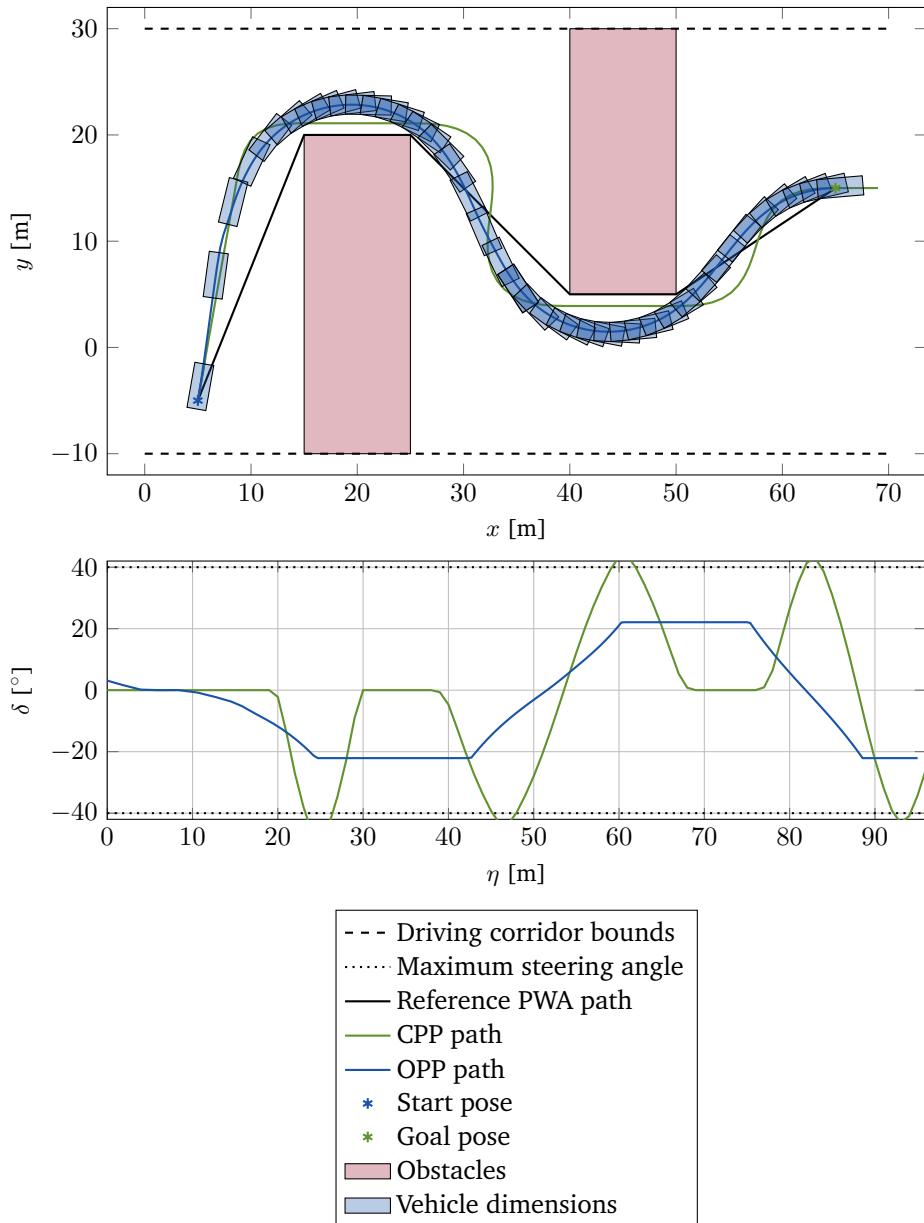


Figure 5.4: *Tight maneuvering scenario.* Resulting vehicle paths in the global frame (upper plot) and control inputs (lower plot). To account for vehicle dimensions, CPP assumed an obstacle inflated by a safety margin. For OPP, the upper plot visualizes vehicle dimensions (displayed every 5<sup>th</sup> sampling) and it can be seen that obstacles are avoided tightly.

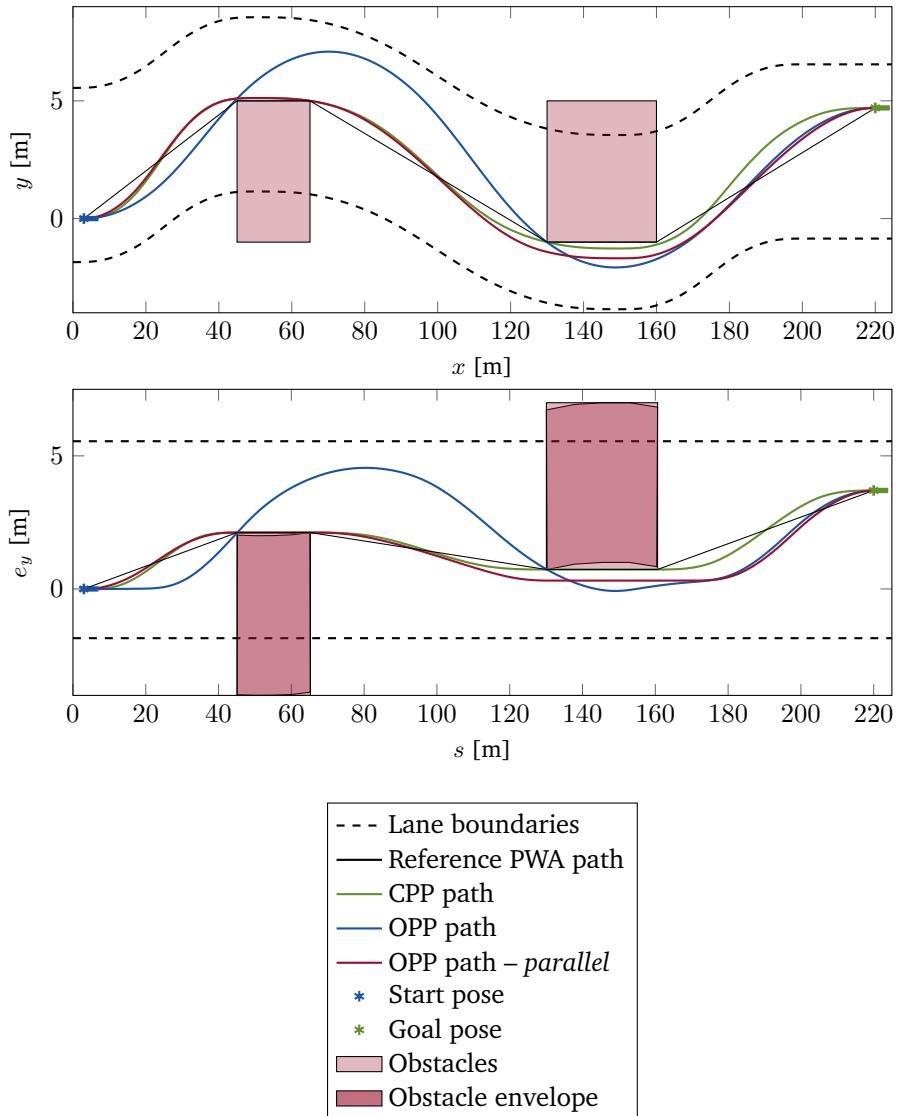


Figure 5.5: *Low curvature road scenario.* Comparison of the resulting paths in the global (upper plot) and in the road-aligned frame (lower plot), in which the path planning is performed. The obstacles are represented already inflated by a safety margin and the light gray lines (lower plot) indicate conservative road bounds to account for vehicle dimensions.

where  $\sigma_{e_y}$  and  $\sigma_{e_\psi}$  are slack variables.

A benefit of the proposed optimization-based path planning framework is that additional constraints can easily be added. To enforce the overtaking of  $P$  obstacles in parallel, we may add

$$e_{\psi,p,\text{obs}} - \sigma_{e_\psi} \leq e_{\psi,j} \leq e_{\psi,p,\text{obs}} + \sigma_{e_\psi}, \quad \forall j \in \mathcal{J}_l^{\text{obs}}, \quad (5.23)$$

for  $p = 1, \dots, P$ , and where

$$\mathcal{J}_l^{\text{obs}} = \{j : s_{p,\text{obs,start}} \leq s_j \leq s_{p,\text{obs,end}}, j = 1, \dots, N\},$$

where  $e_{\psi,p,\text{obs}}$  denotes the heading of the rectangle-envelope of obstacle  $p$ , located between  $s_{p,\text{obs,start}}$  and  $s_{p,\text{obs,end}}$ .

Figure 5.5 depicts the outcome of OPP and CPP for this scenario. First, OPP computes a path that uses the entire space within the driving corridor reaching road boundary saturation and a non-parallel overtaking of obstacles. Second, when including constraints (5.23) in (5.20), OPP overtakes the obstacles in parallel as desired and produces a path that is further laterally displaced from the second obstacle in comparison to CPP, which is enforced to proceed along fixed waypoints placed at the edges of the inflated obstacle. Third, note the difference between planned paths in the global and road-aligned frame. While obstacles are overtaken in parallel in the road-aligned frame for OPP with constraints (5.23) and CPP, the resulting global-frame paths avoid obstacles in a curved fashion after retransformation. Finally, two iterations were required to meet the termination criterion introduced in Section 5.1.6. This can be explained by the fact that the initial PWA reference path already served as a sufficiently good reference.

### 5.2.4 Overhang Minimization Scenario

In this scenario, we evaluate the effectiveness of the optimization-based path planning framework, described in Section 5.1, by comparing the amount of overhang exiting the assigned driving lane, when tracking the road centerline and the optimized path. To be able to provide performance bounds, we derive an estimate of the minimum necessary road width (i.e., such that the wheelbase is always inside the road limits) depending on the vehicle geometry and road curvature.

#### Minimum Road Width Estimate

Figure 5.6 depicts the geometry of the problem of computing the minimum road width depending on the road curvature  $\kappa_s = \rho_s^{-1}$  and the vehicle dimensions. Using this information, we can derive the minimum road width necessary for the vehicle to travel with all its wheels inside the lane boundaries. Moreover, we can also derive an expression for an estimate of the amount of vehicle overhang outside the

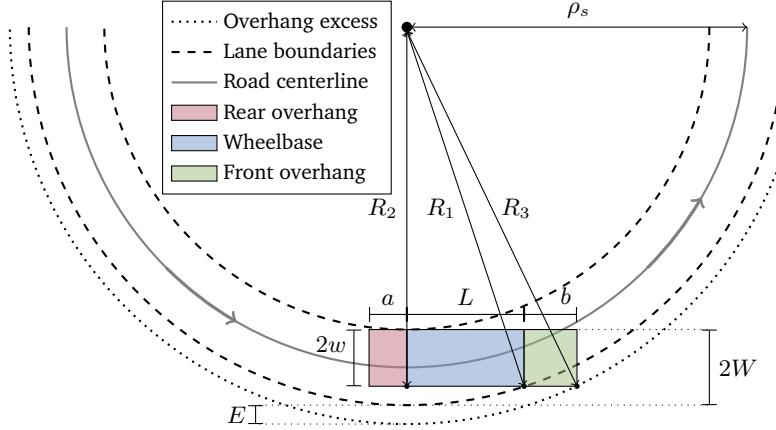


Figure 5.6: Geometric relations used in the computation of the minimum road width.

assigned driving lane. From Figure 5.6, we get

$$R_1 = \sqrt{R_2^2 + L^2}, \quad (5.24a)$$

$$R_3 = \sqrt{R_2^2 + (L + b)^2}. \quad (5.24b)$$

Furthermore,

$$R_1 = \rho_s + W, \quad (5.25a)$$

$$R_2 = \rho_s - W + 2w. \quad (5.25b)$$

From the above relations, we determine the minimum road width as

$$W = \frac{L^2 + 4w^2 + 4\rho_s w}{4(\rho_s + w)}, \quad (5.26)$$

and the estimation of the amount of vehicle overhang outside the assigned driving lane

$$E = R_3 - R_1. \quad (5.27)$$

Note that this estimate considers that the vehicle is traveling tangentially to the road inner-lane. Note that this estimate is rather conservative, as the vehicle could have a different heading angle allowing for a smaller road width. As we demonstrate in the next subsection, the vehicle does not need to travel tangentially to the road inner-lane to minimize the amount of overhangs outside the assigned driving lane.



Figure 5.7: Scania bus used as experimental and research platform (courtesy of Scania CV AB).

### Complete Path Optimization

The reference path is recorded at Scania test facilities in Södertälje, Sweden. The discretization interval is set to 0.25 meters. The maximum path curvature is  $0.117 \text{ m}^{-1}$  and we use a lane width of 2.5 meters, slightly larger than the estimated minimum road width (5.26) of 2.19 meters. The vehicle dimensions, illustrated in Figure 5.2, and the input constraints are based on the Scania bus shown in Figure 5.7. The dimensions are  $a = 2.66 \text{ m}$ ,  $b = 3.34 \text{ m}$ ,  $L = 6 \text{ m}$ , and  $w = 1.27 \text{ m}$ . The input constraints are  $-\kappa_{\min} = \kappa_{\max} = 0.18 \text{ m}^{-1}$  and  $-c_{\min} = c_{\max} = 0.03 \text{ m}^{-1}/\text{s}$ .

Figure 5.8 depicts the outcome of (5.20). The first term of the cost function is chosen again as  $J_{\text{smooth}}(\bar{\kappa}) = \max |\bar{\kappa}| + \max |D_1 \bar{\kappa}|$ . Moreover,  $\lambda_{\text{foh}} = \lambda_{\text{roh}} = 1$  and  $\lambda_{\text{wb}} = 10^3$ . It can be seen that the vehicle wheels always lie inside the assigned driving lane. Note that the obtained shape of the optimal path *cuts the corner* to use the assigned driving lane more efficiently.

Figure 5.9 compares the optimal path with the centerline. When tracking the centerline, the largest amount of front overhang outside the assigned driving lane is 2.67 meters, while when tracking the optimal path it is only 1.47 meters. Therefore, our proposed method reduces the amount of overhang outside the assigned driving lane by 45%. From (5.27), the estimated amount of overhang outside the assigned driving lane is 1.66 meters. Note that, as discussed in Section 5.2.4, the optimal path is not tangent (i.e., with the same heading) to the road inner-lane.

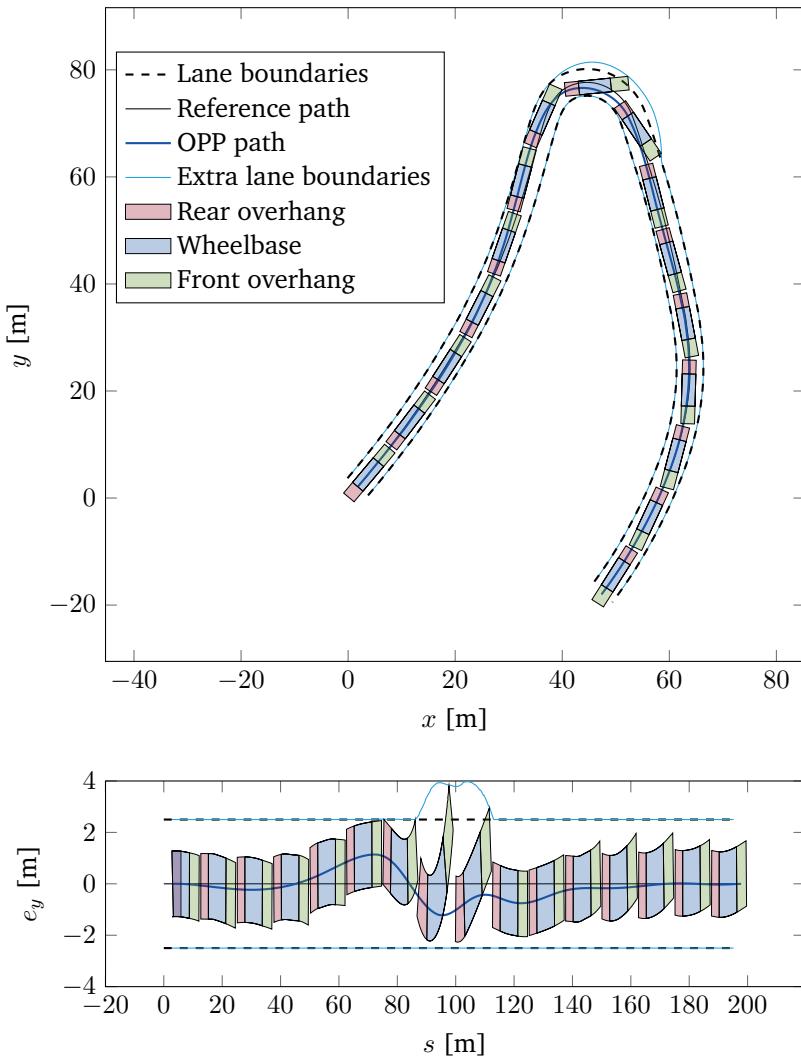


Figure 5.8: *Overhang minimization scenario*. Resulting vehicle path in the global frame (upper plot) after the retransformation from the road-aligned frame (lower plot), in which the optimization problem is solved. The plot depicts vehicle body (displayed every 50<sup>th</sup> sample), where we distinguish the three different vehicle parts (roh, wb, and fooh). According to the problem optimization, the vehicle wheelbase is always inside the lane boundaries, while minimizing the amount of the overhang that exceeds them. Also, the amount of the vehicle overhang outside the lane boundaries is shown.

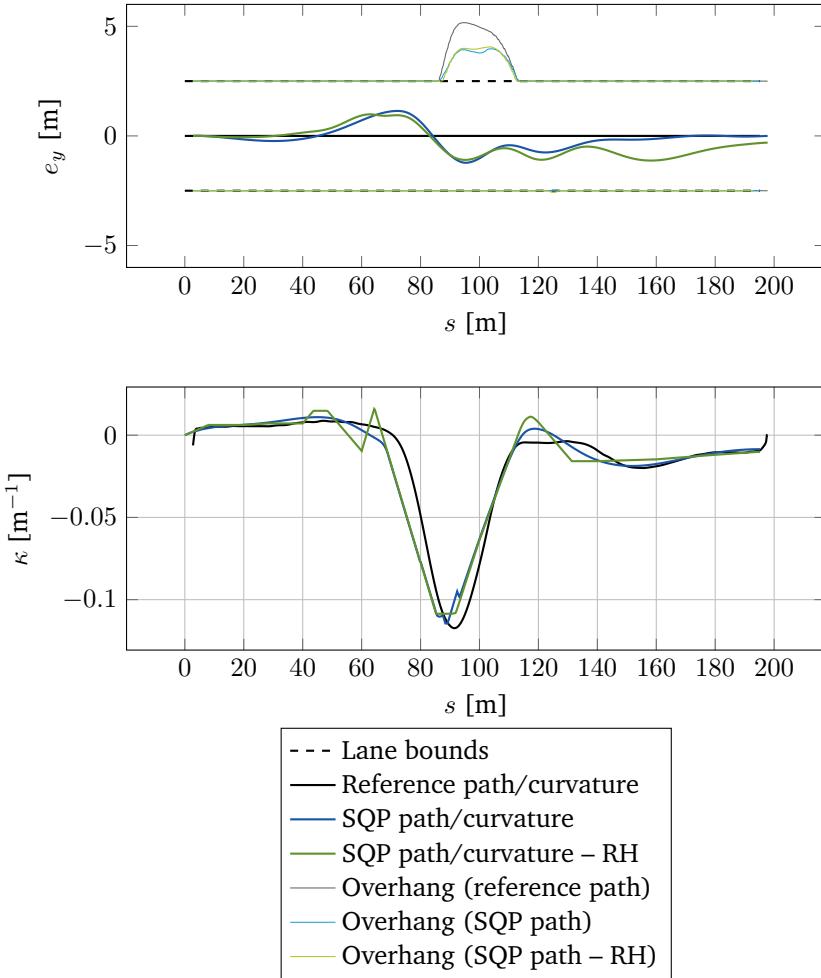


Figure 5.9: *Overhang minimization scenario*. Different paths comparison in the road-aligned frame, together with the respective amount of the vehicle overhang outside the lane boundaries (upper plot). Vehicle curvature comparison (lower plot).

### Receding-Horizon Path Optimization

A drawback with the proposed method is its relatively high computational burden. The scenario presented has 195 meters and 780 sampling points. The optimization solver takes 2.16 seconds to find the solution. A possible alternative is to solve the problem in a Receding-Horizon (RH) fashion. Instead of using the whole reference path, the optimization can be performed over shorter subsets of the path, while moving along it. Moreover, the solutions computed at each step can be used as new references to improve the linearization quality and also reduce the solver execution time in the next step.

In this scenario, an optimal path is computed every 5 meters, using a prediction horizon of 10 meters. Figure 5.9 compares the outcome of (5.20) in a RH fashion against optimizing over the whole reference path. When tracking the RH optimal path, the amount of front overhang that exceeds lane boundaries is 1.56 meters, only 0.09 meters more than when tracking the optimal path. Moreover, the average computation time is 0.30 seconds, with a maximum of 0.55 seconds. These execution times demonstrate the feasibility of applying the method in real-time, since a vehicle would not drive faster than 5 m/s in a road with curvature exceeding  $0.11 \text{ m}^{-1}$  (Lima et al., 2015a). Therefore, at 5 m/s solutions are computed every second, which comfortably fits the solver execution time. Figure 5.9 depicts the input signals computed using the complete path optimization and the RH approach. As expected, due to the inherent suboptimality of the RH approach, the computed curvature is not as smooth as in the complete path optimization case. Note that, for comparison purposes, the cost function used in the RH approach is exactly the same as in the offline case. We can get smoother solutions by tuning the cost function by, for instance, penalizing more non-smooth curvature signals. In any case, it can be seen that the computed optimal path is smooth.

## 5.3 Summary

In this chapter, we presented an optimization-based path planning framework for autonomous vehicles. The vehicle motion is predicted using a linearized and discretized space-based vehicle model in a road-aligned coordinate frame. The main novelty was the incorporation of linearized vehicle dimension constraints within an SLP or SQP algorithm, which improved the path planning performance in highly constrained environments. The method also captured the fact that vehicle constraints in the road-aligned frame are severely deformed due to lengthy vehicles or large curvature road scenarios. In simulation, the effectiveness of the method was shown in three different scenarios: a tight maneuvering scenario, a low curvature road scenario, and an overhang minimization scenario. In the last scenario, we demonstrated the real-time applicability of the framework by using it in a receding-horizon fashion, with almost no loss of performance.



## Chapter 6

# Progress Maximization Model Predictive Controller

Progress maximization (i.e., traveling time minimization) plays an important role not only in racing, but also in efficient and safe autonomous driving applications. Autonomous vehicles are expected to emerge in closed environments, such as gravel pits, mining areas, construction sites, and loading terminals. These production sites are part of a highly competitive market, hence increasing efficiency and productivity by reducing the vehicle traveling time is of utmost importance.

The outline of this chapter is as follows. In Section 6.1, we present the main idea behind the development of a progress maximization MPC and recall the space-based road-aligned vehicle model, to which we add extra states to accurately model a vehicle at high speeds. This model is successively linearized at each time step in order to allow posing a convex optimization problem. Several linear approximations of the vehicle dynamics constraints are introduced, including a novel polytopic approximation of the ‘g-g’ diagram (see Section 2.1.5). In Section 6.2, we illustrate the effectiveness of the proposed controller in simulation, where a non-linear dynamic vehicle is controlled to maximize the progress along a track, taking into consideration possible obstacles. In Section 6.3, we summarize the chapter and provide some concluding remarks.

### 6.1 Problem Formulation

In this section, we present a high-level design of a motion controller that maximizes on-road progress along a given path under vehicle and road constraints. These constraints consist of a driving corridor (e.g., lane boundaries), actuator physical limitations, and vehicle handling constraints. Furthermore, we define and motivate the different components necessary to formulate the Progress Maximization MPC (PM-MPC). We introduce the vehicle model in the road-aligned frame, the vehicle dynamics, the driving corridor constraints, and the cost function.

In short, the problem addressed in this chapter is the design of a controller that solves the following generic optimal control problem

$$\begin{aligned} \max & \quad \text{Progress along a given path} \\ \text{s.t.} & \quad \text{Vehicle model,} \\ & \quad \text{Actuation limits,} \\ & \quad \text{Handling limits,} \\ & \quad \text{Driving corridor.} \end{aligned}$$

In MPC, the prediction model trades off between fidelity and complexity. On the one hand, maximizing the progress along a given path requires high-speed vehicle driving and therefore, vehicle dynamics (e.g., sideslip and load transfer) can have significant impact. On the other hand, fast computational times and long prediction horizons are hard to achieve with nonlinear models and constraints. Therefore, we propose a two-folded approach. First, a kinematic vehicle model is used, which, by definition, does not include the forces that affect the motion. In order to account for the vehicle dynamics, explicit constraints are introduced in the MPC. We use a ‘g-g’ diagram to model the maximum lateral and longitudinal acceleration the vehicle can have simultaneously. Finally, to ensure that the tires of the vehicle are always operating in their linear force region, we limit the lateral acceleration depending on the maximum slip angle allowed, which in turn, constrains the vehicle longitudinal speed.

### 6.1.1 Vehicle Model

The time-domain equations of the kinematic vehicle model, describing the vehicle pose  $(x, y, \psi)$  in the global frame are given by (3.5). In this chapter, we complement model (3.5) with

$$\begin{aligned} \dot{v}_x &= a_x, \\ \dot{\kappa} &= c, \\ \dot{a}_x &= j_x, \end{aligned} \tag{6.2}$$

where  $v_x$  is the longitudinal component of the vehicle velocity,  $\kappa$  is the vehicle curvature,  $a_x$  is the vehicle longitudinal acceleration,  $c$  is the vehicle curvature rate, and  $j_x$  is the vehicle longitudinal jerk. We can relate the vehicle curvature  $\kappa$  with its steering angle  $\delta$  by

$$\kappa = \frac{\tan(\delta)}{L},$$

where  $L$  is the vehicle wheelbase.

We represent the vehicle in a road-aligned frame, introducing the variable  $s$  that represents the distance along a given path. The time-domain equations modeling the lateral displacement  $e_y$  and heading displacement  $e_\psi$  between the vehicle and

the road as a function of space  $s$  are given by (3.7). In order to eliminate the time and speed-dependency, the vehicle model is expressed in the space-domain by (3.8) complemented with

$$\begin{aligned} v'_x &= \frac{(1 - e_y \kappa_s)}{v_x \cos(e_\psi)} a_x, \\ \kappa' &= \frac{(1 - e_y \kappa_s)}{v_x \cos(e_\psi)} c, \\ a'_x &= \frac{(1 - e_y \kappa_s)}{v_x \cos(e_\psi)} j_x. \end{aligned} \quad (6.3)$$

Let  $z = [e_y, e_\psi, v_x, \kappa, a_x]^T$  and  $u = [c, j_x]^T$ . The linearized vehicle dynamics, at sample  $k$ , are described by  $z(k+1) = A_d(k)z(k) + B_d(k)u(k) + h_d(k)$ .  $A_d(k)$ ,  $B_d(k)$ , and  $h_d(k)$  are obtained via linearization of the space-based road-aligned vehicle model around a given reference path  $\{e_{y,j}^{\text{ref}}\}_{j=0}^N = 0$ ,  $\{e_{\psi,j}^{\text{ref}}\}_{j=0}^N = 0$ , and  $\{\kappa_j^{\text{ref}}\}_{j=0}^{N-1}$  using a first-order Taylor approximation and then discretization applying zero-order hold, as explained in Section 3.2. The MPC prediction horizon is represented by  $N$ . The control inputs are curvature  $\kappa$  and longitudinal acceleration  $a_x$ . However, in (6.3) they are represented as states and the model inputs are curvature rate  $c$  and longitudinal jerk  $j_x$ . Therefore, in order to have an accurate model discretization, in which the rates of the vehicle inputs are constant between samples, the control inputs between samples are also constant. This allows discretizing the model using larger distances, which allows longer prediction horizons.

The curvature rate and longitudinal jerk are limited by constant bounds as

$$\begin{aligned} \kappa_{\min} &\leq \kappa \leq \kappa_{\max}, \\ c_{\min} &\leq c \leq c_{\max}, \\ a_{x,\min} &\leq a_x \leq a_{x,\max}, \\ j_{x,\min} &\leq j_x \leq j_{x,\max}. \end{aligned} \quad (6.4)$$

The space-based modeling and the consequent elimination of speed dependence is of extreme importance in our formulation, since velocity is modeled as a separated state. In any case, the lateral and longitudinal modeling are not decoupled, as the vehicle dynamics constraints introduced in the remainder of the section demonstrate. If the vehicle lateral dynamics ( $e_y, e_\psi$ ) would be dependent on the longitudinal velocity, it would not be possible to have uniform spatial discretization, since space and time discretization would be influenced by vehicle speed.

### 6.1.2 Handling Limits

We describe the handling limits of the vehicle using three relationships. First, the vehicle slip angles, which are required to be small, depend on the vehicle lateral acceleration. Second, the maximum longitudinal speed allowed depends on the

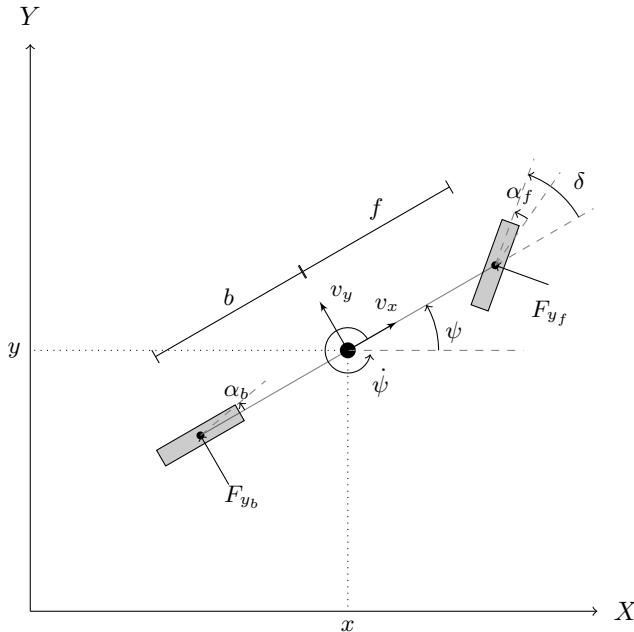


Figure 6.1: Illustration of a dynamic vehicle model.

maximum vehicle lateral acceleration and curvature. Third, the maximum force the tires can produce are modeled using a ‘g-g’ diagram.

We start by introducing the set of differential equations describing the lateral and longitudinal forces and the momentum of a 2-axles bicycle model<sup>1</sup>.

The dynamics of a vehicle described by a 2-axles bicycle model, as illustrated in Figure 6.1, are defined by

$$\begin{aligned} m(\dot{v}_x - \dot{\psi}v_y) &= F_x - F_{y_f} \sin \delta, \\ m(\dot{v}_y + \dot{\psi}v_x) &= F_{y_b} + F_{y_f} \cos \delta, \\ J_z \ddot{\psi} &= fF_{y_f} \cos \delta - bF_{y_b}, \end{aligned}$$

where  $v_x$  and  $v_y$  denote the longitudinal and the lateral component of the vehicle velocity, respectively, and  $\dot{\psi}$  denotes the yaw rate. The constants  $m$  and  $J_z$  denote the vehicle mass and moment of inertia about the yaw axis, respectively, and  $f$  and  $b$  represent the distances from the center of gravity to the front and rear axles,

---

<sup>1</sup>In Section 3.1, we introduce the corresponding equations for a 4-axles bicycle model. That model is used for simulation analysis of the motion control design in Chapters 7 and 8. In this chapter, we are interested in deriving the dynamics of a standard bicycle model.

respectively. The longitudinal force  $F_x = ma_x$  and the front axle wheel angle  $\delta$  are the inputs of the vehicle model. The lateral forces acting on each axle  $F_{y_f}$  and  $F_{y_b}$  are assumed to be linear functions of the slip angles

$$\begin{aligned} F_{y_f} &= -C_f \alpha_f, \\ F_{y_b} &= -C_b \alpha_b, \end{aligned} \quad (6.6a)$$

where the constants  $C_f$  and  $C_b$  are the cornering stiffnesses of the front and the rear axles, respectively, and the slip angles are assumed to be small. We neglect any longitudinal slip. The slip angles are defined as

$$\begin{aligned} \alpha_f &= \arctan\left(\frac{v_y + \dot{\psi}f}{v_x}\right) - \delta, \\ \alpha_b &= \arctan\left(\frac{v_y - \dot{\psi}b}{v_x}\right). \end{aligned}$$

Assuming that the vehicle is at steady-state cornering and that  $v_x \gg v_y$ , we can relate the curvature of the vehicle with its steering angle using

$$\delta = \arctan((L + K_{us}v_x^2)\kappa),$$

where  $L = f + b$  is the wheelbase of the vehicle and  $K_{us}$  is the understeering gradient of the vehicle, which, by definition, contains the information about the vehicle mass, its center of mass, and the wheels cornering stiffnesses. Also, we can relate the vehicle slip angles with its lateral acceleration

$$\begin{aligned} \alpha_f &= -\frac{mb}{LC_f}a_y, \\ \alpha_b &= -\frac{mf}{LC_b}a_y. \end{aligned}$$

From these equations, it is possible to compute the maximum lateral acceleration  $a_{y,\max}$  for the maximum slip angle allowed  $\alpha_{\max}$  such that the tires are operating in their linear region.

Still assuming steady-state cornering, the vehicle lateral acceleration is given by

$$a_y = v_x^2 \kappa, \quad (6.9)$$

and so, the vehicle maximum speed  $v_{x,\max}$  can be derived from (6.9) as

$$v_{x,\max} = \sqrt{\left|\frac{a_{y,\max}}{\kappa}\right|}, \quad (6.10)$$

where  $a_{y,\max}$  is the vehicle maximum lateral acceleration allowed.

In order to formulate PM-MPC as a convex optimization problem, we are interested in describing the vehicle maximum speed as a linear function of curvature. To that end, we approximate (6.10) using the linear terms of its Taylor expansion around  $\kappa_{\text{ref}}$ , yielding

$$v_{x,\max} \approx Q_\kappa \kappa + Q,$$

where

$$\begin{aligned} Q_\kappa &= -\frac{\sqrt{a_{y,\max}} \kappa_{\text{ref}}}{2|\kappa_{\text{ref}}|^{\frac{5}{2}}}, \\ Q &= \sqrt{\frac{a_{y,\max}}{|\kappa_{\text{ref}}|}} + \frac{\sqrt{a_{y,\max}} \kappa_{\text{ref}}^2}{2|\kappa_{\text{ref}}|^{\frac{5}{2}}}. \end{aligned}$$

Finally, it is necessary to describe the interaction between the lateral and longitudinal forces. The maximum force a tire can produce is limited by friction. The vehicle handling limits are often described using a friction limit circle on a ‘g-g’ diagram.

The friction limit circle can be described by

$$(\mu g)^2 = a_x^2 + a_y^2, \quad (6.11)$$

where  $\mu$  is an effective friction coefficient, and  $g$  is the gravity acceleration. Although  $\mu$  is constant, under forward acceleration other limiting factors than grip come into play (e.g., engine power in higher gears, aerodynamic drag). Therefore, typical ‘g-g’ diagrams are distorted on the upper part accounting for those limitations. Again, we are interested in using (6.11) as a constraint limiting the vehicle combined lateral and longitudinal acceleration and still be able to formulate an optimization problem that can be solved using standard QP solvers. Therefore, the ‘g-g’ diagram limit circle is conservatively approximated by an inscribed polytope with  $N_v$  vertices, as shown in Figure 6.2. For the sake of accurately approximating the limit circle, the approximating polytope should always include the lateral and longitudinal acceleration limits. Then, the friction circle can be described by a set of linear constraints that depend on the vehicle longitudinal speed and acceleration and on the vehicle curvature. The larger the number of vertices  $N_v$ , the more accurate is the ‘g-g’ diagram approximation. However, increasing the number of vertices leads to a larger number of constraints and consequently to an optimization problem that is computationally more expensive.

To obtain these constraints, first note that each edge of the polytope approximation is of the form  $a_x = p_{m,j} a_y + p_{b,j}$ , where  $p_{m,j}$  is the slope and  $p_{b,j}$  the  $y$ -intercept of edge  $j$ , and  $j = 1, \dots, N_v$ . Then, with the same purpose of obtaining a convex optimization problem, we linearize each one of the expressions describing the vehicle edge around a given  $v_{x,\text{ref}}$  and  $\kappa_{\text{ref}}$  and use (6.9) to obtain linear constraints of the form

$$H_j a_x \leq G_{v,j} v_x + G_{\kappa,j} \kappa + G_j,$$

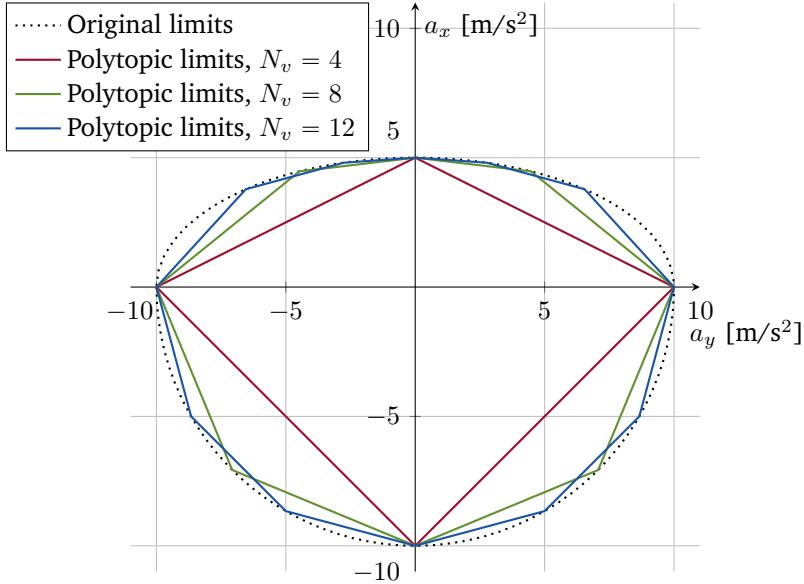


Figure 6.2: Example of a ‘g-g’ diagram and some polytopic approximations.

where

$$\begin{aligned}
 H_j &= \text{sign}(p_{b,j}), \\
 G_{v,j} &= \text{sign}(p_{b,j})2p_{m,j}\kappa_{\text{ref}}v_{x,\text{ref}}, \\
 G_{\kappa,j} &= \text{sign}(p_{b,j})p_{m,j}v_{x,\text{ref}}^2, \\
 G_j &= \text{sign}(p_{b,j})(p_{b,j} - 2p_{m,j}\kappa_{\text{ref}}v_{x,\text{ref}}^2).
 \end{aligned} \tag{6.12}$$

The vector versions of (6.12), where  $j = 1, \dots, N_v$ , are  $\bar{H}$ ,  $\bar{G}_v$ ,  $\bar{G}_\kappa$ , and  $\bar{G}$ .

### 6.1.3 Driving Corridor

The driving corridor (e.g., lane boundaries), when the vehicle is modeled in a road-aligned frame, is defined by spatially dependent state bounds

$$e_{y,\min}(s) \leq e_y(s) \leq e_{y,\max}(s).$$

In the presence of obstacles (moving or static), a corridor can be determined based on heuristics (overtaking left or right). We assume that a driving corridor is given. Moreover, we assume forward motion, which can also be defined by another spatially dependent state bounds

$$e_{\psi,\min}(s) \leq e_\psi(s) \leq e_{\psi,\max}(s),$$

where  $-\frac{\pi}{2} < e_{\psi,\min}(s) \leq 0$  and  $0 \leq e_{\psi,\max}(s) < \frac{\pi}{2}$ .

### 6.1.4 Progress Along a Given Path

In the road-aligned vehicle model, the velocity along the path  $\dot{s}$  can be seen as a measure of progress along  $s$  and is given by

$$\dot{s} = \frac{1}{1 - \kappa_s e_y} v \cos(e_\psi). \quad (6.13)$$

Maximizing the progress along a given path is equivalent to maximize  $\dot{s}$  (or to minimize  $\frac{1}{\dot{s}}$ ). However,  $\frac{1}{\dot{s}}$  is clearly non-convex. To overcome this, we perform a second-order Taylor approximation around  $e_{\psi,\text{ref}} = 0$  and a first-order Taylor approximation around  $e_{y,\text{ref}} \neq 0$  and  $v_{x,\text{ref}} \neq 0$  and obtain

$$\min \gamma_1 e_\psi^2 + \gamma_2 e_y + \gamma_3 v_x,$$

where

$$\gamma_1 = \frac{1}{2} \frac{1 - e_{y,\text{ref}} \kappa_s}{v_{x,\text{ref}}}, \quad (6.14)$$

$$\gamma_2 = -\frac{\kappa_s}{v_{x,\text{ref}}}, \quad (6.15)$$

$$\gamma_3 = -\frac{1 - e_{y,\text{ref}} \kappa_s}{v_{x,\text{ref}}^2},$$

where the terms that do not depend on model states or inputs are not included. In other words, the cost function encourages progress maximization, since it

1. minimizes the heading displacement with respect to the path heading. This also makes the approximation  $e_{\psi,\text{ref}} = 0$  valid;
2. maximizes the lateral displacement with the same sign as the road curvature (i.e., when the road curves to the left  $\kappa_s > 0$  and, therefore,  $e_y > 0$  and vice-versa);
3. maximizes the longitudinal speed.

### 6.1.5 Optimal Control Problem Formulation

We formulate the PM-MPC as receding-horizon optimal control problem as follows:

$$\min_{\bar{c}, \bar{j}_x, \sigma} \sum_{i=1}^N \gamma_{1,i} e_{\psi,i}^2 + \gamma_{2,i} e_{y,i} + \gamma_{3,i} v_{x,i} + \sigma^T \bar{\beta} \sigma \quad (6.16)$$

$$\text{s.t. } z_i = A_d(i-1)z_{i-1} + B_d(i-1)u_i + h_d(i-1), \quad i = 1, \dots, N, \quad (6.17)$$

$$z_0 = z(t), \quad (6.18)$$

$$e_{y,\min,i} - \sigma_{e_y} \leq e_{y,i} \leq e_{y,\max,i} + \sigma_{e_y}, \quad i = 1, \dots, N, \quad (6.19)$$

$$e_{\psi,\min,i} - \sigma_{e_{\psi}} \leq e_{\psi,i} \leq e_{\psi,\max,i} + \sigma_{e_{\psi}}, \quad i = 1, \dots, N, \quad (6.20)$$

$$-\sigma_{v_x} \leq v_{x,i} \leq Q_{\kappa,i} \kappa_i + Q_i + \sigma_{v_x}, \quad i = 1, \dots, N, \quad (6.21)$$

$$\bar{H}_i a_{x,i} \leq \bar{G}_{v,i} v_{x,i} + \bar{G}_{\kappa,i} \kappa_i + \bar{G}_i + \sigma_{a_x}, \quad i = 1, \dots, N, \quad (6.22)$$

$$\kappa_{\min} - \sigma_{\kappa} \leq \kappa_i \leq \kappa_{\max} + \sigma_{\kappa}, \quad i = 1, \dots, N, \quad (6.23)$$

$$c_{\min} \leq c_i \leq c_{\max}, \quad i = 1, \dots, N, \quad (6.24)$$

$$j_{x,\min} \leq j_{x,i} \leq j_{x,\max}, \quad i = 1, \dots, N, \quad (6.25)$$

$$\sigma_{e_y} \geq 0, \sigma_{e_{\psi}} \geq 0, \sigma_{v_x} \geq 0, \sigma_{\kappa} \geq 0, \sigma_{a_x} \geq 0, \quad (6.26)$$

$$(6.27)$$

with decision variables  $\bar{c} = \{c_i\}_{i=1}^N$ ,  $\bar{j}_x = \{j_{x,i}\}_{i=1}^N$  and  $\sigma = [\sigma_{e_y}, \sigma_{e_{\psi}}, \sigma_{v_x}, \sigma_{\kappa}, \sigma_{a_x}]^T$ . The vector  $\sigma$  is a slack variable that ensures feasibility of the optimization problem by softening otherwise hard constraints. The initial state  $z_0$  is given by the current state of the vehicle  $z(t)$  (i.e., the measurement at time  $t$  of the vehicle states), where  $e_y(t)$  and  $e_{\psi}(t)$  are computed using an orthogonal projection of the vehicle position into the reference path. The constraints (6.19) to (6.25) are thoroughly explained and discussed in Sections 6.1.1 to 6.1.3. The cost function is described in Section 6.1.4 and  $\bar{\beta} = \text{diag}(\beta_{e_y}, \beta_{e_{\psi}}, \beta_{v_x}, \beta_{\kappa}, \beta_{a_x})$  penalizes constraint violation.

## 6.2 Simulation Results

To demonstrate the performance of the PM-MPC, we use an S-shaped road with a width of 6 meters, where the centerline consists of clothoid segments appended together<sup>2</sup>. Consequently, the driving corridor is defined (if there are no obstacles) as  $e_{y,\max} = -e_{y,\min} = 3$  m and  $e_{\psi,\max} = -e_{\psi,\min} = \frac{\pi}{4}$  rad. The maximum curvature along the centerline of the road, is  $0.12 \text{ m}^{-1}$ . The polytopic approximation of the ‘g-g’ diagram uses  $N_v = 8$  vertices. In simulation, we noticed using a larger number of vertices for the polytopic approximation had neglectable impact on the optimization problem solution, but a significant one in computational time.

---

<sup>2</sup>In fact, we can use the resultant clothoid-based sparse path from Chapter 4. However, in this chapter, the reference path is not resultant from the sparsification algorithm and it is generated manually beforehand.

We apply optimization problem (6.16) in two different scenarios, namely *offline path optimization* using the whole reference path and *online receding-horizon path optimization*, where the nonlinear vehicle model with dynamics (6.5) is simulated. The vehicle parameters used in the simulations are summarized in Table 6.1.

In the offline path optimization scenario, the optimization problem (6.16) is solved iteratively using the kinematic prediction model (3.8) combined with (6.3). The prediction horizon is the length of the road centerline (or, more precisely, the number of points used to discretize the road centerline every meter). At each iteration, we compute the time it takes for the kinematic vehicle to travel along the road. In the next iteration, the optimal path, speed, and inputs, are used as the reference to accurately linearize the vehicle model and optimization constraints. When the time the vehicle needs to complete the path stops decreasing, it is assumed that the path that maximizes the progress along the road centerline is found. This time is later used as a benchmark to the receding-horizon optimization.

In the online receding-horizon path optimization scenario, the optimization problem (6.16) is also solved iteratively. However, in this case, we use a shorter prediction horizon of  $N = 14$  points with a discretization step of 6 meters. In this case, the number of points in the prediction horizon and the discretization step length are chosen such that the prediction horizon is long enough to optimize over the subsequent turn. A shorter discretization step would be beneficial in terms of model prediction accuracy, since the control input is not constant for so long. However, a shorter discretization step would require more prediction horizon points, which would lead to a more expensive optimization problem. The prediction model is still the kinematic prediction model (3.8) combined with (6.3), but the optimal inputs are used to control the dynamic vehicle model (6.5). We also demonstrate the effectiveness of the PM-MPC in the presence of obstacles, here modeled by changing the driving corridor upper and lower bounds. In the simulation with obstacles, a road width of 8 meters is used instead, to give more space for placing obstacles along the track. Each obstacle is 2 meters wide and is detected when the vehicle is closer than 50 meters from it.

To perform the complete path optimization, we use the MATLAB toolbox YALMIP (Löfberg, 2004) using the solver *quadprog*. To run the PM-MPC in receding-horizon and to evaluate its execution time, we have used CVXGEN (Mattingley and Boyd, 2012). All simulations are conducted on a laptop PC running Windows 10 equipped with an Intel Core i7 CPU@2.70GHz, 16.0GB of memory, and using MATLAB R2016b.

### 6.2.1 Complete Path Optimization

Figure 6.3 depicts the solution of the optimal control problem (6.16) in the global and road-aligned frame, for the complete path optimization scenario. Note that the path uses the space available between the lane boundaries to lower the absolute curvature needed to perform the turns, as shown in Figure 6.5. Therefore, the PM-MPC maximizes the vehicle speed in the cornering phase by using a larger and

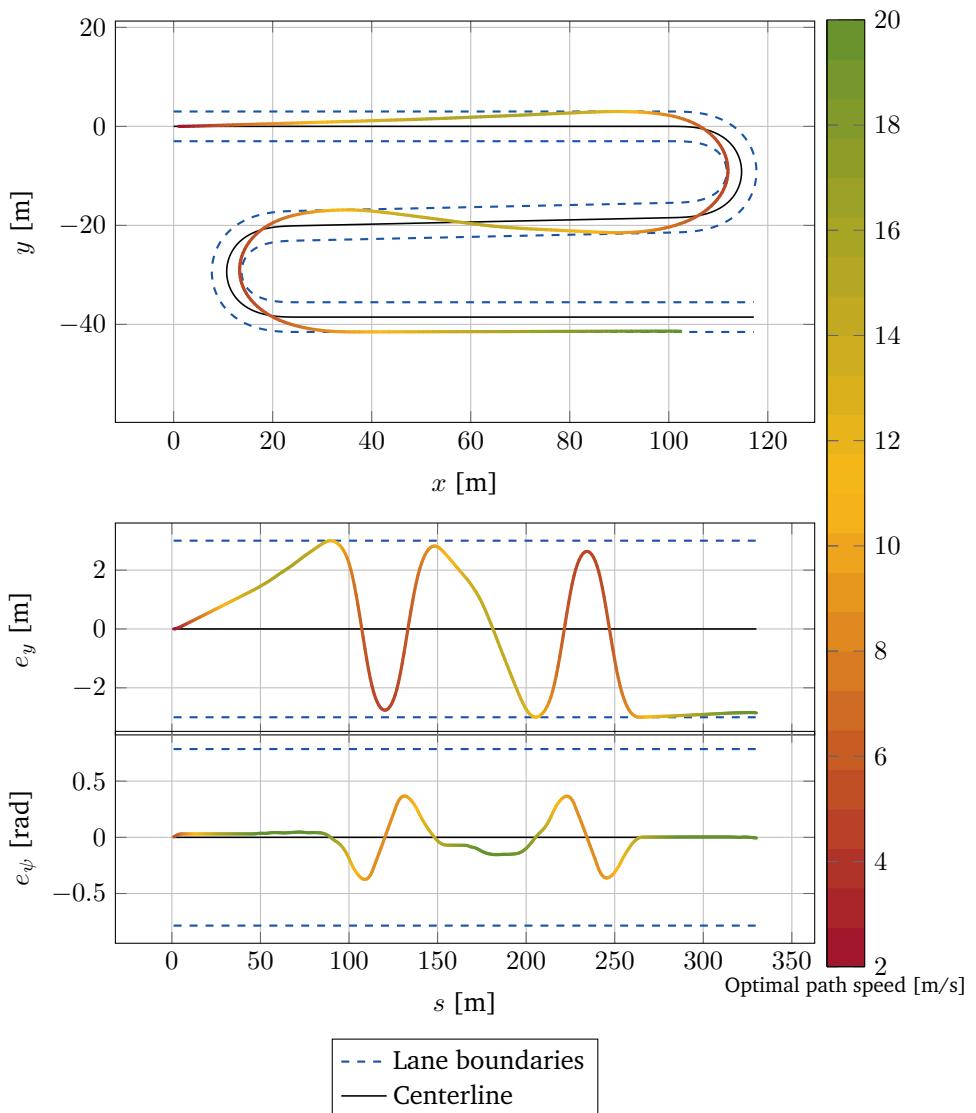


Figure 6.3: *Offline path optimization scenario*. Optimal path resulting from applying optimization (6.16) to the whole path. Note that the path in the global frame (upper plot) is obtained after the transformation from the road-aligned frame (lower plot), in which computations are conducted.

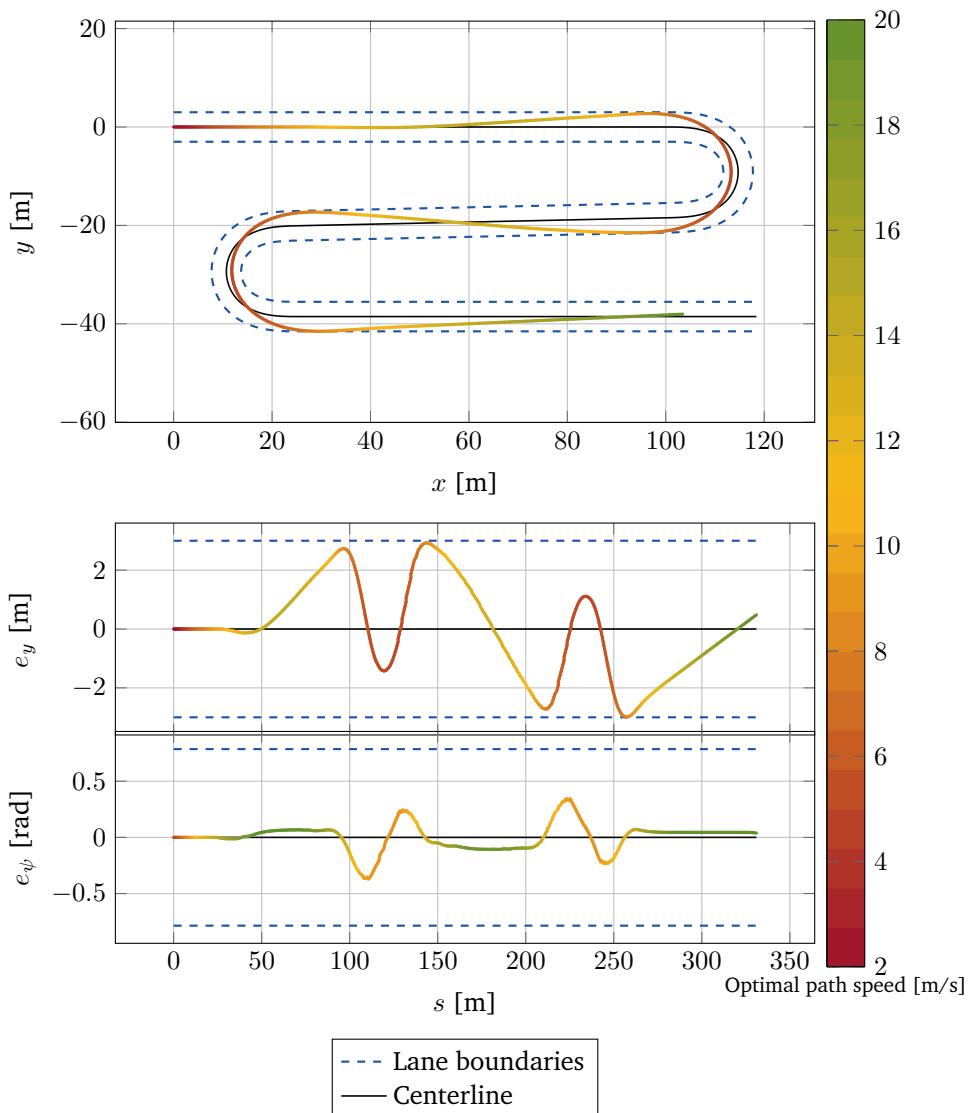


Figure 6.4: *Online receding-horizon path optimization scenario.* Vehicle path resulting from applying optimization (6.16) in receding-horizon while simulating a dynamic vehicle model. Note that the path in the global frame (upper plot) is obtained after the transformation from the road-aligned frame (lower plot), in which computations are conducted.

Parameters	Value
$\kappa_{\max}$	$0.2 \text{ m}^{-1}$
$\kappa_{\min}$	$-0.2 \text{ m}^{-1}$
$a_{x,\max}$	$4.2 \text{ m/s}^2$
$a_{x,\min}$	$-8.4 \text{ m/s}^2$
$a_{y,\max}$	$8.4 \text{ m/s}^2$
$a_{y,\min}$	$-8.4 \text{ m/s}^2$
$L$	4 m
$f$	2 m
$b$	2 m
$m$	2500 kg
$J_z$	$3000 \text{ kgm}^2$
$C_{12}$	20000 kN/rad
$C_{34}$	20000 kN/rad
$\alpha_{\max}$	0.07 rad

Table 6.1: Vehicle parameters

constant turning radius at the apex. The apex is a point on the inside edge of a curve that allows the vehicle to have the smallest curvature and, consequently, the largest speed possible when cornering. The apex can be seen in Figure 6.6, when the vehicle has constant curvature and longitudinal speed, which leads to constant lateral acceleration. As desired, the trail-braking and throttle-on-exit are performed using the maximum handling capacity of the vehicle, here represented by the polytopic approximation of the ‘g-g’ diagram in Figure 6.6. Also, the vehicle has the maximum lateral acceleration at the apex in order to perform the turn as fast as possible.

### 6.2.2 Receding-Horizon Control and Obstacle Avoidance

In the receding-horizon path optimization scenario, the optimization runs twice before an input is given to the vehicle. In the first run, the reference values used in the linearization of the vehicle model and some of the constraints are taken from the previous solution. However, as the horizon shifts forward, these references refer to a previous instant, and therefore, the optimal solution will differ slightly making the linearization no longer accurate. Thus, linearizing a second time around new reference values leads to a better linearization accuracy.

Figure 6.4 depicts the solution of the optimal control problem (6.16) in the global and road-aligned frame, when the optimization is performed in receding-horizon and the first element of the optimal predicted input sequence is applied to the dynamic vehicle (6.5) running at 20 Hz. From the figures, it is clear that the prediction model based on a kinematic vehicle does not capture important aspects, such as sideslip and turning momentum. Consequently, the vehicle does

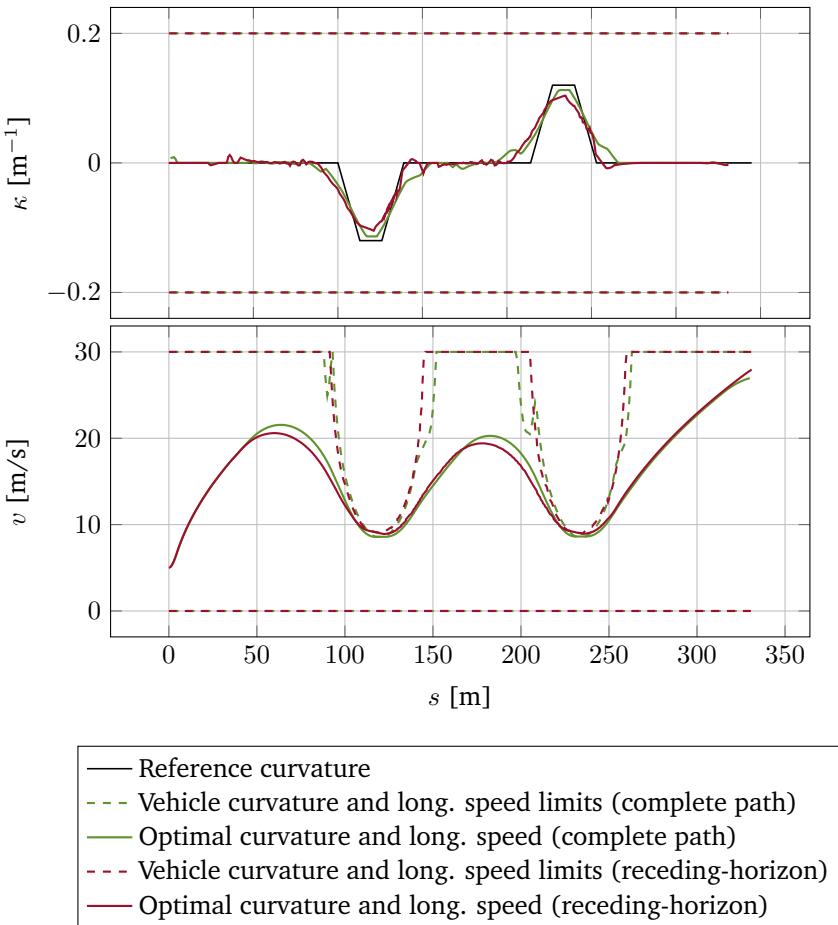


Figure 6.5: *Offline path and online receding-horizon optimization scenario. Vehicle curvature and longitudinal speed comparison.*

not perform the turn tangentially to the inner lane, due to inaccurate predictions. However, by limiting the amount of vehicle lateral acceleration, it is possible to limit the amount of tire slip, minimizing the impact of the unmodeled vehicle dynamics. The tire slip angles are shown in Figure 6.7, where the limits of 0.07 rad are never exceeded (i.e., the tires never leave their linear force region). Note that the lateral acceleration is, in fact, limited based on a lower slip angle value, since the expression relating the lateral acceleration and the slip angle assumes cornering at steady-state and this is not always the case. The vehicle starts to decelerate earlier, when compared to the complete path optimization speed, due to the shorter prediction horizon (see Figure 6.5). Consequently, the controller predicts that the

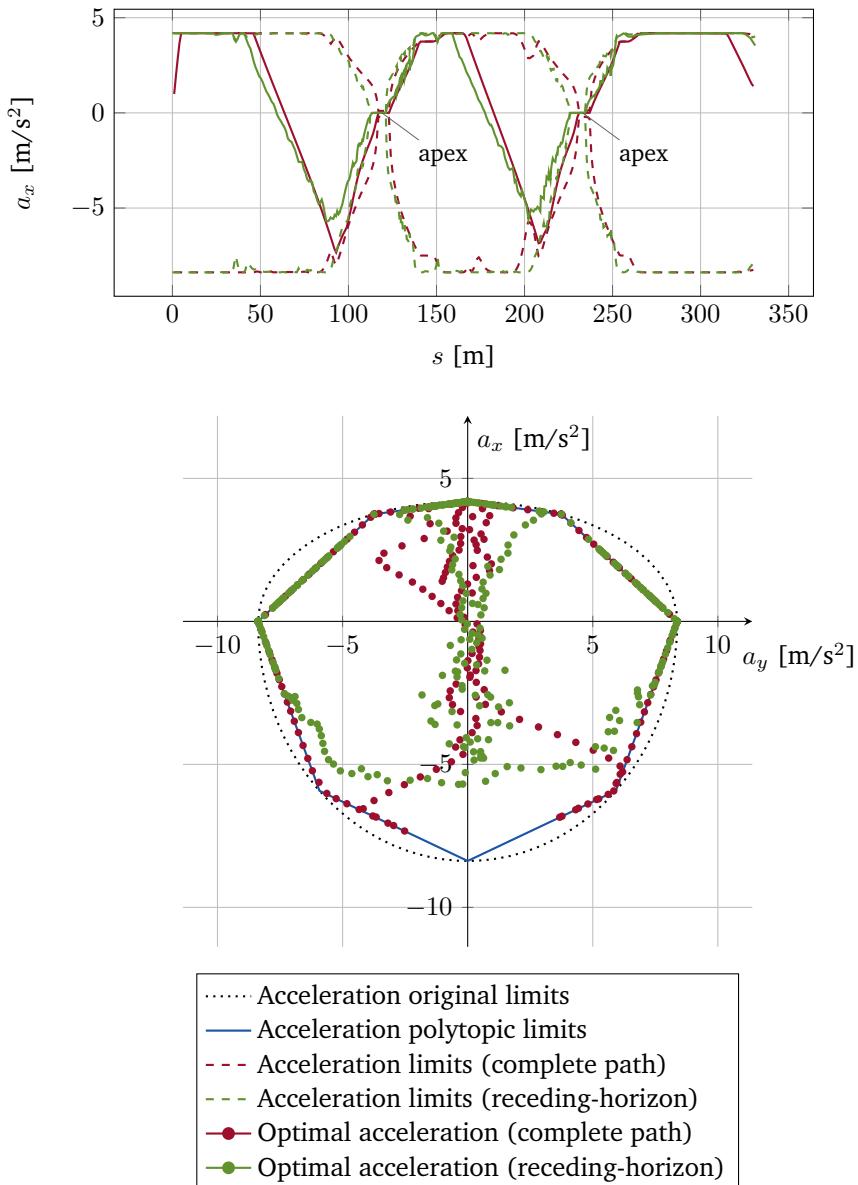


Figure 6.6: *Offline path and online receding-horizon optimization scenario.* Vehicle longitudinal acceleration (upper plot). Vehicle longitudinal and lateral acceleration and the ‘g-g’ diagram (lower plot).

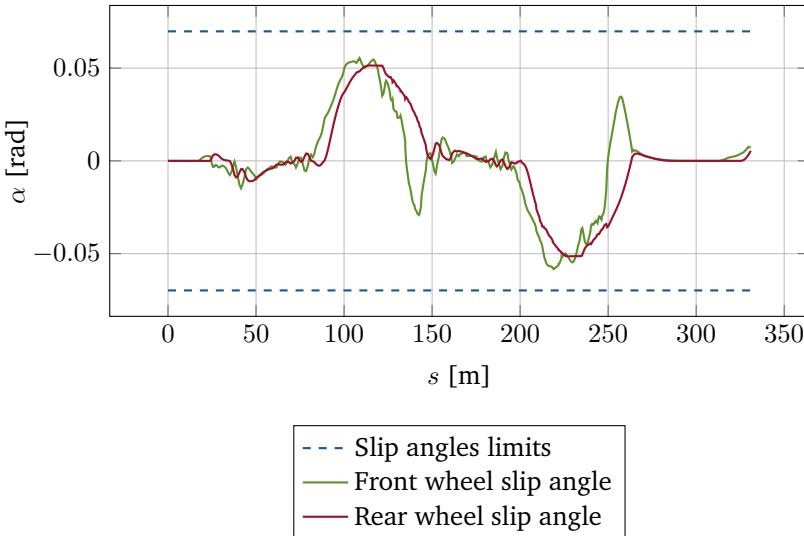


Figure 6.7: *Online receding-horizon path optimization scenario.* Vehicle slip angles.

forthcoming turn has to be overcome with larger curvature than what it actually needs. It is interesting to analyze the vehicle behavior after the last turn. The velocity maximization term on the PM-MPC cost function causes the acceleration to be maximum and consequently, the vehicle curvature to be zero. Even if the heading alignment would require a small curvature then, according to the coarse polytopic approximation of the g-g diagram shown in Figure 6.6, a significantly smaller longitudinal acceleration would be allowed.

Finally, Figure 6.8 depicts the solution of the optimal control problem (6.16) in the global and road-aligned frame, when the optimization is performed in receding-horizon in the presence of obstacles. As shown in Figure 6.8, the obstacles are successively avoided, and PM-MPC uses the space available within the driving corridor to maximize the progress along the centerline. The analysis of vehicle motion after the last turn also holds in this case. Note that the driving corridor is tighter during part of the turn, causing a larger turn exit heading, which requires extra curvature to avoid violating the lane boundaries.

As an example, in the complete path optimization scenario, it is estimated that the kinematic vehicle finishes the 330 meters road (measured along the centerline) in 22.5 seconds. In the receding-horizon optimization scenario, the same road takes 23.5 seconds to complete. Moreover, two iterations of (6.16) take at most 0.040 seconds and in average 0.025 seconds to run.

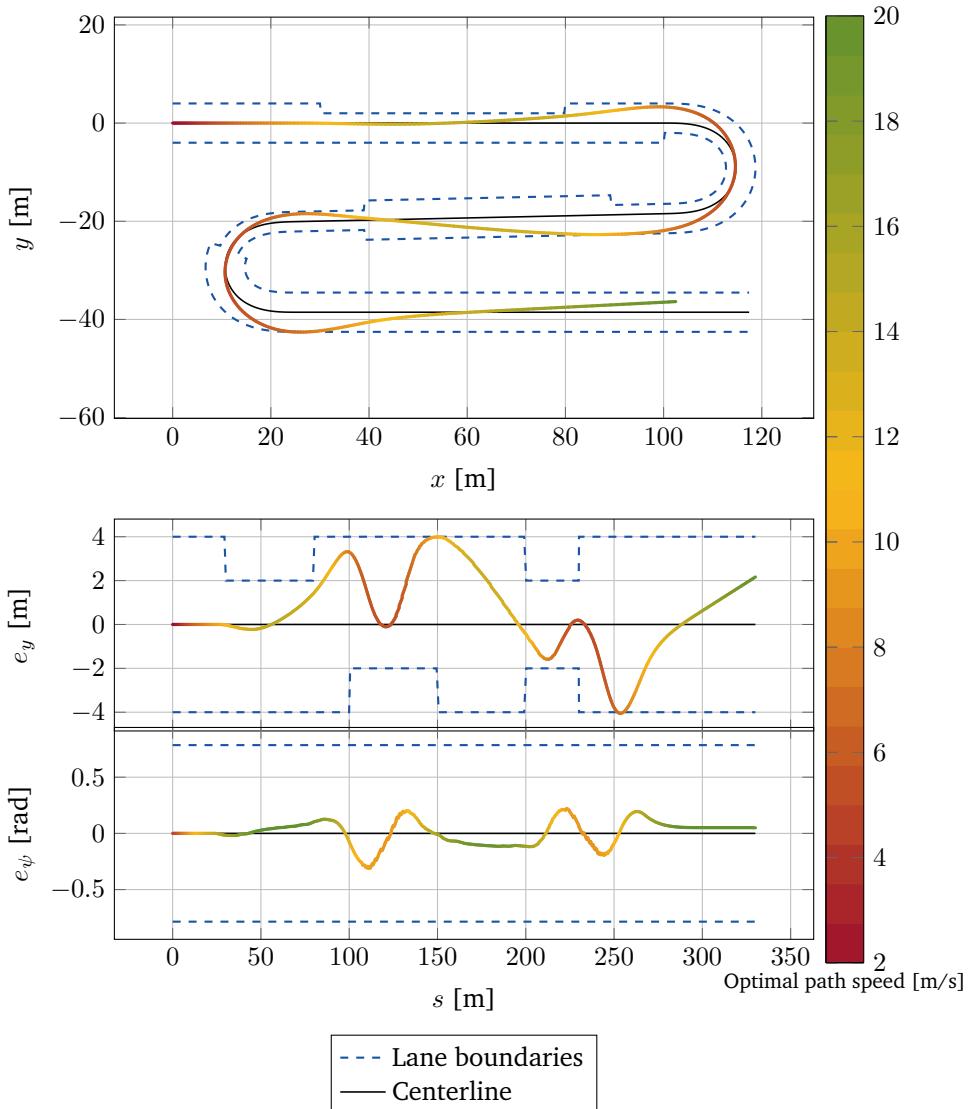


Figure 6.8: *Online receding-horizon path optimization scenario*. Vehicle path resulting from applying optimization (6.16) in receding-horizon while simulating a dynamic vehicle model in the presence of obstacles. Note that the path in the global frame (upper plot) is obtained after the transformation from the road-aligned frame (lower plot), in which computations are conducted.

### **6.3 Summary**

In this chapter, we proposed a PM-MPC for autonomous driving. The goal of such controller is to maximize the progress of the vehicle along a given path while respecting actuator, handling, and driving corridor constraints. The vehicle motion is predicted using a simple kinematic vehicle together with more complex vehicle dynamics constraints. These constraints include a polytopic approximation of a g-g diagram and tire slip restriction to the tire linear force region by limiting the lateral acceleration and consequently, the longitudinal velocity. The PM-MPC was formulated as a convex optimization by linearizing the vehicle model around previous solutions, and by using linearized versions of the constraints modeling the handling dynamics. This yielded extremely efficient computation times, which allowed us to demonstrate the effectiveness of the PM-MPC even in the presence of obstacles.

## Chapter 7

# Smooth and Accurate Model Predictive Controller

Combining smooth driving with accurate path following is challenging due to their contradictory nature. The development of control strategies targeting this behavior is important for autonomous driving in general and for industrial applications in particular, where vehicles are supposed to run uninterruptedly during long periods of time. Aggressive steering results in a high “wear and tear”, which shortens the lifetime of the trucks and, as a consequence, increases the costs for the operators. Moreover, rapid steering wheel angle changes can lead to aggressive and unpredictable driving, increased chance of vehicle rollover, and damage to tires and steering components.

In this chapter, we propose a Smooth and Accurate MPC (SA-MPC) for autonomous driving. Standard MPC approaches aim at minimizing the deviation from a reference path. In our proposed formulation, we add a second objective to improve the driving smoothness. In other words, we minimize the first- and second-order spatial derivatives of the vehicle curvature (i.e., we encourage PWA curvature profiles). The successive linearization of the nonlinear vehicle model leads to a LTV-MPC formulation. Since the beginning of 2016, SA-MPC is the main path following controller used in four different autonomous Scania construction trucks and one autonomous Scania bus.

The outline of this chapter is as follows. In Section 7.1, we present the main idea behind the development of SA-MPC, recall the space-based road-aligned vehicle, design the SA-MPC cost function, and analyze the influence of the controller tuning parameters. In Section 7.2, the performance of the SA-MPC is benchmarked against a Pure-Pursuit Controller (PPC) and a standard MPC. The SA-MPC is also compared with a PPC, when both are deployed in a Scania construction truck, and the validity of the simulation environment is discussed by comparing experimental and simulation results. In Section 7.3, we summarize the chapter and provide some concluding remarks.

## 7.1 Problem Formulation

We intend to design a controller that steers the vehicle to a reference path under certain constraints. Typically, the vehicle inputs are limited due to actuator limitations, and physical constraints are necessary to ensure safety, for example. Moreover, the vehicle dynamics can be described by a discrete-time nonlinear function. Also, we consider that the given reference path is feasible and obstacle free.

In short, the problem addressed in this chapter is the design of a controller that solves the following generic optimal control problem:

$$\max \quad \text{Tracking accuracy} + \text{driving smoothness} \quad (7.1a)$$

$$\text{s.t.} \quad \text{Vehicle model,} \quad (7.1b)$$

$$\text{Actuation limits,} \quad (7.1c)$$

$$\text{Driving corridor.} \quad (7.1d)$$

### 7.1.1 Vehicle Model

Consider the space-based road-aligned model (3.8) presented in Section 3.2, which describes the lateral displacement  $e_y$  and heading displacement  $e_\psi$  between the vehicle and a given reference path. Let the state vector be  $z = [e_y, e_\psi]^T$ , the control input be  $u = \kappa$  and (3.8) be described by  $z' = f(z, u)$ . The reference path is discretized such that  $\{s_j\}_{j=0}^N = \{s_0, s_1, \dots, s_N\}$ . We linearize and discretize the vehicle model around the reference path  $\{e_{y,j}^{\text{ref}}\}_{j=0}^N = 0$ ,  $\{e_{\psi,j}^{\text{ref}}\}_{j=0}^N = 0$ , and  $\{\kappa_j^{\text{ref}}\}_{j=0}^{N-1} = \{\frac{1}{\rho_{s,j}}\}_{j=0}^{N-1}$  using a first-order Taylor approximation and then apply forward Euler discretization, using a discretization time step of  $T_s = 200$  ms. The MPC prediction horizon is given by  $N$ . The linearized and discretized system dynamics are given by

$$z(k+1) = A_d(k)z(k) + B_d(k)u(k) + h_d(k).$$

Furthermore, vehicle curvature and curvature rate are limited by constant bounds

$$\kappa_{\min} \leq \kappa_k \leq \kappa_{\max}, \quad k = 1, \dots, N,$$

$$c_{\min} \leq c_k \leq c_{\max}, \quad k = 1, \dots, N,$$

where  $c_k = \frac{\kappa_{k+1} - \kappa_k}{\Delta_s}$  with  $\Delta_s = T_s v_x$ . It is worth noting that the longitudinal velocity  $v_x$  is assumed constant throughout the prediction horizon due to the slow longitudinal dynamics of a heavy-duty vehicle.

### 7.1.2 Driving Corridor

We assume that a driving corridor (e.g., lane boundaries) is given. When the vehicle is modeled in a road-aligned frame, the driving corridor is defined by spatially

dependent state bounds

$$e_{y,\min,k} \leq e_{y,k} \leq e_{y,\max,k}, \quad k = 1, \dots, N.$$

### 7.1.3 Cost Function

The cost function of the SA-MPC consists of two terms that trade off vehicle motion smoothness (or comfort) for path following accuracy. To that end, the first term of the objective function penalizes the first- and the second-order spatial derivative of the vehicle curvature and the second term penalizes large deviations from the path.

The first-order spatial derivative of the curvature is related to the vehicle lateral jerk, for which high values are perceived by a human as uncomfortable. The second-order spatial derivative of the curvature is related to lateral snap. Consequently, it is desirable to maintain both the first- and second-order spatial derivatives as low as possible while tracking the reference path in order to achieve smooth driving. With this formulation, the vehicle curvature is encouraged to be PWA. Since the curvature in clothoids varies linearly with the path arc-length, we use the smoothness and comfort characteristics of clothoids in the controller formulation. Clothoids are used for road design, since their curvature varies linearly with the path arc-length. This provides a linear change in the vehicle steering angle to perform the turn (Marzbani et al., 2015). High values of lateral jerk are perceived by humans as uncomfortable and, by limiting the constant rate of change, we limit the amount of lateral jerk, minimizing discomfort and snap. Lateral acceleration is not explicitly considered in the cost function, but it is typically limited by saturating the vehicle speed with *a priori* knowledge of the path curvature.

To calculate the second derivative of the curvature, we use a second-order difference operator  $D_2$  that is inversely weighted with the distance between the waypoints (i.e., the shorter the distance, the higher the weight is). The operator  $D_2$  used to approximate the second-order derivatives of the curvature is based on finite differences. The first derivative of the curvature function with respect to traveled distance is approximated as

$$\frac{d\kappa(s)}{ds} \approx \frac{\kappa_{i+1} - \kappa_i}{\Delta s}, \quad i = 1, \dots, N - 1, \quad (7.2)$$

and consequently the second discrete derivative is

$$\frac{d^2\kappa(s)}{ds^2} \approx \frac{\frac{\kappa_{i+1} - \kappa_i}{\Delta s} - \frac{\kappa_i - \kappa_{i-1}}{\Delta s}}{\Delta s} = \frac{(\kappa_{i+1} - 2\kappa_i + \kappa_{i-1})}{\Delta s^2}, \quad i = 1, \dots, N - 1. \quad (7.3)$$

Since  $D_2$  is a matrix operator over a vector, we stack (7.3) from  $i = 1, \dots, N$  and get

$$D_2 = \frac{1}{\Delta s^2} \begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 & 1 \end{bmatrix}. \quad (7.4)$$

Similarly to  $D_2$ ,  $D_1$  is obtained by stacking (7.2)

$$D_1 = \frac{1}{\Delta s} \begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & \ddots & \ddots & & \\ & & & 1 & -1 & \end{bmatrix}. \quad (7.5)$$

#### 7.1.4 Optimal Control Problem Formulation

Assuming that a full measurement or estimate of the state  $\mathbf{z}(t)$  is available, the SA-MPC is formulated as the following LTV-MPC problem, which is solved at every time instant  $t > 0$

$$\min_{\bar{\kappa}, \sigma_{e_y}} \|D_2 \bar{\kappa}\|_2^2 + \alpha \|D_1 \bar{\kappa}\|_2^2 + \bar{\lambda} \|\bar{\sigma}_{e_y}\|_2^2 \quad (7.6a)$$

$$\text{s.t. } z_{j+1} = A_d(j)z_j + B_d(j)\kappa_j + h_d(j), \quad j = 0, \dots, N-1, \quad (7.6b)$$

$$z_0 = z(t), \quad (7.6c)$$

$$e_{y,j,\min} - \sigma_{e_y} \leq e_{y,j} \leq e_{y,j,\max} + \sigma_{e_y}, \quad j = 1, \dots, N, \quad (7.6d)$$

$$\kappa_{\min} \leq \kappa_j \leq \kappa_{\max}, \quad j = 1, \dots, N, \quad (7.6e)$$

$$c_{\min} \leq c_j \leq c_{\max}, \quad j = 1, \dots, N-1, \quad (7.6f)$$

$$\bar{\sigma}_{e_y} \geq 0, \quad (7.6g)$$

where  $\bar{\kappa} \in \mathbb{R}^{N+1}$  is the curvature vector to be optimized,  $\bar{\lambda} \in \mathbb{R}_+^{N \times N}$  is a diagonal matrix of the form  $\text{diag}(\lambda_1, \dots, \lambda_N)$  that penalizes constraint violations, and  $\alpha \in \mathbb{R}_+$  penalizes the curvature change rate. The cost function is introduced in Section 7.1.3. Note that the first element of  $\bar{\kappa}$  is equal to the previous optimal curvature inputted to the vehicle. The initial state  $z_0$  is given by the current state of the vehicle  $z(t)$  (i.e., the measurement at time  $t$  of the vehicle states), where  $e_y(t)$  and  $e_\psi(t)$  are computed using an orthogonal projection of the vehicle position into the reference path. The constraints (7.6d) to (7.6f) are thoroughly explained and discussed in Sections 7.1.1 and 7.1.2. To guarantee feasibility, constraints (7.6d) are softened by means of a slack variable  $\bar{\sigma}_{e_y} \in \mathbb{R}_+^N$ , which in turn is minimized in the cost function. When  $e_{y,j,\max} = e_{y,j,\min} = 0$ , deviations from the reference path are explicitly minimized in the cost function.

**Remark.** Although it is not used, the cost function (7.6a) can include a  $\ell_1$ -norm regularization term  $\|D_2\bar{\kappa}\|_1$  to induce sparsity in the obtained curvature commands, as explored in Chapter 4.

**Remark.** It is assumed that a full measurement or estimate of the state  $\mathbf{z}(t)$  is available at the current time  $t$ . However, this is not always the case for the real application, since time delays are almost unavoidable in the real system. In fact, the construction vehicle, in which the controller is deployed, presents a time delay (see Section 3.1). In our controller design, the delay present in the system is not explicitly accounted for. Nevertheless, to deal with it, the vehicle state after the time delay is estimated and used as the current state  $\mathbf{z}(t)$ .

### 7.1.5 Controller Parameters Influence

It is crucial to understand how the driving corridor width  $e_{y,\max}$  and the cost function parameters  $\bar{\lambda}$  and  $\alpha$  influence the control performance. The parameter  $\bar{\lambda}$  sets how accurate the path following is expected to be. The parameter  $\alpha$  sets the importance of maintaining the current constant curvature throughout the prediction horizon.

Figure 7.1 shows the effect of  $e_{y,\max}$  on the predicted path and vehicle curvature. The larger the corridor width is, the smoother the curvature function is. Note that the predicted path always respects the given driving corridor.

Figure 7.2 depicts the influence of  $\bar{\lambda}$  (constant over the prediction horizon) on the predicted path and vehicle curvature. The smaller  $\bar{\lambda}$  is, the smoother the curvature function is. However, the predicted vehicle path may violate the driving corridor constraints as  $\bar{\lambda}$  gets smaller. In the extreme case  $\bar{\lambda} = 0$ , the predicted vehicle curvature is equal to the current vehicle curvature throughout the prediction horizon.

Figure 7.3 demonstrates the impact of  $\alpha$  on the predicted path and vehicle curvature. The larger  $\alpha$  is, the more constant (i.e., tendentially constant) the predicted vehicle curvature function is. This is an intended property of the controller, since the curvature rate penalization term is included to regulate the amount of steering needed to follow the path. This way, it is possible to encourage small curvature changes. If  $\alpha$  is too large, the predicted path may violate the driving corridor constraints.

## 7.2 Experimental and Simulation Results

To demonstrate the effectiveness of the proposed control design, we perform experimental tests on a Scania construction truck. The performance of the SA-MPC is compared with a PPC and a standard MPC. The PPC and the standard MPC are introduced in Section 2.1.6 and Section 2.3.2.

To evaluate and validate the control design before deploying it in the experimental platform, we develop an accurate simulation environment that resembles

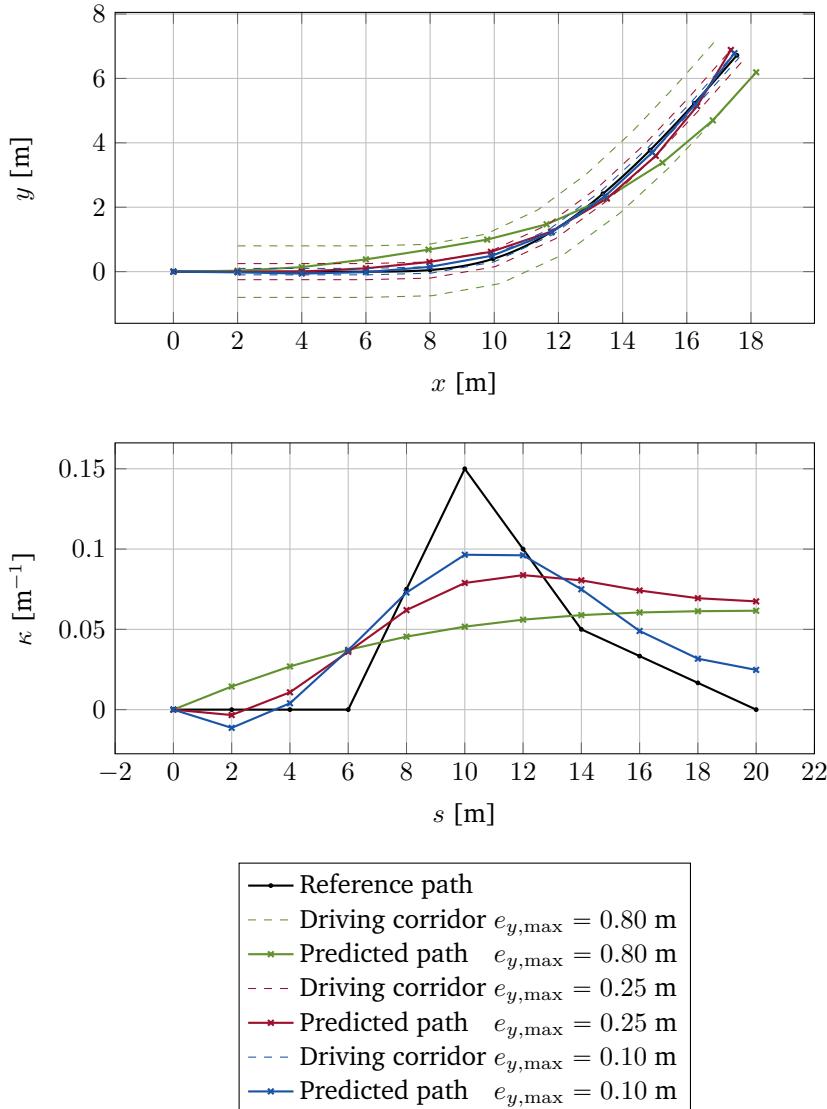


Figure 7.1: *Driving corridor width influence.* SA-MPC predicted path with  $N = 10$ ,  $\alpha = 0$ , and  $\lambda_i = 100$ , where  $i = 1, \dots, N$ .

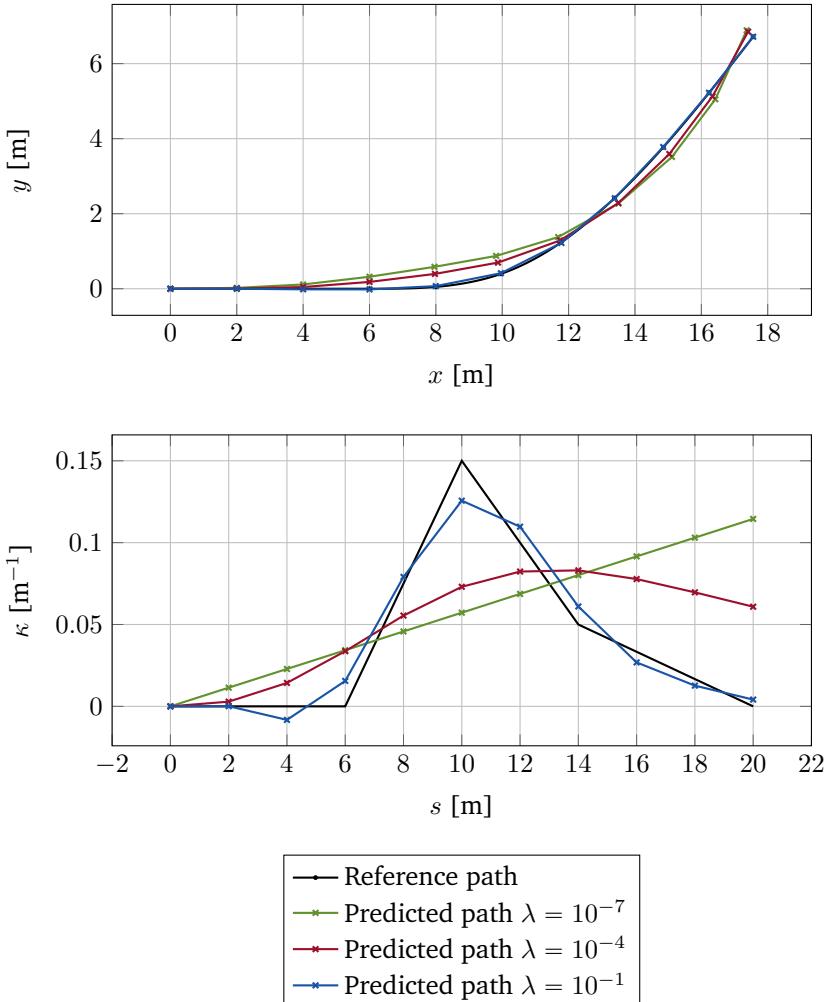


Figure 7.2: *Path following accuracy influence.* SA-MPC predicted path with  $N = 10$ ,  $e_{y,\max} = 0$  m, and  $\alpha = 0$ . In this case, it is assumed  $\lambda_1 = \lambda_2 = \dots = \lambda_N$ .

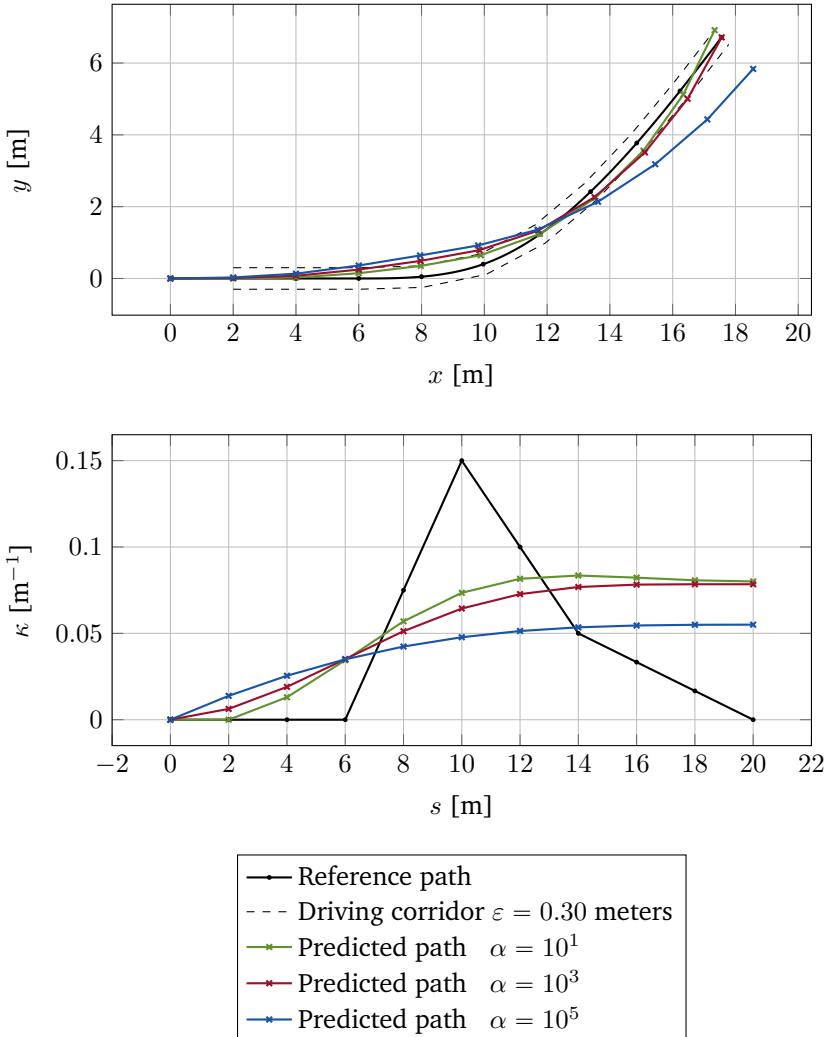


Figure 7.3: *Curvature rate penalization influence.* SA-MPC predicted path with  $N = 10$ ,  $e_{y,\max} = 0.3$  m, and  $\lambda_i = 1$ , where  $i = 1, \dots, N$ .

the real system. The lateral dynamics of the simulation model are described by a 4-axles nonlinear bicycle model with 2 steering axles in the front and 2 traction axles in the back, as described in Section 3.1. It is based on the modified Scania G480 construction truck shown in Figure 3.1. We analyze the performance of the controllers in terms of path following accuracy and driving smoothness and discuss the simulation environment reliability as a test environment.

The main idea behind the experimental controller benchmarking is to determine the quality of the designed controller, compared to standard approaches. The SA-MPC formulation intends to provide driving smoothness. Therefore, the SA-MPC is compared with a standard MPC to understand the effect that the smoothing term has on path tracking accuracy. Finally, both MPC approaches are compared with a PPC. Although being a simple controller, the PPC is widely used in industry due to its ease of implementation and satisfactory results.

### 7.2.1 Reference Paths

As in Chapter 4, the reference paths are obtained using real data from Scania test track facilities located in Södertälje, Sweden, south of Stockholm (see Figure 7.4). The reference paths consist of a waypoint sequence, two meters apart from each other<sup>1</sup>. The *precision test track* is a challenging gravel road designed to perform tests with autonomous mining trucks, since it resembles a mining site. It is a narrow road with an approximate length of 1.5 km. It has sharp turns where consequently, the vehicle speed cannot be higher than 50 km/h. The curvature in these turns typically exceeds  $0.05 \text{ m}^{-1}$ . Therefore, it is an appropriate scenario to evaluate the performance of control algorithms designed to be accurate at low speeds. In addition, a *high-speed test track* consists of two long straights and two sharp U-turns (one with curvature  $0.04 \text{ m}^{-1}$  and the other with  $0.05 \text{ m}^{-1}$ ). On one of the straights the speed limit is 70 km/h, while on the other straight the speed limit is 90 km/h.

### 7.2.2 Simulation Results

Experimental evaluation requires some level of prior controller testing and validation. Therefore, we set up a simulation environment in MATLAB/Simulink resembling, as much as possible, the real system with the same modules and interfaces as those present in the vehicle. The results presented in this section demonstrate that the simulation environment is sufficiently accurate to allow, for example, controller tuning before deploying it on the vehicle. Also, in simulation we can extend our analysis further than experimentally, since we are not constrained by implementation issues and platform availability. In the simulation environment, we compare the PPC, the standard MPC, and the SA-MPC.

The simulation results are shown in Figures 7.5, 7.6, 7.7, and 7.8. In the box plots depicted in some of these figures, the central mark is the median, the edges

---

<sup>1</sup>We do not use the clothoid-based path sparsification proposed in Chapter 4.

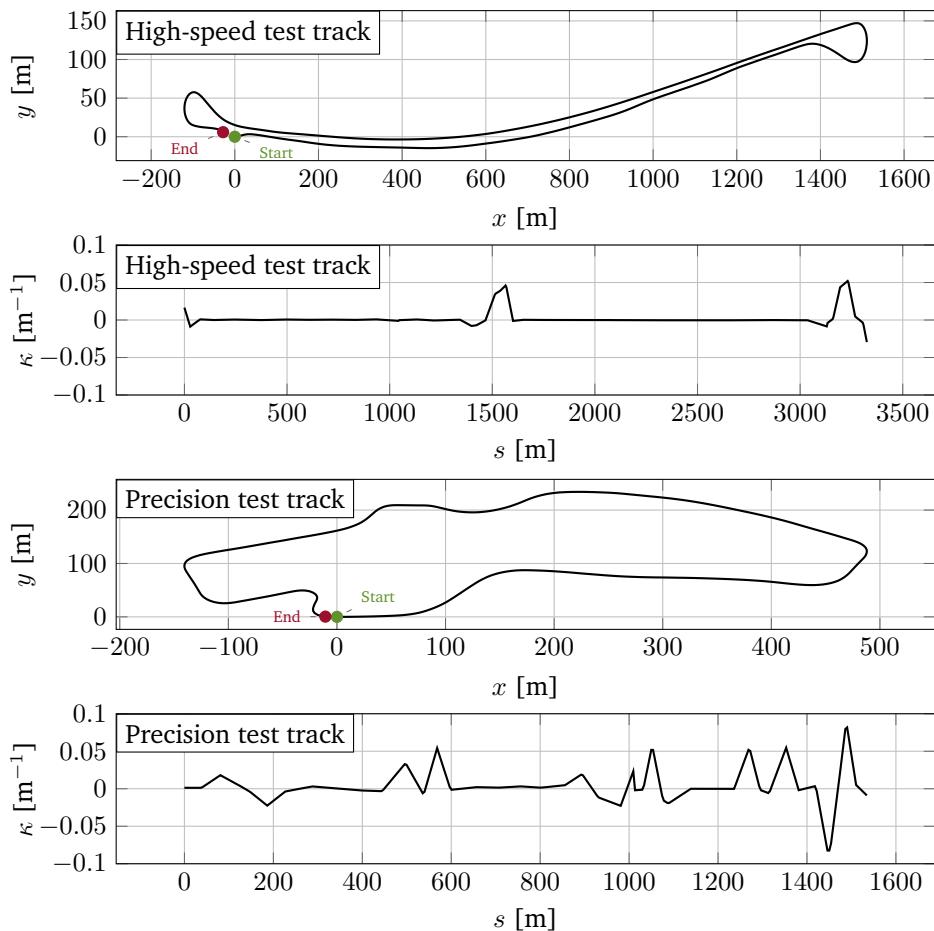


Figure 7.4: Reference paths used in experimental evaluation. The two upper plots represent the high-speed test track reference path and curvature. The two lower plots precision test track reference path and curvature.

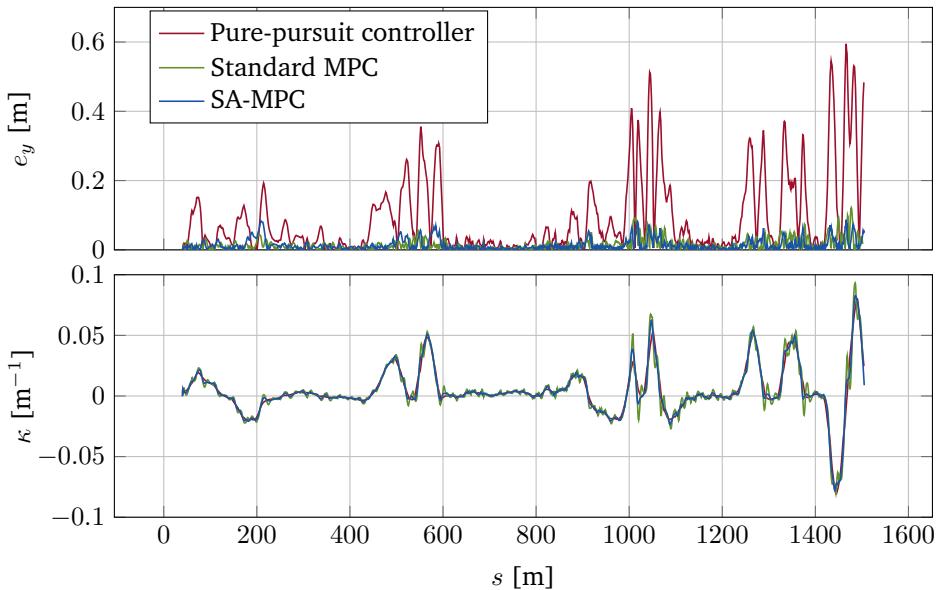


Figure 7.5: *Precision test track simulation*. The upper plot depicts the lateral deviation from the path. The lower plot shows the vehicle curvature.

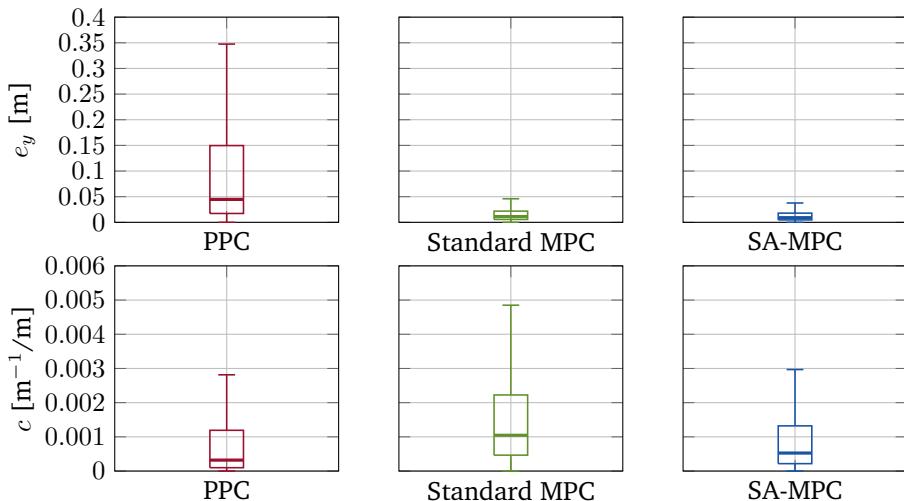


Figure 7.6: *Precision test track simulation*. Statistical analysis of the lateral deviation from the path and the curvature change rate.

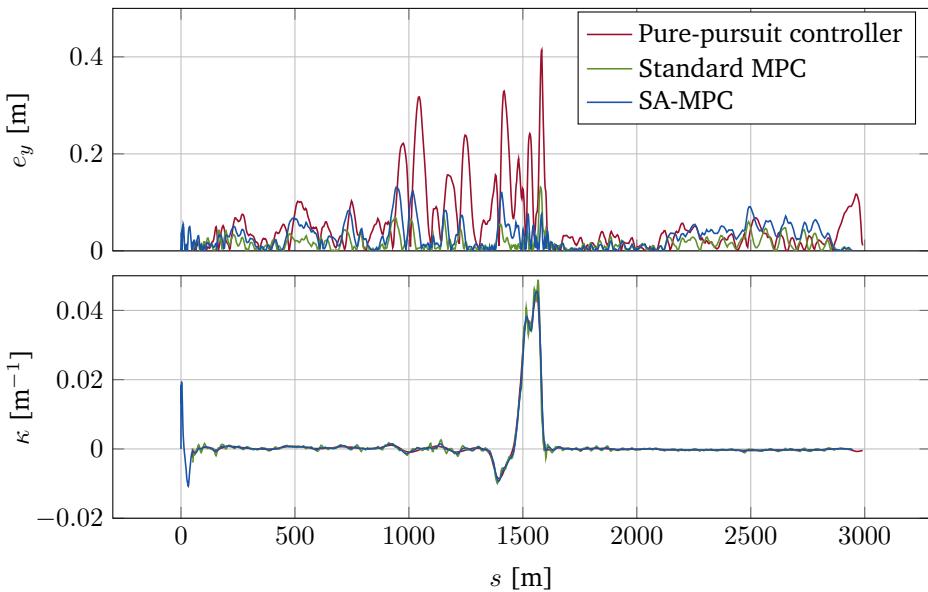


Figure 7.7: *High-speed test track simulation*. The upper plot depicts the lateral deviation from the path. The lower plot shows the vehicle curvature.

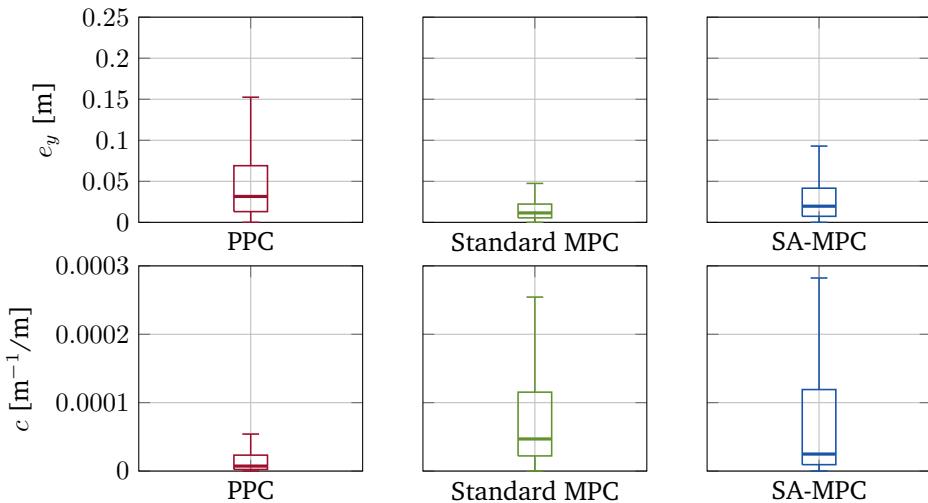


Figure 7.8: *High-speed test track simulation*. Statistical analysis of the lateral deviation from the path and the curvature change rate.

Parameters	PPC	MPC	SA-MPC
Look ahead time	1.2 s		
Prediction horizon, $N$		10	10
Model discretization time step, $T_s$		200 ms	200 ms
Controller frequency	50 Hz	50 Hz	50 Hz
State penalization matrix, $Q$		diag(50, 50, 0.1)	
Input penalization matrix, $R$		500	
Driving accuracy, $\lambda_i$			200
Curvature rate penalization, $\alpha$			200
Driving corridor width, $e_{y,\max}$			0

Table 7.1: Controllers parameters for simulation evaluation.

		Max. [m]	Avg. [m]	Std. dev. [m]
Precision test track	PPC	0.59	0.10	0.12
	MPC	0.12	0.02	0.02
	SA-MPC	0.09	0.02	0.02
High-speed test track	PPC	0.41	0.06	0.08
	MPC	0.13	0.02	0.02
	SA-MPC	0.13	0.03	0.02

Table 7.2: Lateral deviation from the path in simulation.

of the box are the 25<sup>th</sup> and 75<sup>th</sup> percentiles and the whiskers extend to the most extreme data points not considered outliers. Table 7.1 summarizes the controllers parameters values for each controller. The simulation results are summarized in Table 7.2.

On the precision test track, the vehicle speed is typically between 4 m/s and 10 m/s, while on the high-speed test track the speed can reach 25 m/s on one of the straights. In this case, both the standard MPC and SA-MPC approaches outperform the PPC. To be able to draw conclusions about the performance of the controllers, we also take into account the curvature change rate data shown in Figures 7.6 and 7.8. Both SA-MPC and the standard MPC have similar and good performance, since both have a low average deviation and a maximum lateral deviation from the path below 20 cm. However, the standard MPC approach is more aggressive comparing the vehicle curvature change rate. On both scenarios PPC is the smoothest controller. In contrast, the standard MPC approach is the most aggressive, but also the most accurate. The PPC achieves a smoother control, but at the cost of a significant loss of accuracy.

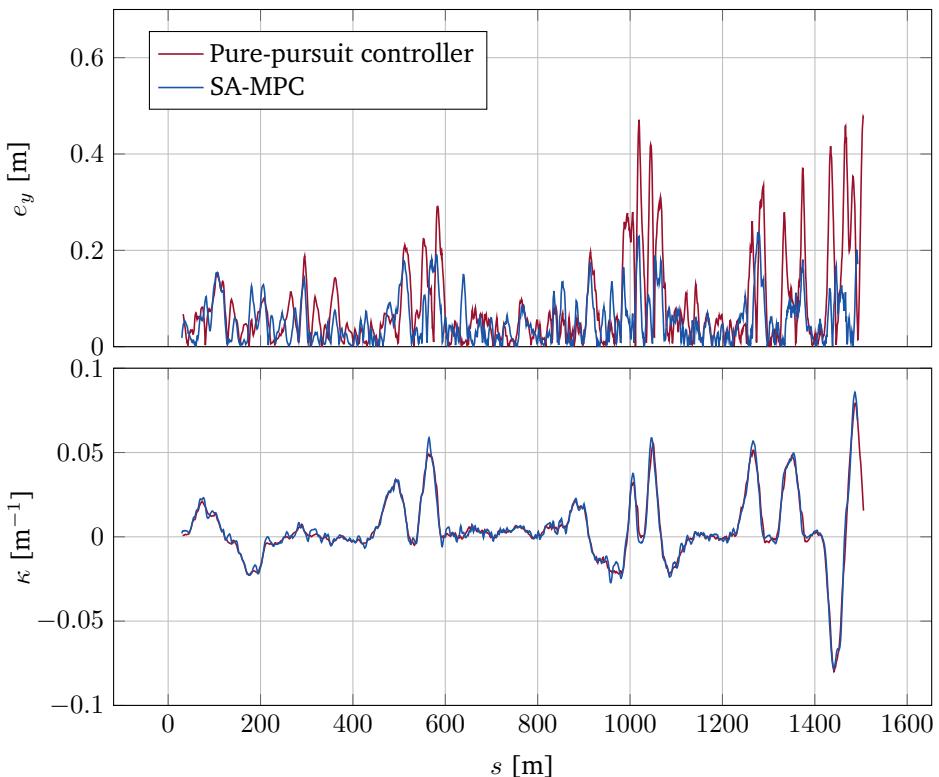


Figure 7.9: *Precision test track experiment*. The upper plot depicts the lateral deviation from the path. The lower plot shows the vehicle curvature.

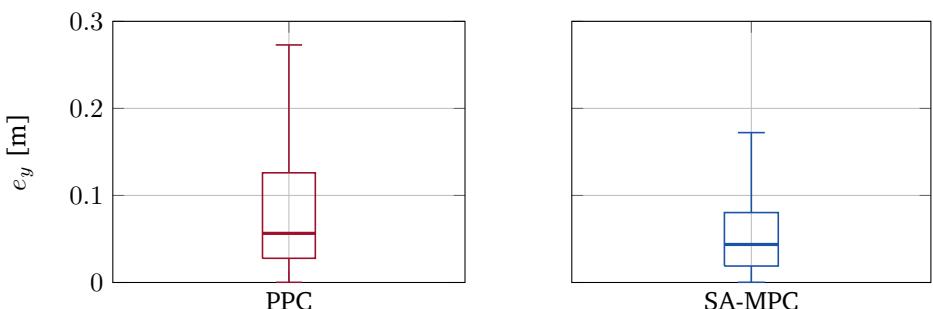


Figure 7.10: *Precision test track experiment*. Statistical analysis of the lateral deviation from the path.

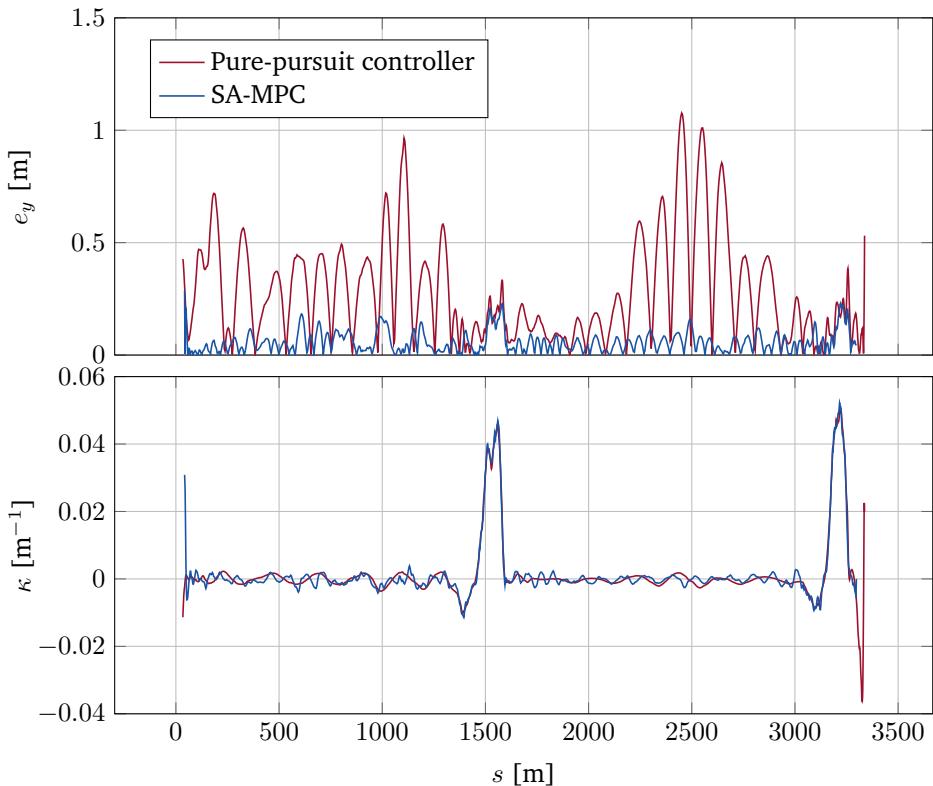


Figure 7.11: *High-speed test track test track experiment.* The upper plot depicts the lateral deviation from the path. The lower plot shows the vehicle curvature.

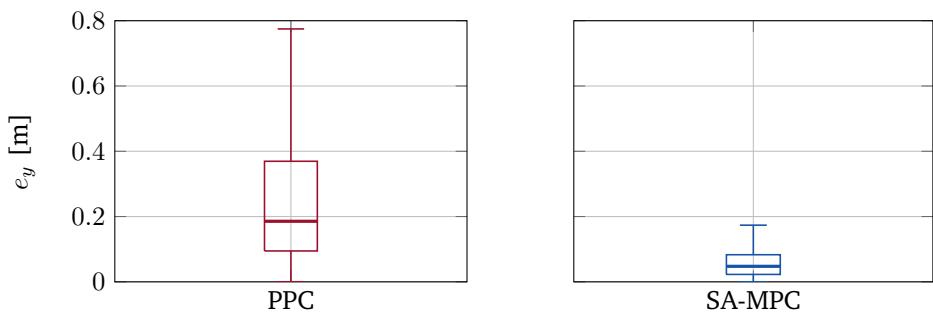


Figure 7.12: *High-speed test track experiment.* Statistical analysis of the lateral deviation from the path.

Parameters	PPC	SA-MPC
Look ahead time	1.2 s	
Prediction horizon, $N$		10
Model discretization time step, $T_s$		200 ms
Controller frequency	50 Hz	50 Hz
Driving accuracy, $\lambda_i$		20
Curvature rate penalization, $\alpha$		20
Driving corridor width, $e_{y,\max}$		0

Table 7.3: Controllers parameters for experimental evaluation.

		Max. [m]	Avg. [m]	Std. dev. [m]
Precision test track	PPC	0.48	0.10	0.10
	SA-MPC	0.24	0.06	0.05
High-speed test track	PPC	1.08	0.25	0.21
	SA-MPC	0.29	0.06	0.05

Table 7.4: Lateral deviation from the path in real experiments.

### 7.2.3 Experimental Results

The truck used in the experimental evaluation is a modified Scania G480 construction truck and is shown in Figure 3.1. The vehicle is equipped with a sensor platform and a servo motor for automated control of the steering column. A detailed description of the system architecture is given in Section 2.1. To perform these experimental tests, the waypoints of the reference paths are added to a database that is available to be queried at run time (i.e., an algorithm constantly determines if the vehicle is near a reference path and feeds the waypoints sequentially to the controller). The reference path can also be provided by a path planner. In this chapter, the reference paths are fixed to allow for a proper comparison between the controllers. We choose to implement the SA-MPC on the truck, as it is the controller with the best performance in the simulation environment, and PPC was already deployed and provided a good benchmark.

The experimental results are shown in Figures 7.9, 7.10, 7.11, and 7.12. The results are summarized in Table 7.4. Table 7.3 contains the controllers parameters used in the experiments. Videos of the experiments are available on Zenodo (Lima et al., 2016a,b).

In both scenarios, it is clear that the SA-MPC outperforms the PPC. Also, the performance of the SA-MPC on the high-speed test track is similar to the one expected from simulation results, which supports our claim of having a realistic simulation environment that can be used prior to experimental evaluation. In contrast, note that the PPC performs quite differently at high speeds in reality when compared with the simulation, due to the neglected high-speed dynamics in the simulation

vehicle model. Also, it means that the PPC appears to be less robust to these dynamics, when compared with the SA-MPC. At low-speeds, the performance of both controllers is good with an average lateral deviation from the path of less than 10 cm. Still, the SA-MPC outperforms the PPC, especially in sharp turns. Note that the SA-MPC is able to run at both high and low speeds with the same parameter configuration.

Both in simulation and experiment, the absence of driving corridor (i.e.,  $e_{y,\max} = e_{y,\min} = 0$ ) means that the SA-MPC only differs from a standard MPC in the cost function, where the first- and second-order derivative of the input are also penalized. For controller performance benchmarking, the best path tracking accuracy possible is sought to compare driving behaviors. There is a clear trade-off between accuracy and smoothness, which is challenging to balance. Nevertheless, SA-MPC is able to effectively combine smooth control with an accurate path following.

### 7.3 Summary

In this chapter, we proposed an SA-MPC for autonomous driving. The first- and second-order spatial derivatives of the vehicle curvature are explicitly considered in the SA-MPC cost function. Consequently, the vehicle curvature is encouraged to be PWA as a function of space in order to provide smooth driving.

The SA-MPC formulation includes input and state constraints. The state constraints define a driving corridor where the vehicle should remain (e.g., lane boundaries). Nevertheless, these constraints are softened to ensure optimization feasibility. Moreover, we discussed the influence of the controller parameter through illustrative simulation examples.

Finally, simulation and experimental results using the SA-MPC were presented. We compared the proposed controller to existing approaches, such as a PPC and a standard MPC. The performance was measured in terms of lateral deviation from the path and curvature change rate (i.e., driving smoothness). In our experiments and simulations, the PPC was the least accurate controller. The SA-MPC and the standard MPC were the most accurate controllers. Analyzing the curvature change rate provided by the controllers, we concluded that the SA-MPC is smoother than the standard MPC with a similar path accuracy. The SA-MPC formulation plays a crucial role in the driving smoothness, since, unlike a standard MPC, it encourages smooth curvature predictions.

Experimentally, we compared the SA-MPC with a PPC, both deployed in a Scania construction truck. As predicted from the simulations, the SA-MPC clearly outperformed the PPC. Even using a linearized kinematic vehicle to predict the vehicle motion, the performance of the controller is very promising, since the maximum deviation from the path never exceeds 30 cm and on average is 6 cm. Also, we concluded that tuning the controllers parameters using the simulation environment provided a good estimate of the actual controller performance.



## Chapter 8

# Stability of Linear Time-Varying Model Predictive Controller

Autonomous vehicles will inevitably face emergency situations, in which they may need to maneuver aggressively to avoid, for example, an imminent collision. Thus, when removing the human-factor from the equation, ensuring vehicle stability during safety-critical events is of utmost importance when developing commercial autonomous vehicles. Verified stability is a key aspect for safe and reliable autonomous vehicles.

ADAS play a major role of support to the driver both in critical and tedious situations, reducing the number of traffic accidents and fatalities. Nevertheless, the design of the motion controller module for autonomous vehicles must attain a stable behavior and cannot rely on an eventual unstable behavior being avoided by driving assistance technology. In this chapter, we address the problem of LTV-MPC design in order to ensure closed-loop stability. On the one hand, ensuring vehicle stability facilitates controller certification and standardization when entering the development phase. On the other hand, shortening the prediction horizon gives room to more computational demanding modules.

One of the strengths of MPC is the possibility of explicitly include additional constraints and cost terms, which lead to closed-loop stability guarantees. One of the most popular strategies for ensuring closed-loop stability using MPC (see Mayne et al. (2000) and references therein) is to use the optimization value function as a Lyapunov function. Moreover, the analysis is convenient if incorporating both a terminal cost and a terminal state set in the optimal control problem. The terminal cost is chosen such that it is equal to the infinite-horizon value function in a suitable neighborhood of the origin (i.e., the terminal state set). Hence, it is possible to use the known advantages of an infinite-horizon control, such as guaranteed stability.

The outline of this chapter is as follows. In Section 8.1, we recall the problem of reference tracking using an LTV-MPC controller, and present the nonlinear space-based vehicle model in the road-aligned coordinate frame used in the autonomous

driving example. In Section 8.2, we propose the terminal cost and terminal state set, which are used for proving LTV-MPC feasibility and closed-loop stability in Section 8.3. In Section 8.4, we demonstrate the effectiveness of the inclusion of the terminal cost and terminal state set in the MPC design through simulations and experiments using a Scania truck. In Section 8.5, we summarize the chapter and provide some concluding remarks.

## 8.1 Problem Formulation

Consider the problem of controlling the discrete-time nonlinear time-varying system (2.5) with constraints (2.6) to track a given time-varying reference. Consider the reference state  $z_r(k) \in \mathcal{Z} \in \mathbb{R}^n$  and the input  $u_r(k) \in \mathcal{U} \in \mathbb{R}^m$ . Let  $\bar{z}_r(k) = \{z_r(k)\}_{k=0}^{N-1}$  be the reference path obtained by applying an input sequence  $\bar{u}_r(k) = \{u_r(k)\}_{k=0}^{N-1}$  to system (2.5). Then, the first-order Taylor approximation and forward Euler discretization of (2.5) around  $\bar{z}_r(k)$  and  $\bar{u}_r(k)$  yields the linear time-varying (or linear space-varying) system (3.11), which can be written in the following form

$$\tilde{z}(k+1) = A_d(\xi(k))\tilde{z}(k) + B_d(\xi(k))\tilde{u}(k), \quad (8.1)$$

where  $\tilde{z}(k) = z(k) - z_r(k)$ ,  $\tilde{u}(k) = u(k) - u_r(k)$ , and  $\xi(k) = [z_r(k)^T, u_r(k)^T]^T$  is a parameter vector known at every step  $k$ . Note that in (8.1) we write matrices  $A_d(k)$  and  $B_d(k)$  appearing in (3.11) explicitly as a function of a parameter  $\xi(k)$ . Also,  $\xi(k) \in \Xi, \forall k$ , where  $\Xi$  is a closed set containing all the possible values of  $\xi$  such that

$$\Xi = \{\xi \in \mathbb{R}^{n+m} : \xi_{\min} \leq \xi \leq \xi_{\max}\},$$

where the inequality operations are considered element-wise. The LTV system (8.1) is a first-order approximation of the nonlinear system (2.5). It describes the deviations from the state trajectory  $z_r(k)$  when a sequence of inputs  $u_r(k)$  is applied to the system.

Let  $\gamma \in \Gamma$  represent a model described by a specific pair  $(A_d(\xi), B_d(\xi))$ , where the set  $\Gamma$  is a multi-plant description defined as

$$\Gamma = \{(A_d, B_d) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m} : A_d = A_d(\xi), B_d = B_d(\xi), \xi \in \Xi\}. \quad (8.2)$$

Note that each  $\gamma \in \Gamma$  is time-invariant and depends on a (known) parameter  $\xi$ .

### 8.1.1 Vehicle Model

Consider the space-based road-aligned model (3.8) presented in Section 3.2, which describes the lateral displacement  $e_y$  and heading displacement  $e_\psi$  between the vehicle and a given reference path. Let the state vector be  $z = [e_y, e_\psi]^T$ , the control input be  $u = \kappa$  and (3.8) be described by  $z' = f(z, u)$ . The reference path

is discretized such that  $\{s_j\}_{j=0}^N = \{s_0, s_1, \dots, s_N\}$ . The nonlinear model (3.8) is linearized and discretized using a first-order Taylor approximation and then apply forward Euler discretization, around a reference path  $z_r(k) = [e_{\psi,r}(k), e_{y,r}(k)]^T = [0, 0]^T$ ,  $k \geq 0$  given by a reference sequence of inputs  $\bar{u}_r(k)$ . Thus, we obtain an LTV model of the form (8.1)

$$\begin{bmatrix} e_y(k+1) \\ e_\psi(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta_s \\ -\kappa_r^2(k)\Delta_s & 1 \end{bmatrix} \begin{bmatrix} e_y(k) \\ e_\psi(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta_s \end{bmatrix} \tilde{\kappa}(k), \quad (8.3)$$

where it is assumed that the reference input curvature  $\kappa_r = \kappa_s = \frac{1}{\rho_s}$ . The parameter  $\xi(k)$  is, in this particular case, only dependent on the road curvature  $\kappa_r(k)$ . In fact,  $\Delta_s = T_s v_x$  could also be seen as a parameter, but in the following analysis it is set as constant (i.e., the vehicle travels with constant speed). Consequently, in the remainder of the chapter and for the sake of simplicity,  $\xi(k) = \kappa_r(k)$ .

Furthermore, vehicle curvature and curvature rate are limited by constant bounds

$$\begin{aligned} \kappa_{\min} &\leq \kappa_k \leq \kappa_{\max}, \quad k = 0, \dots, N-1, \\ c_{\min} &\leq c_k \leq c_{\max}, \quad k = 0, \dots, N-1, \end{aligned}$$

where  $c_k = \frac{\kappa_{k+1} - \kappa_k}{\Delta_s}$ . The longitudinal speed  $v_x$  is assumed constant throughout the prediction horizon due to the slow longitudinal dynamics of a heavy-duty vehicle.

### 8.1.2 Linear Time-Varying Model Predictive Control

Assume that a full measurement or estimate of the state  $z(t)$  is available at the current time  $t$ . Then, the following LTV-MPC problem with a prediction horizon of  $N$  points can be formulated

$$\min_{\bar{u}_t} \quad \tilde{z}_{N|t}^T Q_f \tilde{z}_{N|t} + \sum_{k=0}^{N-1} \tilde{z}_{k|t}^T Q \tilde{z}_{k|t} + \tilde{u}_{k|t}^T R \tilde{u}_{k|t} \quad (8.4a)$$

$$\text{subject to} \quad \tilde{z}_{k+1|t} = A_d(\xi(k|t)) \tilde{z}_{k|t} + B_d(\xi(k|t)) \tilde{u}_{k|t}, \quad k = 0, \dots, N-1, \quad (8.4b)$$

$$\tilde{z}_{k|t} \in \tilde{\mathcal{Z}}, \quad k = 0, \dots, N-1, \quad (8.4c)$$

$$\tilde{u}_{k|t} \in \mathcal{U}, \quad k = 0, \dots, N-1, \quad (8.4d)$$

$$\tilde{z}_{N|t} \in \tilde{\mathcal{Z}}_f, \quad (8.4e)$$

$$\tilde{z}_{0|t} = \tilde{z}(t), \quad (8.4f)$$

where  $\bar{u}_t = \{\bar{u}_{k|t}\}_{k=0}^{N-1}$  is the sequence of inputs to be optimized, which are constrained to be in a convex polytope  $\mathcal{U}$ . The difference between the state vector at time  $t+k$  predicted at time  $t$  and the reference state vector  $z_r(t+k)$  is defined by  $\tilde{z}_{k|t}$ . The difference between the predicted and the reference state is constrained

to be in a convex polytope  $\tilde{\mathcal{Z}}$ . The notation  $z_{k|t}$  stands for the state  $z$  at time  $t + k$  predicted at time  $t$ . The notation is analogous for  $u_{k|t}$ . Note that  $z(t)$  is then the actual state  $z$  at time  $t$ . The matrices  $R$ ,  $Q$ , and  $Q_f$  are positive definite and penalize deviations from the reference input, state, and terminal state, respectively.

**Remark.** *It is assumed that a full measurement or estimate of the state  $\mathbf{z}(t)$  is available at the current time  $t$ . However, this is not always the case for the real application, since time delays are almost unavoidable in the real system. In fact, the construction vehicle, in which the controller is deployed, presents a time delay (see Section 3.1). In our controller design, the delay present in the system is not explicitly accounted for. Nevertheless, to deal with it, the vehicle state after the time delay is estimated and used as the current state  $\mathbf{z}(t)$ .*

Let  $\bar{u}_t^* = \{u_{k|t}^*\}_{k=0}^{N-1}$  be the optimal solution of (8.4) at time  $t$ . The first element of  $\bar{u}_t^*$  is applied to system (8.1) at time  $t$ . In the next sampling time, the optimal control problem (8.4) is solved again using the new state measurements, and the horizon is shifted forward (i.e., receding-horizon control).

In receding-horizon control, the optimization problem may lead us into an infeasible situation (i.e., there does not exist a sequence of control inputs for which the constraints are satisfied). Also, even if the optimization problem is always feasible, the computed optimal control inputs may not lead to an asymptotically stable closed-loop system. In general, feasibility and closed-loop stability are not ensured in (8.4).

This chapter addresses the problem of how to choose the terminal weight  $Q_f$  and the terminal constraint set  $\tilde{\mathcal{Z}}_f$  such that closed-loop stability and feasibility are ensured for all  $\gamma \in \Gamma$ . Conditions on how the terminal weight  $Q_f$  and the terminal constraint set  $\tilde{\mathcal{Z}}_f$  should be chosen such that closed-loop stability and feasibility are ensured have been extensively studied in the past three decades (Mayne et al., 2000). The main idea is to find a feedback control law  $F(k)$  that stabilizes the unconstrained system (8.1) inside an invariant terminal region  $\tilde{\mathcal{Z}}_f$ . Then, the closed-loop is stable if the cost function (8.4a) accounts for the infinite-horizon cost. The finite-horizon cost (when the controller is constrained) is added to the infinite-horizon cost (when the controller is unconstrained (i.e., when the system enters  $\tilde{\mathcal{Z}}_f$ ). Then, for an unconstrained linear system the convergence to the origin is exponential (Borrelli et al., 2017).

## 8.2 Model Predictive Control Design for Autonomous Driving

The LTV-MPC (8.4) is used to control a nonholonomic vehicle to a given reference using a linearized and discretized space-based kinematic vehicle model (8.3). Knowing the reference curvature  $\kappa_r(k)$  *a priori* allows the definition of the set  $\Xi$ , where  $\xi_{\min} = \kappa_{r,\min}$  and  $\xi_{\max} = \kappa_{r,\max}$ . Imposing the terminal state to lie inside the terminal constraint set  $\tilde{\mathcal{Z}}_f$  and with a proper choice of  $Q_f$ , closed-loop stability and feasibility can be ensured for all  $\gamma \in \Gamma$ .

### 8.2.1 Computation of the Terminal Constraint $\tilde{\mathcal{Z}}_f$

When the model is LTI, a typical choice for  $\tilde{\mathcal{Z}}_f$  is the maximal positive invariant set  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$  for the closed-loop system

$$\tilde{z}(k+1) = \left( A_d(\xi(k)) + B_d(\xi(k))L_{\text{LQR}}(\xi(k)) \right) \tilde{z}(k), \quad (8.5)$$

where  $L_{\text{LQR}}$  is the associated LQR gain (i.e., the unconstrained infinite time optimal controller gain).

However, since our model is LTV, there are several maximal positive invariant sets  $\bar{\mathcal{O}}_\infty^{\text{LQR}}(\gamma)$  and several different LQR feedback controllers  $L_{\text{LQR}}(\xi)$ , one for each  $\xi \in \Xi$ . Therefore, the maximal positive invariant set  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$  that is invariant for all  $\gamma \in \Gamma$  is sought (i.e.,  $\bar{\mathcal{O}}_\infty^{\text{LQR}} \subseteq \text{Pre}_\gamma(\bar{\mathcal{O}}_\infty^{\text{LQR}})$ ,  $\forall \gamma \in \Gamma$ ). The notation  $\text{Pre}_\gamma(\cdot)$  denotes the one-step controllable set (see Definition 2.5) using a specific model  $\gamma$ .

Recall the geometric condition for invariance expressed in Theorem 2.1 and let  $\Gamma_d$  be a discretized version of the multi-plant  $\Gamma$  with a finite number of LTI models. Then, the following recursion

$$\Omega_{k+1} = \bigcap_{\gamma \in \Gamma_d} \text{Pre}_\gamma(\Omega_k) \cap \Omega_k, \quad \Omega_0 = \tilde{\mathcal{Z}}, \quad (8.6)$$

eventually converges to the maximal positive invariant set (see Definition 2.8) (Kolmanovsky and Gilbert, 1998), i.e.,

$$\bar{\mathcal{O}}_\infty^{\text{LQR}} = \lim_{k \rightarrow \infty} \Omega_k.$$

Note that, the matrix  $A_d(\xi(k))$  of model (8.3) can be described as belonging to a convex uncertainty polytope

$$\Lambda = \{A_d \in \mathbb{R}^{2 \times 2} : A_d = \sum_{i=1}^2 \lambda_i A_{d,i}, \sum_{i=1}^2 \lambda_i = 1, \lambda_i \geq 0\}.$$

Therefore, it is possible to show that a polyhedral robust invariant set (i.e., robust with respect to the varying parameter) can be constructed only using the vertices of the uncertainty polytope (Blanchini, 1999; Pluymers et al., 2005). In other words, if  $\Gamma_d$  includes, at least, the vertices of  $A_d(\xi(k))$ , then (8.6) computes the maximum invariant set for  $\Gamma$ .

All the operations with sets are performed using the multiparametric toolbox (MPT) for MATLAB (Kvasnica et al., 2004).

#### Computation of $\bar{\mathcal{O}}_\infty^{\text{LQR}}$ Without Input Rate Constraints

Figure 8.1(a) depicts an example of the computation of a  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$  for model (8.3), where the input is saturated outside  $-0.18$  and  $0.18 \text{ m}^{-1}$ , which are the curvature limits of the construction truck later considered in the experimental evaluation. The

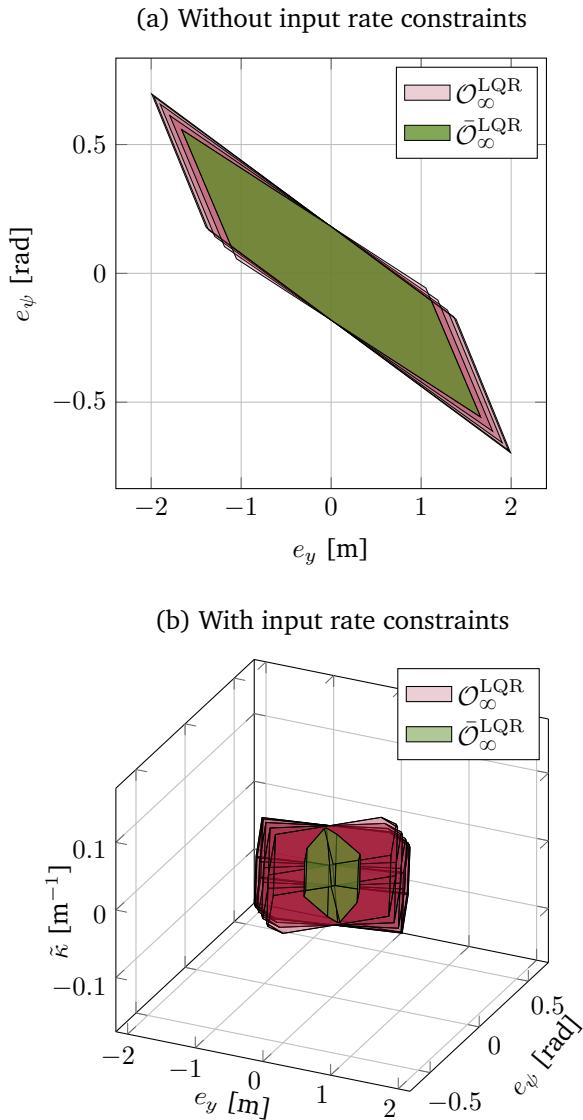


Figure 8.1: Maximal positive invariant set for each (red) LTI vehicle model and for all (green) LTI vehicle models.

parameters used are  $|\kappa_r| \leq 0.18 \text{ m}^{-1}$ ,  $Q = I$ ,  $R = 1$ , and  $\Delta s = 1 \text{ m}$ . The figure also shows some  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$  for different  $\kappa_r$ . The geometric interpretation of the computed set is intuitive. If the vehicle is on the left of the path (positive  $e_y$ ), the heading displacement  $e_\psi$  tends to be negative such that the vehicle tends to point inwards to the path, and vice-versa. For the model (8.3), in average, each LTI invariant set takes around 0.2 seconds to compute and the LTV invariant set takes around 1 second.

Since  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$  is a convex polytope set, the constraint (8.4e) can be written as

$$H\tilde{z}_{N|t} \leq h,$$

where  $H \in \mathbb{R}^{2 \times h_n}$  and  $h \in \mathbb{R}^{h_n}$ , where  $h_n$  represents the number of planes that define the set.

### Computation of $\bar{\mathcal{O}}_\infty^{\text{LQR}}$ With Input Rate Constraints

Figure 8.1(b) depicts an example of the computation of a  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$  for model (8.3), when the input rate constraints are considered. The LTI invariant sets  $\mathcal{O}_\infty^{\text{LQR}}$  and the LTV invariant sets  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$  also depend on the input curvature  $\kappa$ . The input rate is saturated outside  $0.05$  and  $0.05 \text{ m}^{-1}/\text{s}$ , and the parameters used are the same as before. Again, if the vehicle is on the left of the path (positive  $e_y$ ), the heading displacement  $e_\psi$  tends to be negative, and the curvature  $\kappa$  tends to be positive such that the vehicle always curves towards the path, and vice-versa. For model (8.3), in average, each LTI invariant set takes around 0.3 seconds to compute and the LTV invariant set takes 1.5 seconds. So, it seems that the computational time of the invariant sets is linearly proportional to the number of states used.

In this case,  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$  is also a convex set, but with one extra dimension. So the constraint (8.4e) can be written as

$$H\tilde{z}_{N|t} + g^T \tilde{u}_{N-1|t} \leq h,$$

where  $H \in \mathbb{R}^{2 \times h_n}$ ,  $g \in \mathbb{R}^{h_n}$  and  $h \in \mathbb{R}^{h_n}$ , where  $h_n$  represents the number of hyperplanes that define the set.

### Influence of the Penalization Matrices $Q$ and $R$

Figures 8.2 and 8.3 show the influence of  $R$  and  $Q$  on  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$ . The larger the deviations from the reference input (reflected in  $R$ ) are penalized, the larger the LTV invariant set  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$ . A large terminal set is desired, since the convergence region, in which stability can be ensured, is larger; however, it is not necessary to ensure stability in cases where the vehicle is too far away from the reference. The higher the cost of being far from the reference (i.e., increasing  $Q_{11}$ , the first diagonal element of  $Q$ ), the smaller the invariant set. This is expected, since if the lateral deviation from the reference is highly penalized, the controller becomes too aggressive, resulting in a smaller invariant set and, consequently, in a smaller attraction region,

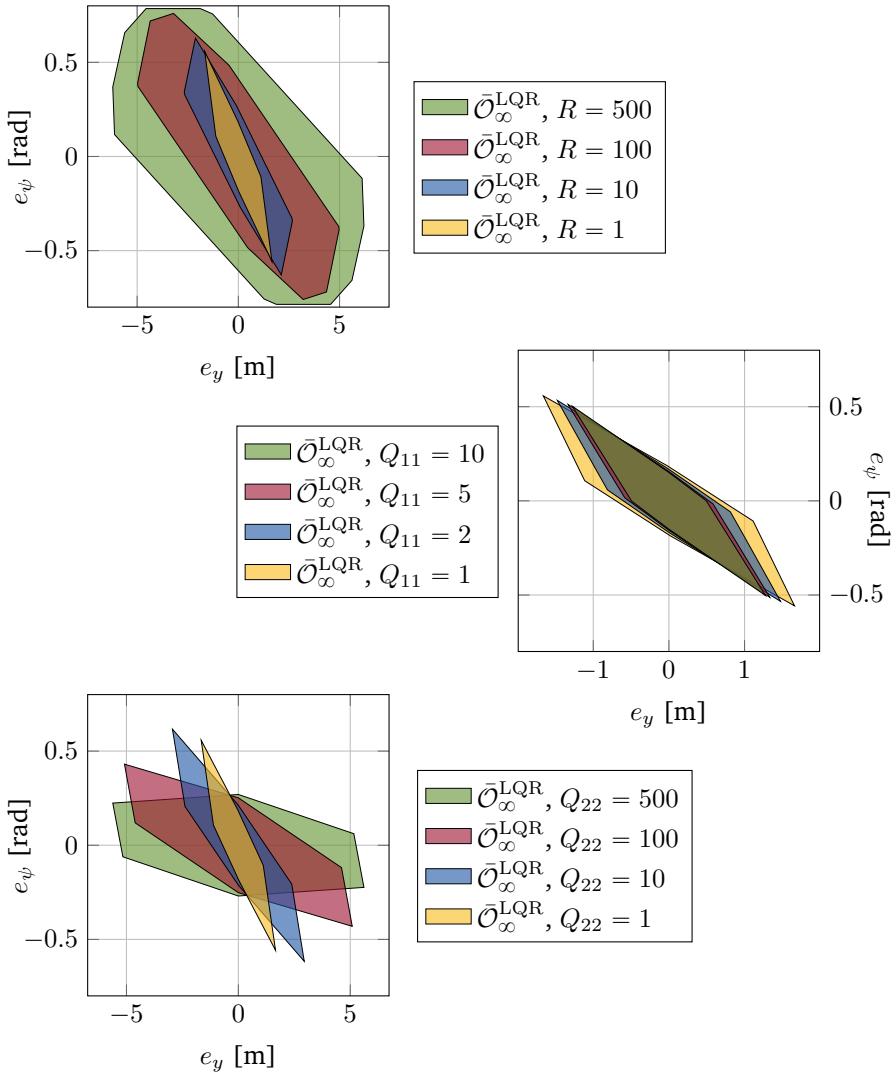


Figure 8.2: Maximum positive invariant set for all  $\gamma \in \Gamma$  ignoring input rate constraints. At the top, the input penalization matrix  $R$ , which is related to the curvature  $\kappa$ , is varied. In the middle, the first element of the state penalization matrix  $Q$ , which is related to the lateral displacement  $e_y$ , is varied. At the bottom, the second element of the state penalization matrix  $Q$ , which is related to the heading displacement  $e_\psi$ , is varied.

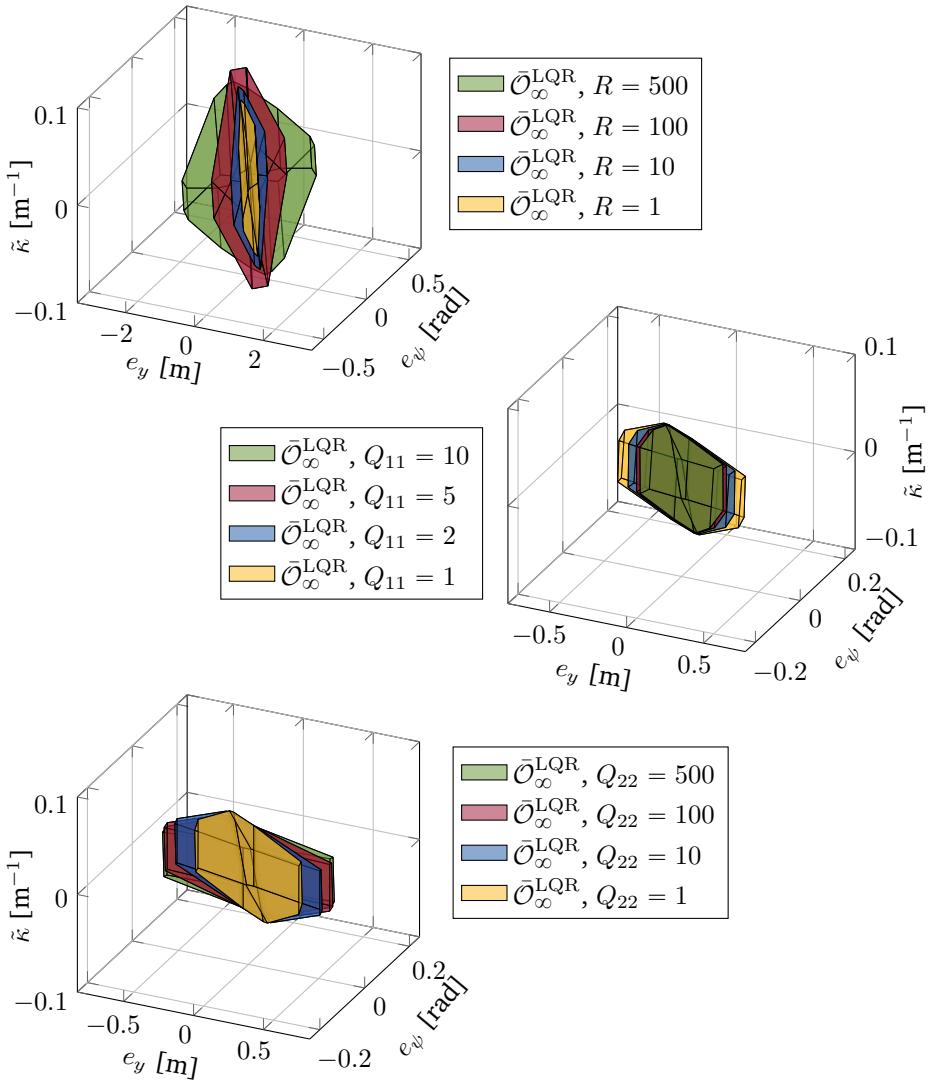


Figure 8.3: Maximum positive invariant set for all  $\gamma \in \Gamma$  considering input rate constraints. At the top, the input penalization matrix  $R$ , which is related to the curvature  $\kappa$ , is varied. In the middle, the first element of the state penalization matrix  $Q$ , which is related to the lateral displacement  $e_y$ , is varied. At the bottom, the second element of the state penalization matrix  $Q$ , which is related to the heading displacement  $e_\psi$ , is varied.

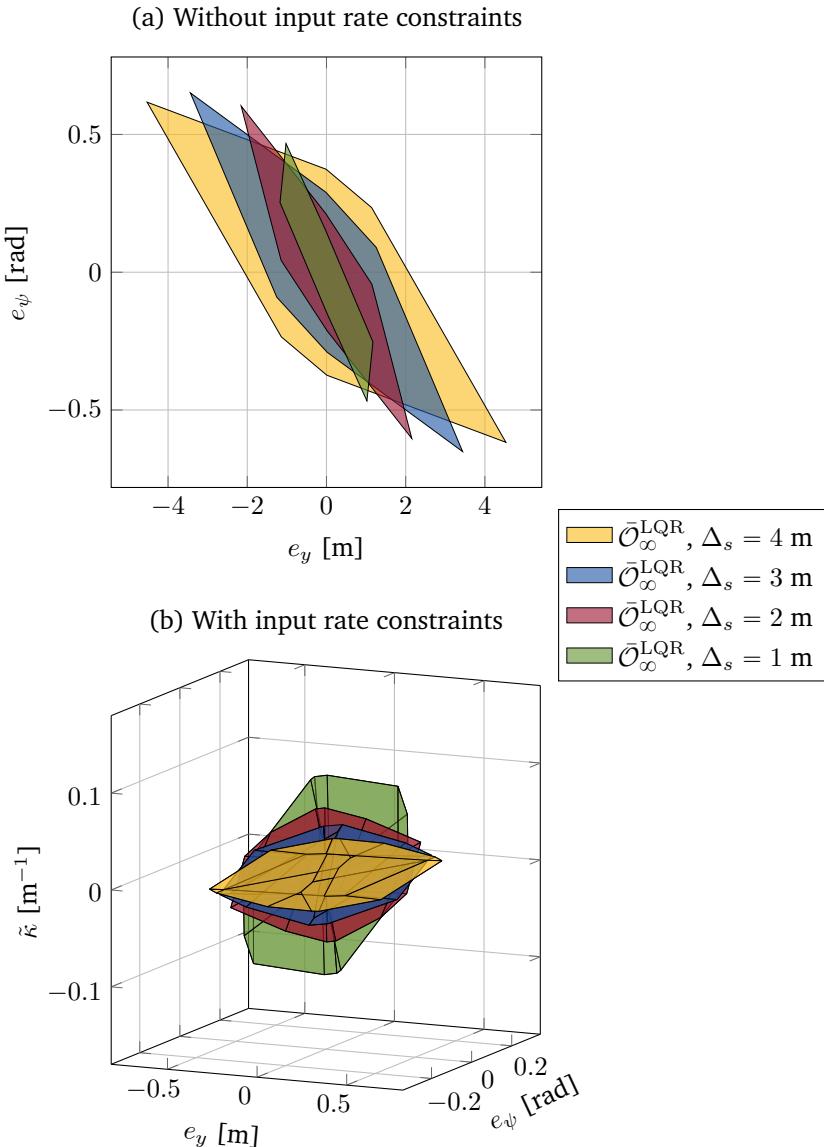


Figure 8.4: Maximum positive invariant set for all  $\gamma \in \Gamma$  for different sampling distances  $\Delta_s = vT_s$ . At the top, LTV invariant set *ignoring* input rate constraints. At the bottom, LTV invariant set *considering* input rate constraints.

which can cause infeasibility problems. Finally, the influence of the heading displacement penalization (i.e.,  $Q_{22}$ , the second diagonal element of  $Q$ ) is reflected in a larger  $e_y$  region covered and smaller  $e_\psi$  region covered. The interpretation lies in the fact that, if heading displacement is expensive, then the invariant set avoids including large  $e_\psi$ . The tuning of the state penalization matrix  $Q$  does not influence significantly the LTV invariant set curvature range, while in the case of the input penalization matrix  $R$  it clearly does.

### Influence of Speed

Assuming constant speed within the prediction horizon distance is a valid assumption, since in autonomous heavy-duty vehicle applications the driving should not be aggressive and the vehicle has (very) slow dynamics. However, the vehicle drives at different speeds, so it is relevant to investigate how different velocities influence the maximum positive invariant set of the system (8.3) considering or ignoring input rate constraints. The maximum positive invariant set for a discrete set of speeds can be computed offline and stored in a look-up table. Then, depending on the current vehicle speed, the terminal state constraint is chosen accordingly online. Figure 8.4 shows different  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$  for different speeds (equivalently the sampling distance  $\Delta_s$ , since the sampling time is maintained constant  $T_s = 100$  ms). It can be seen that the higher the speed the more skewed the shape of the invariant set. As the speed increases, the prediction distance also increases and consequently, the invariant set is typically larger allowing larger  $e_\psi$  that are directly correlated with larger  $e_y$ . Note that, when input rate constraints are considered, the set curvature range is significantly reduced for higher speeds. This is logical and a desired behavior as the vehicle travels faster (i.e., takes less time to travel the same distance) and, therefore, is able to steer less within the same distance.

#### 8.2.2 Computation of the Terminal Cost $Q_f$

When the model is LTI (i.e., for a specific  $\xi \in \Xi$ ), it can be shown that a control law  $u(t) = F(\xi)\tilde{z}(t)$  that stabilizes the system (e.g., the LQR feedback control law) results in an infinite-horizon cost given by

$$J_\infty^\star(\tilde{z}(t)) = \tilde{z}(t)^T P(\xi(t)) \tilde{z}(t) = \sum_{k=0}^{\infty} \tilde{z}_{k|t}^T Q \tilde{z}_{k|t} + \tilde{u}_{k|t}^T R \tilde{u}_{k|t}, \quad (8.7)$$

where  $P(\xi)$  is given by the solution of the ARE for the system (8.1) for a specific  $\xi \in \Xi$ , i.e.,

$$A(\xi)^T \left( P(\xi) - P(\xi)B(\xi)(B(\xi)^T P(\xi)B(\xi) + R)^{-1} B(\xi)P(\xi) \right) A(\xi) + Q - P(\xi) = 0, \quad (8.8)$$

which can be rewritten as

$$A_{\text{cl}}(\xi)^T P(\xi) A_{\text{cl}}(\xi) + F(\xi)^T R F(\xi) + Q - P(\xi) = 0, \quad (8.9)$$

where  $A_{\text{cl}}(\xi) = (A(\xi) + B(\xi)F(\xi))$ . Therefore, the terminal cost  $\tilde{z}_{N|t}^T Q_f \tilde{z}_{N|t}$  is typically chosen as the solution of the algebraic Riccati equation for the system (8.1) for a given  $\xi \in \Xi$ .

The determination of an upper bound of the infinite-horizon predicted cost for all  $\gamma \in \Gamma$  is fundamental to prove LTV-MPC closed-loop stability through Lyapunov techniques. In other words, a  $\bar{P}$  such that

$$A_{\text{cl}}(\xi)^T \bar{P} A_{\text{cl}}(\xi) + F(\xi)^T R F(\xi) + Q - \bar{P} \leq 0, \quad \forall \xi \in \Xi, \quad (8.10)$$

is sought. Note that, from (8.9)

$$F(\xi)^T R F(\xi) + Q = P(\xi) - A_{\text{cl}}(\xi)^T P(\xi) A_{\text{cl}}(\xi),$$

then we can rewrite (8.10) as

$$A_{\text{cl}}(\xi)^T (\bar{P} - P(\xi)) A_{\text{cl}}(\xi) - (\bar{P} - P(\xi)) \leq 0, \quad \forall \xi \in \Xi. \quad (8.11)$$

Our approach to find such  $\bar{P}$  is to let  $\bar{P} = \beta P(\xi')$ ,  $\xi' \in \Xi$ , where a  $\beta > 1$  is chosen such that (8.11) holds. To find the closed-form expression for  $P(\xi)$  for all  $\xi \in \Xi$ , the associated algebraic Riccati equation is solved symbolically and the only positive definite solution is chosen. Figure 8.5 shows the largest eigenvalues of the left-hand side of the inequality (8.11), for all possible  $\xi \in \Xi$  and  $\xi' \in \Xi$  as a function of  $\beta$ , when input constraints are ignored. Also, we investigate the influence of  $Q$  and  $R$  on the eigenvalues of the left-hand side of (8.11). It can be seen that the evolution of the maximum eigenvalue depending on  $\beta$  is monotonically decreasing. The inequality (8.11) is only satisfied when all the eigenvalues of the left-hand side are nonpositive. In fact, one of the eigenvalues is always negative. The other eigenvalue is negative for most  $\beta$  but can be positive for  $\beta$  values close to 1. The results are analogous when input rate constraints are considered, with one more negative eigenvalue.

Figure 8.6 shows the eigenvalues of the left-hand side of inequality (8.10) for a fixed  $\beta = 1.2$  and ignoring input rate constraints. The eigenvalues of the left-hand side of the inequality (8.10) are computed as a function of  $\xi$  for each  $\xi'$ . The eigenvalues of the left-hand side matrix are always negative, which means that the inequality (8.10) is satisfied when using  $\beta = 1.2$ . The results are analogous when input rate constraints are considered, with one more negative eigenvalue.

Note that it is possible to compute the smallest  $\beta > 1$  such that (8.11) holds. However, in practice, the vehicle prediction differs from the actual vehicle motion. Then, it is desired  $\beta$  to be chosen such that a conservative estimate of infinite-horizon cost (8.7) is achieved.

When input rate constraints are considered, the terminal cost term includes, not only the terminal states, but also the terminal input. Therefore, the MPC cost function (8.4a) needs to be modified. Let  $w = [\tilde{z}_{N|t}, \tilde{u}_{N-1|t}]^T$ , then the terminal cost term becomes  $w^T Q_f w$ .

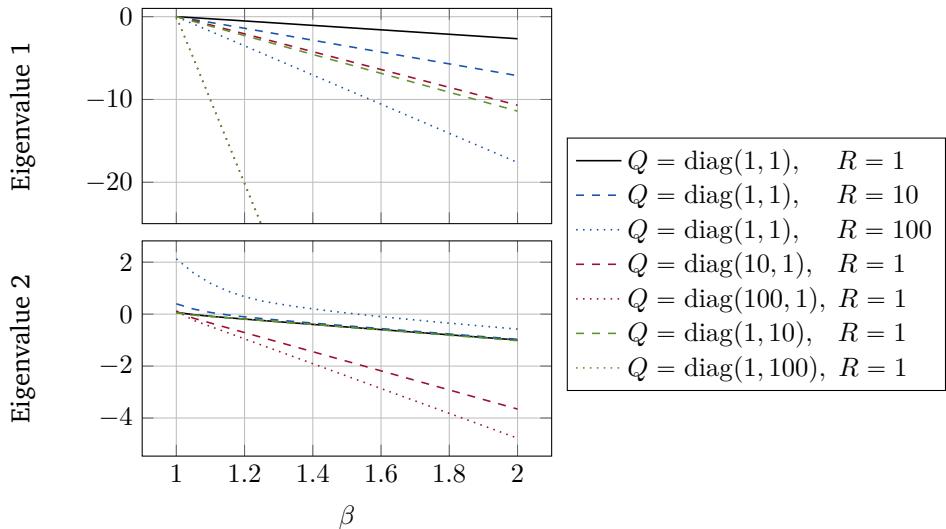


Figure 8.5: Largest eigenvalues of the left-hand side of the inequality (8.11). When both eigenvalues are nonpositive the inequality (8.10) is satisfied.

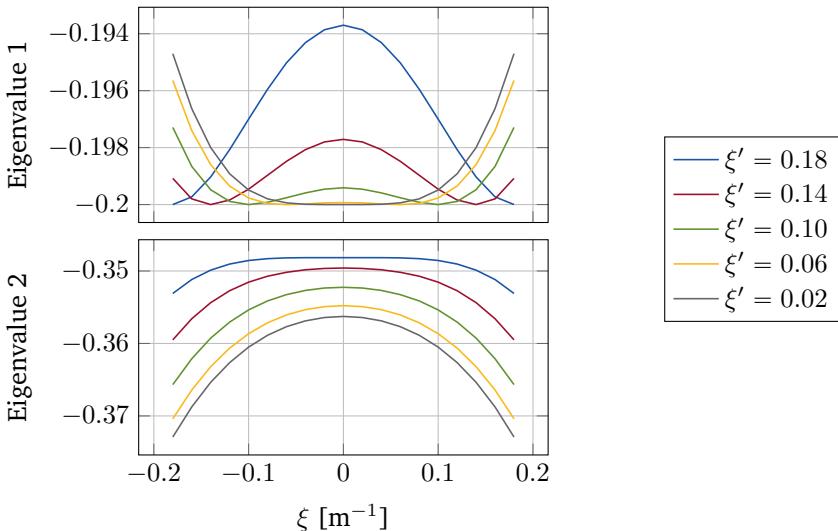


Figure 8.6: Eigenvalues of the left-hand side of the inequality (8.10) for different  $\xi' = \kappa_r$  as a function of  $\xi = \kappa_r$  and  $\beta = 1.2$ . The eigenvalues are always negative, so the inequality (8.10) is satisfied.

### 8.3 Closed-loop Stability and Feasibility

In this section, Lyapunov techniques are used to prove closed-loop stability of the proposed LTV-MPC scheme (8.4), following the notation and argumentation flow used by Borrelli et al. (2017). The terminal constraint and terminal cost are determined via the methods explained in Section 8.2.

**Assumption 8.1.** *The state penalization matrices  $Q$  and  $Q_f$  and the input penalization matrix  $R$  are symmetric positive definite, i.e.,  $Q = Q^T > 0$ ,  $Q_f = Q_f^T > 0$ , and  $R = R^T > 0$ .*

**Assumption 8.2.** *The sets  $\tilde{\mathcal{Z}}$ ,  $\tilde{\mathcal{Z}}_f$ , and  $\tilde{\mathcal{U}}$  are closed and contain the origin in their interior.*

**Assumption 8.3.**  *$\tilde{\mathcal{Z}}_f$  is control invariant and  $\tilde{\mathcal{Z}}_f \subset \tilde{\mathcal{Z}}$ .*

**Assumption 8.4.** *There is no model mismatch between the prediction model and the plant model, i.e.,*

$$\begin{aligned} A(\xi(k|t)) &= A(\xi(k-n|t+n)), \\ B(\xi(k|t)) &= B(\xi(k-n|t+n)), \end{aligned} \tag{8.12}$$

for all  $t > 0$ ,  $k = 0, \dots, N-1$ , and  $n = 0, \dots, k$ .

To prove closed-loop stability of the MPC controller under the proposed terminal constraints and terminal cost, it is necessary to ensure recursive feasibility of the controller (i.e., the computed control sequence  $\bar{u}_{t+1}^*$  is feasible at time  $t+1$  whenever the optimization at time  $t$  is feasible).

**Theorem 8.1** (LTV-MPC Feasibility). *The problem (8.4) is feasible for all  $t \geq 0$  if  $Q_f = \bar{P}$ ,  $\tilde{\mathcal{Z}}_f = \bar{\mathcal{O}}_\infty^{\text{LQR}}$ , and if  $z(0) \in \mathcal{K}_N(\bar{\mathcal{O}}_\infty^{\text{LQR}})$ .*

*Proof.* If  $z(0) \in \mathcal{K}_N(\bar{\mathcal{O}}_\infty^{\text{LQR}})$ , then the system is feasible at  $t = 0$  (i.e., there exists a sequence of  $N$  inputs that brings the system to the set  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$ ). By definition of  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$ , the system constraints are instantaneously satisfied at all points in  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$ , and  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$  is invariant under the control law  $u(t) = L_{\text{LQR}}(\xi)\tilde{z}(t)$ ,  $\forall \xi \in \Xi$ , since  $\bar{\mathcal{O}}_\infty^{\text{LQR}} \subset \mathcal{O}_\infty^{\text{LQR}}$ . Thus, the system is feasible for all  $t \geq 0$ .  $\square$

**Theorem 8.2** (Closed-loop stability of the LTV-MPC scheme). *Consider the model (8.1), its constraints (2.6), and the LTV-MPC controller (8.4). Let Assumptions 8.1 - 8.4 hold. The terminal penalty matrix  $Q_f$  is chosen as  $\bar{P}$ , and the terminal constraint  $\tilde{\mathcal{Z}}_f$  is chosen as  $\bar{\mathcal{O}}_\infty^{\text{LQR}}$ . Then, the state of the closed-loop system converges to the origin. Moreover, the origin of the closed-loop system is asymptotically stable with domain of attraction  $\mathcal{K}_N(\bar{\mathcal{O}}_\infty^{\text{LQR}})$ .*

*Proof.* Consider problem (8.4) at time  $t$ . The argument  $\xi(t)$  from  $A_d$  and  $B_d$  is dropped for the sake of simplicity. Let  $z(t) \in \mathcal{K}_N(\bar{\mathcal{O}}_\infty^{\text{LQR}})$  and let  $\check{u}_t^\star = \{\tilde{u}_{k|t}^\star\}_{k=0}^{N-1}$  be the optimizer of problem (8.4) and  $\check{z}_t^\star = \{\tilde{z}_{k|t}^\star\}_{k=0}^N$  be the corresponding optimal state trajectory. After the implementation of  $\tilde{u}(t) = \tilde{u}_{0|t}^\star$  and using Assumption 8.4,  $\tilde{z}(t+1) = \tilde{z}_{1|t} = A_d\tilde{z}(t) + B_d\tilde{u}_{0|t}^\star$  is obtained. Let  $J_N^\star(\tilde{z}(t))$  be the optimal total predicted cost of (8.4a) when applying the  $\check{u}_t^\star$  to the system state  $\tilde{z}(t)$ .

Consider now problem (8.4) for  $t+1$  and lets construct an upper bound on  $J_N^\star(\tilde{z}(t+1))$ . Consider the sequence

$$\check{u}_{t+1} = \{\tilde{u}_{1|t}^\star, \dots, \tilde{u}_{N|t}^\star, F\tilde{z}_{N|t}^\star\},$$

and the corresponding state trajectory resulting from the initial state  $z(t+1)$ ,

$$\check{z}_{t+1} = \{\tilde{z}_{1|t}^\star, \dots, \tilde{z}_{N|t}^\star, (A_d + B_dF)\tilde{z}_{N|t}^\star\},$$

where  $F$  is the LQR feedback control law for the corresponding  $\xi$  at  $N+1$ . Let  $J_N(\tilde{z}(t+1))$  be the predicted cost (8.4a) when applying  $\check{u}_{t+1}$  to  $\tilde{z}(t+1)$ . It is necessary to show that  $J_N^\star(\tilde{z}(t))$  is a Lyapunov function for the closed-loop system. With direct calculations it is possible to derive the following result

$$\begin{aligned} J_N(\tilde{z}(t+1)) &= J_N^\star(\tilde{z}(t)) - (\tilde{z}_{0|t}^T Q \tilde{z}_{0|t} + \tilde{u}_{0|t}^T R \tilde{u}_{0|t}) \\ &\quad + \tilde{z}_{N|t}^T ((A_d + B_dF)^T \bar{P} (A_d + B_dF) \\ &\quad + Q + F^T RF - \bar{P}) \tilde{z}_{N|t}, \end{aligned} \quad (8.13)$$

where, if  $\beta > 1$  is chosen as explained in Section 8.2.2,

$$\tilde{z}_{N|t}^T \bar{P} \tilde{z}_{N|t} \geq \tilde{z}_{N|t}^T ((A_d + B_dF)^T \bar{P} (A_d + B_dF) + Q + F^T RF) \tilde{z}_{N|t} > 0, \quad (8.14)$$

and therefore, since  $\bar{P}$  gives the largest infinite-horizon predicted cost with respect to all systems  $\gamma \in \Gamma$ ,

$$J_N(\tilde{z}(t+1)) \leq J_N^\star(\tilde{z}(t)) - (\tilde{z}_{0|t}^T Q \tilde{z}_{0|t} + \tilde{u}_{0|t}^T R \tilde{u}_{0|t}), \quad (8.15)$$

and, since, by construction,  $J_N^\star(\tilde{z}(t+1)) \leq J_N(\tilde{z}(t+1))$ ,

$$J_N^\star(\tilde{z}(t+1)) - J_N^\star(\tilde{z}(t)) \leq -(\tilde{z}_{0|t}^T Q \tilde{z}_{0|t} + \tilde{u}_{0|t}^T R \tilde{u}_{0|t}), \quad (8.16)$$

where the right-hand side is clearly negative definite due to the positive definiteness of  $Q$  and  $R$ .

Assumption 8.1 and (8.16) ensure that  $J_N^\star(\tilde{z}(t))$  strictly decreases along the state trajectories of the closed-loop system (8.5) for any  $\tilde{z} \in \mathcal{K}_N(\bar{\mathcal{O}}_\infty^{\text{LQR}})$ ,  $\tilde{z} \neq 0$ . In addition to the fact that  $J_N^\star(\tilde{z}(t))$  decreases, it is also lower bounded by zero

and the state trajectories generated by the closed-loop system (8.5) starting from any  $\tilde{z}(0) \in \mathcal{K}_N(\bar{\mathcal{O}}_\infty^{\text{LQR}})$  lie in  $\mathcal{K}_N(\bar{\mathcal{O}}_\infty^{\text{LQR}})$  for all  $t \geq 0$ . Assumptions 8.2 and 8.3 and equation (8.16) are sufficient to ensure that the state of the closed-loop system converges to the origin as  $t \rightarrow \infty$  if the initial state lies in  $\mathcal{K}_N(\bar{\mathcal{O}}_\infty^{\text{LQR}})$ .

Lyapunov theory is used to show that the origin of the closed-loop system (8.5) is asymptotically stable.  $J_N^*(\tilde{z}(t))$  is positive definite due to Assumption 8.1. Furthermore, (8.16) states that  $J_N^*(\tilde{z}(t))$  decreases along the closed-loop trajectories. Then,  $\tilde{z}_{0|t}^T Q \tilde{z}_{0|t} + \tilde{u}_{0|t}^{*\top} R \tilde{u}_{0|t} \rightarrow 0$  as  $t \rightarrow \infty$ .  $\square$

## 8.4 Experimental and Simulation Results

This section demonstrates the effectiveness of using the terminal constraint  $\tilde{\mathcal{Z}}_f = \bar{\mathcal{O}}_\infty^{\text{LQR}}$  and cost  $Q_f = \bar{P}$  both in simulation and experimentally. The evaluation is done in a scenario that resembles an emergency scenario (see Figure 8.7). The vehicle travels in a straight line and suddenly a fictitious obstacle appears in front, forcing the vehicle to steer away from it. For purposes of experiment repeatability, the vehicle, at the moment of obstacle perception, always steers from a straight line to another straight line located one meter apart. The MPC does not have any preview information about when this change is supposed to happen. The driving performance (e.g., accuracy or smoothness) is not the focus of our analysis, but yet clear evidence of (in)stability is sought. Both in simulation and experimentally, the vehicle drives with a constant speed of 8 m/s. The road-aligned kinematic model (3.8) is used as a prediction model. The only parameter modified in each run is the first element of the state penalization matrix  $Q_{11}$ , which accounts for the penalization of the vehicle lateral displacement  $e_y$  with respect to the reference path. The other parameters are  $Q_{22} = 10$  and  $R = 10$ , and the prediction horizon is  $N = 3$  steps, using a spatial discretization of  $T_s = 100$  ms. In this scenario, only  $Q_{11}$  is changed, since larger  $Q_{22}$  or  $R$  yield less aggressive control and consequently stable behavior.

The terminal state constraints and cost are computed offline, the controller runs at 50 Hz, and each MPC optimization is solved using qpOASES (Ferreau et al., 2014). For practical reasons (i.e., to ensure feasibility all the time), a slack variable is included in the terminal constraints (8.4e) and penalized quadratically in the cost function (8.4a). The initial tuning is such that the MPC without terminal constraint and cost is stable. A comparison is made between the MPC controllers (8.4) with and without terminal constraints and cost.

**Remark.** *In this scenario, the prediction model is LTI, since the reference path is a straight line. However, from the point of view of the MPC, the reference path is unknown after the last point of the prediction horizon and can have any (bounded) curvature. So, stability is guaranteed according to Theorem 8.2.*

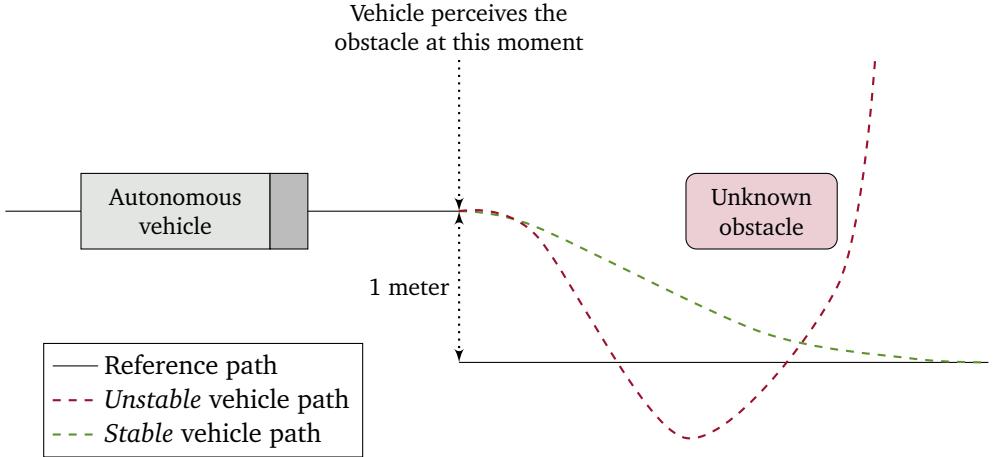


Figure 8.7: Emergency obstacle avoidance scenario illustration. The vehicle travels at constant speed in a straight line. Suddenly, an obstacle appears in front of the vehicle.

For the sake of simplicity, in the remainder of the section, the controllers are referred as follows:

- *Controller A* - LTV-MPC *without* terminal cost and terminal state set;
- *Controller B* - LTV-MPC *with* terminal cost and terminal state set *ignoring* input rate constraints;
- *Controller C* - LTV-MPC *with* terminal cost and terminal state set *considering* input rate constraints.

### 8.4.1 Simulation Results

The same MATLAB/Simulink simulation environment thoroughly presented in Section 7.2.2 is used. There, the simulation environment is shown to be sufficiently accurate to allow the controller evaluation and tuning before deploying it on the vehicle.

Figure 8.8 shows the lateral deviation from the path and the steering angle computed during the simulation. It is clear that the performance differences between using or not terminal cost and terminal state set when  $Q_{11} = 1$  are marginal. However, at  $Q_{11} = 5$  the differences in performance are more noticeable, where controllers B and C converge to the reference path, while controller A is unstable. In fact, note that controller C overshoots less than controller B, due to the knowledge of the input rate constraints. Increasing  $Q_{11}$ , controller A is still unstable and

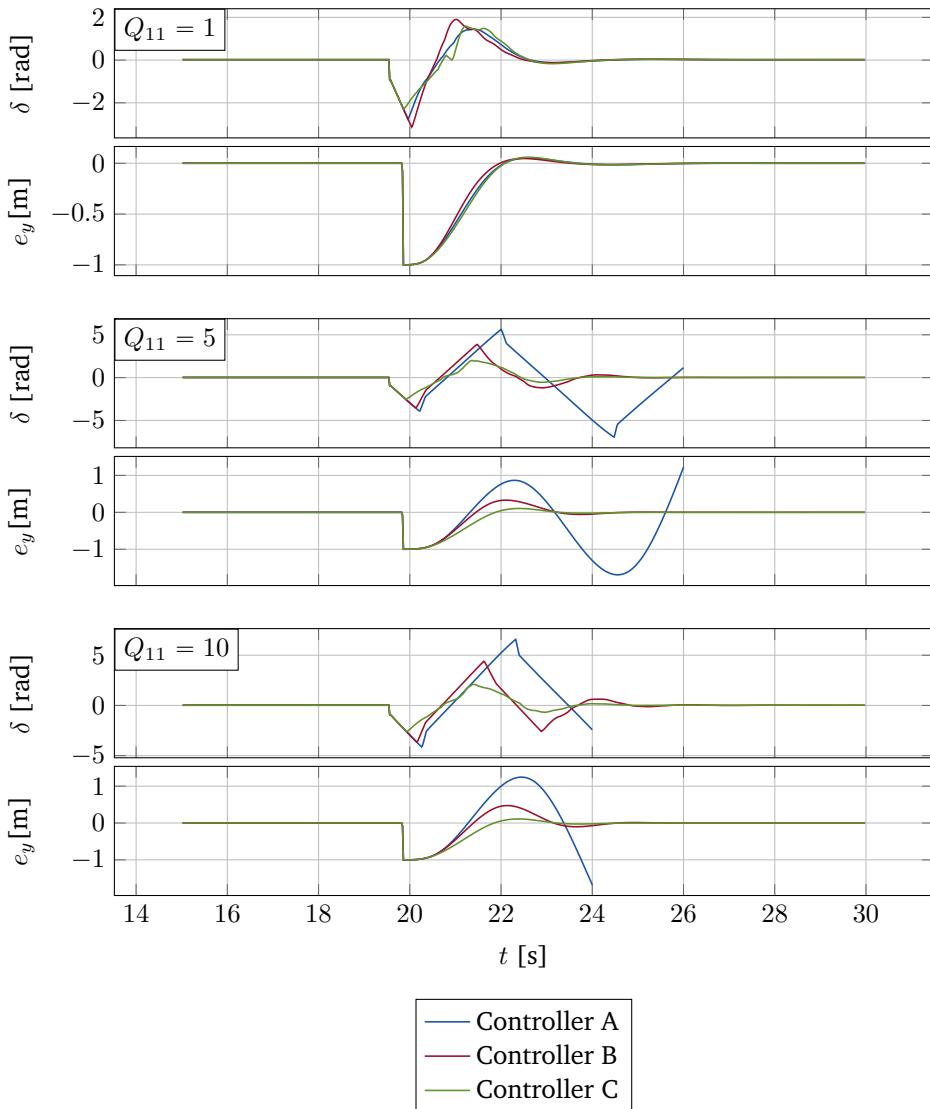


Figure 8.8: *Simulation of the MPC controller (8.4).* Just before 20 seconds of simulation, the obstacle is detected and one meter reference change happens. The effectiveness of the terminal cost and terminal state set (with and without input rate constraints) derived in Sections 8.2.1 and 8.2.2 are evaluated for different MPC parameter tuning.

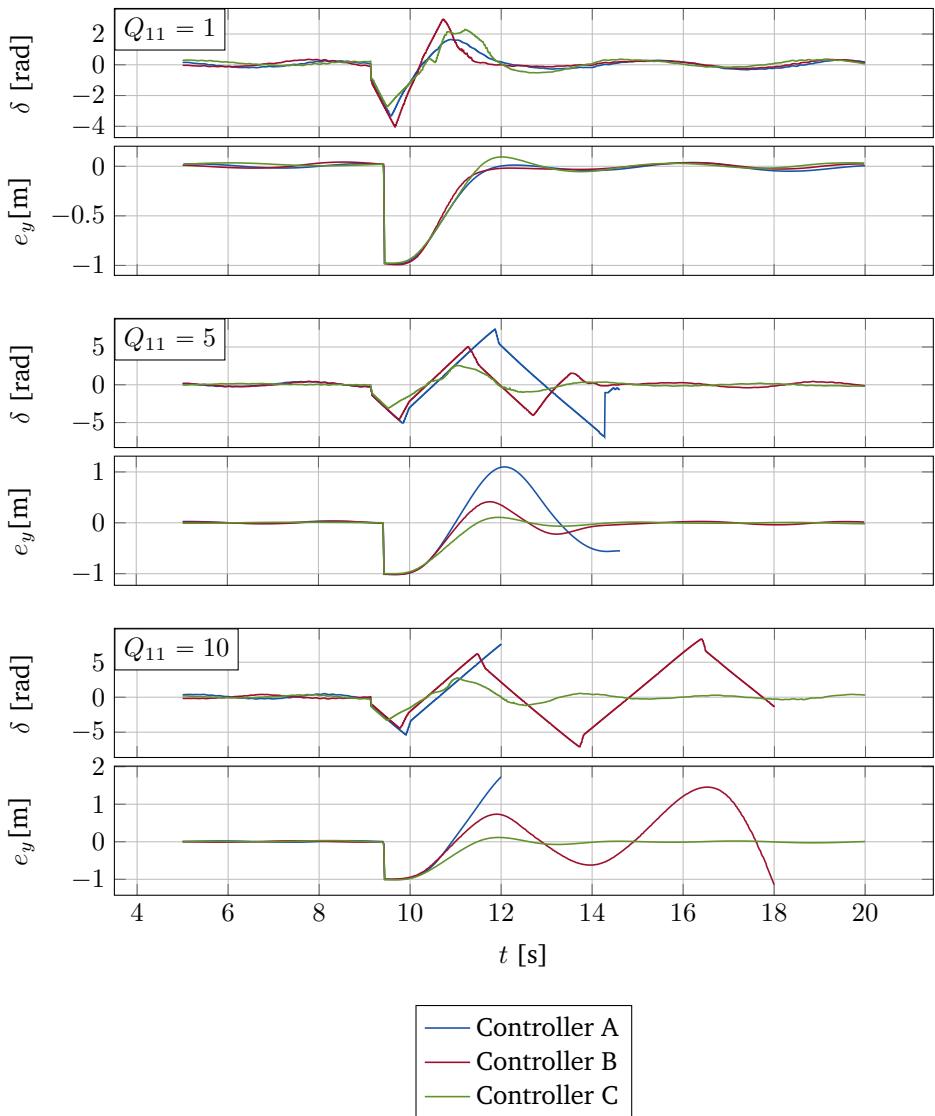


Figure 8.9: *Experiment of the MPC controller (8.4).* Just before 10 seconds of experiment, the obstacle is detected and one meter reference change happens. The effectiveness of the terminal cost and terminal state set (with and without input rate constraints) derived in Sections 8.2.1 and 8.2.2 are evaluated for different MPC parameter tuning.

controller B overshoots even more, while controller C keeps the same performance as before. Due to the knowledge of the input rate constraints, controller C is less aggressive than controllers A or B (i.e., longer rise time), which ultimately yields stability.

Note that, simulating a nonlinear model with dynamics, the Assumption 8.4 of no model mismatch does not hold anymore. Nevertheless, the control design yields closed-loop stability, since it is a conservative approach, as it considers all possible LTI models to construct the terminal state set and cost. This resembles, in some aspects, the tube-based robust MPC schemes (Mayne et al., 2005). In the robust MPC scheme, the state and input constraints set is shrunk depending on the upper bound of the disturbance subject to the system. In the LTV-MPC case, the terminal state set is also smaller and contained inside the terminal constraint set of each LTI model individually. In summary, there is some inherent robustness in this approach. However, its quantification lies outside of the scope of this work.

#### 8.4.2 Experimental Results

The truck used in the experimental evaluation is a modified Scania G480 construction truck and is shown in Figure 3.1. The vehicle is equipped with a sensor platform and a servo motor for automated control of the steering column. The reference path, consisted of two straight lines one meter apart, is generated offline and is located at Scania test facilities near Södertälje, Sweden. Figure 8.9 shows the deviation from the path and the steering angle computed during the experiment. The main conclusions from the simulation analysis are, in general, also valid here. The main difference is that, when increasing  $Q_{11}$ , controller B becomes unstable earlier. It is clear that controller C is stable for the different parameter tunings evaluated. A video of the experiment is available on Zenodo (Lima et al., 2017d).

The prediction horizon is the same as in simulation. Note that the prediction horizon,  $N\Delta_s = 4.8$  meters, is a fairly short one. In this particular scenario and for the speed considered, increasing the prediction horizon is actually stable. In fact, Chapter 7 shows experiments where an autonomous truck with an MPC is stable without terminal cost and terminal state set and  $N = 10$  and  $T_s = 200$  ms. To possibly induce instability using larger prediction horizon steps, it would be required to drive faster and perform more aggressive and dangerous maneuvers. However, it is not desirable to perform such experiments due to safety reasons.

In practice, the same reasoning about robustness provided in the simulation results section holds.

### 8.5 Summary

In this chapter, we investigated the design of the terminal cost and terminal state set when designing and implementing a reference tracking LTV-MPC for an autonomous vehicle such that closed-loop stability is guaranteed.

To predict the vehicle motion in the MPC, a linearized version of a nonlinear space-based kinematic model on a road-aligned coordinate frame was used. The determination of the terminal constraint and cost is performed using a multi-model representation that spans the possible LTI vehicle models within a predefined parameter range. The terminal constraint is the maximal positive invariant set for all the models in the multi-model description. The terminal cost is the upper bound on the cost-to-go incurred by applying an LQR control law to any of the possible models in the multi-model representation. The upper bound was obtained by positively scaling one of Riccati matrices resulting from the cost-to-go calculation. By including the terminal cost and terminal state set in the controller formulation, closed-loop asymptotic stability of the LTV-MPC scheme is proved through Lyapunov arguments.

Finally, the proposed control design was evaluated in simulation and experimentally. The scenario considered resembled an emergency situation, where at constant speed, the vehicle had to avoid a fictitious obstacle. The effectiveness of the terminal cost and terminal state set that include the input rate constraints was clear. When the terminal cost and terminal state set were not used, increasing controller aggressiveness led to controller instability, even when using terminal cost and terminal state set without considering input rate constraints. On the other hand, when using the terminal cost and terminal state set that consider the input rate constraints, the controller was always stable.



## Chapter 9

# Conclusions and Future Work

**A**utonomous driving technology promises more energy efficient, more convenient, and safer transport systems. However, there are still countless open problems to address, namely in motion planning and control.

In this thesis, we addressed the problem of motion planning and control of autonomous heavy-duty vehicles using optimization-based techniques, such as convex optimization and receding-horizon control. First, we described the modeling of the lateral and longitudinal vehicle dynamics for the development of an accurate simulation environment. We derived the space-based kinematic vehicle model in the road-aligned frame that was used throughout the thesis to predict the vehicle motion. We proposed an algorithm that allows for a clothoid-based path description using a small number of points. We introduced an optimization-based framework, in which path planning problems can be formulated as LP or QP and where vehicle body dimensions can be explicitly taken into account. We addressed the problem of progress maximization (i.e., minimization of traveling time) using an MPC, in which the vehicle operated close to its handling limits. We designed and experimentally evaluated a Smooth and Accurate MPC (SA-MPC) tailored for industrial vehicles, where one of the goals is to reduce the vehicle “wear and tear” during its operation. Finally, we theoretically proved and experimentally demonstrated feasibility and stability of an LTV-MPC designed for autonomous path following.

In the following sections, we discuss, in detail, the main conclusions and present open questions for future research directions.

### 9.1 Conclusions

First, we presented two different vehicle models. To develop an accurate simulation environment, the lateral dynamics of a truck were described by a 4-axle bicycle model validated against real truck data. We identified the curvature request to steering angle dynamics of a truck and determined some important vehicle parameters, such as the wheel angle ratio between the two front axles and the understeer-

ing gradient. We then described the longitudinal and the cruise controller models developed by Scania CV AB and integrated them in the overall vehicle model. We presented the kinematic vehicle model, which does not include any dynamics, but is valid at low speeds (i.e., when vehicle lateral dynamics are negligible). We also derived the vehicle model in the road-aligned spatial frame, where the vehicle motion is described with respect to a given path. We recalled the notions of linearization and discretization of a continuous-time or -space nonlinear system. Although simple, this model has proven to be accurate enough, since at high speeds the vehicle mostly drives straight and in high curvature roads the vehicle drives slowly. In both cases, the lateral dynamics are considered negligible.

Second, we presented an algorithm that produces a sparse path representation, where a reduced number of clothoids is used to describe a reference path. This approach relies on a reweighted  $\ell_1$ -norm approximation of the  $\ell_0$ -norm. The clothoid-based path has few points where each point, the so-called kink-point, has embedded the clothoid segment properties, resulting in a computationally efficient path description. The number of clothoids used to describe the path is dependent on the maximum deviation allowed between the original and the clothoid-based path. The smaller the maximum allowed deviation, the larger the number of clothoid segments.

Third, we introduced an optimization-based path planning framework for autonomous vehicles. The main benefit of the proposed approach is the efficient incorporation of actuator, obstacle avoidance, and, foremost, vehicle dimension constraints. The path planning problem was formulated as an LP or QP, and the inclusion of vehicle dimensions constraints directly in the optimization problem aims at improving path planning performance in very constrained environments. Moreover, vehicle body constraints deformation, due to lengthy vehicles or large curvature roads scenarios, was also modeled in the same framework. In particular, such modeling was effectively applied to an overhang minimization scenario for an autonomous bus. In simulation, the effectiveness of the method was shown in two other scenarios: a tight maneuvering scenario and a low curvature road scenario.

Fourth, we designed a Progress Maximization MPC (PM-MPC) for autonomous driving. The problem of maximizing the progress (i.e., minimizing the traveling time) along a given path (e.g., the centerline of a road) is of utmost importance for efficiency and productivity increase in industrial applications. Besides maximizing the progress along a given path, the controller also respects actuator, handling, and driving corridor constraints. In order to achieve real-time implementable solutions, we proposed that the vehicle motion prediction is done by means of a kinematic vehicle model combined with vehicle dynamics constraints. These constraints represent the vehicle handling limits, modeled using a ‘g-g’ diagram, which limits the vehicle acceleration. In particular, we approximate the vehicle ‘g-g’ diagram by an inscribed convex polytope. PM-MPC was formulated as a convex optimization problem by linearizing the vehicle model around previous solutions, and by using linearized versions of the constraints modeling the handling dynamics. The proposed controller optimizes the vehicle lateral and longitudinal motion simulta-

neously and its effectiveness was demonstrated, in simulation, even in the presence of obstacles.

Fifth, we designed an SA-MPC for lateral control of an autonomous truck. The MPC cost function explicitly maximizes comfort and smooth driving by minimizing the first- and second-order spatial derivatives of the vehicle curvature. The problem formulation was inspired in clothoids, known by their comfort properties and used in road design, which have a null curvature second derivative (i.e., PWA curvature function). In addition, the SA-MPC includes input and state constraints. The first are related to the vehicle curvature and curvature rate. For the second, we define a driving corridor where the vehicle motion prediction must be. The state constraints are soft constraints and therefore, a slack variable that penalizes deviations from the path is also minimized in the cost function. We analyzed the influence of the controller parameters and concluded that low path following accuracy requirements and a wide driving corridor contribute to a smoother driving. Moreover, the curvature rate penalization parameter encourages small curvature changes. The SA-MPC is compared against existing approaches, such as a Pure-Pursuit Controller (PPC) and a standard MPC. In simulation, we compared SA-MPC with a standard MPC and a PPC. Also, we deployed the SA-MPC on a Scania construction truck and compared it with a PPC. The performance of the controllers was evaluated in terms of path following accuracy and curvature rate. To that end, we provided a statistical analysis of the data. We concluded that the PPC is the smoothest controller but is also the least accurate one. The SA-MPC and the standard MPC had similar path tracking accuracy performances. However, the SA-MPC was smoother than the standard MPC with almost no loss of path following accuracy. In the end, we showed that the SA-MPC has an extremely promising performance in real experiments with an average absolute lateral deviation from the path of 6 cm and never exceeding 30 cm in challenging tracks, at both high and low speeds. Since the beginning of 2016, SA-MPC is the main path following controller used in four different autonomous Scania construction trucks and one autonomous Scania bus.

Sixth, we proposed a novel terminal cost and a terminal state set in order to guarantee closed-loop stability when designing and implementing an LTV-MPC for autonomous path following. The determination of the terminal constraint and cost was performed using a multi-model representation that spans the possible LTI vehicle models within a predefined parameter range. The terminal constraint was designed as the maximal positive invariant set for all the models in the multi-model description. The terminal cost was obtained by positively scaling any of the Riccati matrices that represent the infinite cost-to-go incurred by applying an LQR control law to any of the possible models in the multi-model representation. By including the terminal cost and terminal state set in the controller formulation, closed-loop asymptotic stability of the LTV-MPC scheme was proved through Lyapunov arguments. Finally, the effectiveness of the proposed MPC design was evaluated in simulation and in real experiments with a Scania construction truck in a scenario that resembles an emergency maneuver, where the vehicle avoids a fictitious ob-

stacle. The controller successfully stabilized an autonomous Scania construction truck, when other controllers, with no or milder terminal cost and terminal state set constraints, could not stabilize the vehicle.

## 9.2 Future Work

In this section we discuss possible future work directions on motion planning and control design for autonomous driving.

### 9.2.1 Online Tuning of Model Parameters

The SA-MPC, presented in Chapter 7, still requires a significant amount of tuning and the prediction accuracy can be further improved. For this reason, we believe that the development of a procedure to systematically tune the parameters of the controller while driving (i.e., online) would be extremely useful. One possible, interesting approach would be to combine machine learning or system identification techniques with model predictive control to adapt the controller or model parameters depending on the environment. This would make it possible to use the same controller in several different vehicles.

### 9.2.2 Advanced Vehicle Models for Prediction

The PM-MPC, presented in Chapter 6, uses a kinematic vehicle model combined with vehicle dynamics constraints to control a vehicle at its handling limits. However, these handling limits were derived using a linear tire model, which typically overestimates the vehicle lateral forces and load transfers during high cornering phases. Consequently, in practice, it would be hard to bring the vehicle to the limits of the ‘ $g-g$ ’ diagram without large prediction errors that could drive the vehicle outside of the driving corridor. Therefore, it would be interesting to consider more advanced tire models, eventually coupling the lateral and longitudinal forces. Also, considering a dynamic vehicle model directly in the prediction model could improve the controller performance.

### 9.2.3 Delay Estimation

As explained in Chapter 3 and mentioned in Chapters 7 and 8 , the delay between curvature request and actual vehicle curvature is one of the most prominent factors that need to be taken into account when designing the motion controller. As of today, this delay is considered static and estimated offline. However, it has been seen that the delay is not static and depends on the vehicle speed and curvature. Therefore, estimating this delay online may greatly improve the controller performance.

### 9.2.4 Robust Control

So far, the utilization of a simple model has proven to be sufficient for a good controller performance under standard driving conditions. It would be interesting to investigate the limitations of the proposed approach in more challenging driving conditions. For example, in a mining scenario, roads are slippery, have high slopes and banking angles, and are full of debris. Also, the vehicle is subject to different loads that influence the driving behavior. Therefore, there is a need to investigate and design more complex vehicle models providing better motion prediction in more advanced scenarios. One possible approach would be to extend the current formulation in the light of robust MPC to mitigate the influence of model uncertainties in the controller performance.

### 9.2.5 Longitudinal Controller Design

The design of a longitudinal controller was outside the scope of this thesis. The main reason was that commercial solutions, such as the cruise controller, already exist and perform well. Currently, a reference generator sends a speed request to the cruise controller. However, there is no guarantee that the speed profile is actually performed. Trajectory tracking (i.e., following a reference parametrized in time) requires controlling accurately the vehicle speed such that, for example, obstacles can be safely avoided.

### 9.2.6 Combined Lateral and Longitudinal Controller Design

The SA-MPC, presented in Chapter 7, assumes constant speed and is only responsible for the lateral control of the vehicle. However, as seen in Chapter 6, when the vehicle dynamics cannot be neglected, the lateral and longitudinal control must be designed together. This would improve the vehicle prediction quality and enlarge the controller operation scope.

### 9.2.7 Integrated Motion Planning and Control

On the one hand, the motion planning algorithms described in this thesis assume the existence of a lower layer responsible for accurately following the planned path and speed profile. On the other hand, the motion controllers that we proposed assume the existence of an upper layer responsible for planning, where obstacles are taken into account and a feasible path is fed to the controller. As of today, the controller assumes that the received path is obstacle free and that the path planner is in charge of replanning in case of a sudden environment change. An interesting research direction is the combination between planning and control. In emergency situations, there might be no time for replanning. Therefore, the controller should be designed to be aware of the environment and act accordingly to the different situations. Another interesting research direction, is the clear understanding of the

pros and cons of delegating more responsibilities to only one of the modules and where lies the ideal trade-off.

### **9.2.8 Optimization-Based Path Planning Extensions**

We believe that the optimization-based path planning framework presented in Chapter 5 would easily incorporate multi-body vehicles (e.g., articulated buses and truck and trailer configurations). Moreover, path planning in urban environments typically requires dealing with dynamic obstacles. This requires that the speed planning is performed simultaneously with the lateral path planning. It would be interesting to investigate how dynamic obstacles can be projected in the space-based road-aligned frame.

### **9.2.9 Optimization-Based Path Planning Experimental Evaluation**

In Chapter 5, the optimization-based path planning framework was evaluated in simulation. In practice, the path planning algorithm would face different challenges, such as the estimation of the driving corridor bounds and detection of obstacles. Also, it would be interesting to compare the proposed algorithm with experienced bus or truck drivers.

# Acronyms

ADAS	Advanced Driver Assistance Systems
AFS	Active Front Steering
ARE	Algebraic Riccati Equation
CL-RRT	Closed-Loop RRT
CNN	Convolutional Neural Network
EKF	Extended Kalman Filter
EU	European Union
FE	Forward Euler
FFI	Fordonsstrategisk Forskning och Innovation
GDP	Gross Domestic Product
GPS	Global Positioning System
IMU	Inertial Measurement Units
LIDAR	Light Detection and Ranging
LP	Linear Program
LQR	Linear-Quadratic Regulator
LTI	Linear-Time Invariant
LTV	Linear-Time Varying
MDP	Markov Decision Process
MPC	Model Predictive Controller
MPT	Multi-Parametric Toolbox
NMPC	Nonlinear Model Predictive Controller
PM-MPC	Progress Maximization Model Predictive Controller
PPC	Pure-Pursuit Controller
PWA	Piecewise Affine
QP	Quadratic Program
RADAR	Radio Detection and Ranging
RRT	Rapidly-Exploring Random Trees

RTK-GPS	Real-Time Kinematic Global Positioning System
SA-MPC	Smooth and Accurate Model Predictive Controller
SLP	Sequential Linear Program
SQP	Sequential Quadratic Program
V2V	Vehicle-to-Vehicle Communication
V2I	Vehicle-to-Infrastructure Communication
ZOH	Zero-Order Hold

# Bibliography

- ADAMS. *ADAMS Car documentation*. MSC Software Corporation, 2015.
- Alam, A. *Fuel-efficient heavy-duty vehicle platooning*. PhD thesis, KTH Royal Institute of Technology, Stockholm, 2014.
- Alam, A., Besselink, B., Turri, V., Mårtensson, J., and Johansson, K. H. Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency. *IEEE Control Systems*, 35(6):34–56, 2015a.
- Alam, A., Mårtensson, J., and Johansson, K. An experimental study on the fuel reduction potential of heavy-duty vehicle platooning. *Control Engineering Practice*, 38:11–25, 2015b.
- Amidi, O. and Thorpe, C. E. Integrated mobile robot control. In *Mobile Robots V*, volume 1388, pages 504–524. International Society for Optics and Photonics, 1991.
- Åström, K. J. and Wittenmark, B. *Computer-controlled systems: Theory and design*. Courier Corporation, 2013.
- Atlas Copco. Epiroc. URL <https://www.epiroc.com/en-ie/applications/mining/automation>. Retrieved April 12, 2018.
- Audi AG. Audi announces the first vehicle-to-infrastructure (V2I) service – The new traffic light information system, 2016. URL <https://www.audiusa.com/newsroom/news/press-releases/2016/08/audi-announces-first-vehicle-to-infrastructure-service>. Retrieved April 23, 2018.
- Bandyopadhyay, T., Won, K. S., Frazzoli, E., Hsu, D., Lee, W. S., and Rus, D. Intention-aware motion planning. In *Algorithmic Foundations of Robotics X*, pages 475–491. Springer, 2013.
- Banzhaf, H., Nienhäuser, D., Knoop, S., and Zöllner, J. M. The future of parking: A survey on automated valet parking with an outlook on high density parking. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1827–1834, 2017.

- Baran, I., Lehtinen, J., and Popović, J. Sketching clothoid splines using shortest paths. In *Computer Graphics Forum*, volume 29, pages 655–664. Wiley Online Library, 2010.
- Bast, H., Delling, D., Goldberg, A., Müller-Hannemann, M., Pajor, T., Sanders, P., Wagner, D., and Werneck, R. F. Route planning in transportation networks. In *Algorithm Engineering*, pages 19–80. Springer, 2016.
- Beal, C. and Gerdes, J. C. Model predictive control for vehicle stabilization at the limits of handling. *IEEE Transactions on Control Systems Technology*, 21(4):1258–1269, 2013.
- Bellamy, D. and Pravica, L. Assessing the impact of driverless haul trucks in Australian surface mining. *Resources Policy*, 36(2):149–158, 2011.
- Bemporad, A. Model predictive control design: New trends and tools. In *Proceedings of the IEEE Conference on Decision and Control*, pages 6678–6683, 2006.
- Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.
- Bertolazzi, E. and Frego, M. G1 fitting with clothoids. *Mathematical Methods in the Applied Sciences*, 38(5):881–897, 2015.
- Bevan, G. P., Gollee, H., and O’reilly, J. Trajectory generation for road vehicle obstacle avoidance using convex optimization. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 224(4):455–473, 2010.
- Bishop, R. L. There is more than one way to frame a curve. *The American Mathematical Monthly*, 82(3):246–251, 1975.
- Björnsson, Y., Bulitko, V., and Sturtevant, N. R. TBA\*: Time-Bounded A\*. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 431–436, 2009.
- Blanchini, F. Survey paper: Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.
- Bloch, A. M., Reyhanoglu, M., and McClamroch, N. H. Control and stabilization of nonholonomic dynamic systems. *IEEE Transactions on Automatic Control*, 37(11):1746–1757, 1992.
- Borrelli, F., Falcone, P., and Keviczky, T. MPC-based approach to active steering for autonomous vehicle systems. *International Journal of Vehicle Autonomous Systems*, 3(2):265–291, 2005.
- Borrelli, F., Bemporad, A., and Morari, M. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

- Brechtel, S., Gindele, T., and Dillmann, R. Probabilistic MDP-behavior planning for cars. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 1537–1542, 2011.
- Brechtel, S., Gindele, T., and Dillmann, R. Probabilistic decision-making under uncertainty for autonomous driving using continuous pomdps. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 392–399, 2014.
- Bresson, G., Alsayed, Z., Yu, L., and Glaser, S. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220, 2017.
- Brezak, M. and Petrović, I. Real-time approximation of clothoids with bounded error for path planning applications. *IEEE Transactions on Robotics*, 30(2):507–515, 2014.
- Brockett, R. W. Asymptotic stability and feedback stabilization. *Differential geometric control theory*, 27(1):181–191, 1983.
- Broggi, A. *Automatic vehicle guidance: The experience of the ARGO autonomous vehicle*. World Scientific, 1999.
- Broggi, A., Bertozzi, M., Fascioli, A., Bianco, C., and Piazzi, A. The ARGO autonomous vehicle: Vision and control systems. *International Journal of Intelligent Control and Systems*, 3(4):409–441, 1999.
- Broggi, A., Cerri, P., Felisa, M., Laghi, M. C., Mazzei, L., and Porta, P. P. The VisLab intercontinental autonomous challenge: An extensive test for a platoon of intelligent vehicles. *International Journal of Vehicle Autonomous Systems*, 10(3):147–164, 2012.
- Broggi, A., Cerri, P., Debattisti, S., Laghi, M., Medici, P., Molinari, D., Panciroli, M., and Prioletti, A. PROUD - Public Road Urban Driverless-Car Test. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3508–3519, 2015.
- Brown, C. Autonomous vehicle technology in mining. *Engineering and Mining Journal*, 213(1):30, 2012.
- Brown, M., Funke, J., Erlien, S., and Gerdes, J. C. Safe driving envelopes for path tracking in autonomous vehicles. *Control Engineering Practice*, 61:307–316, 2017.
- Buehler, M., Iagnemma, K., and Singh, S. *The 2005 DARPA Grand Challenge: The great robot race*, volume 36. Springer Science & Business Media, 2007.
- Buehler, M., Iagnemma, K., and Singh, S. *The DARPA Urban Challenge: Autonomous vehicles in city traffic*, volume 56. Springer, 2009.

- Cadillac Motor Car Division. V2V safety technology now standard on Cadillac CTS sedans, 2017. URL <http://media.cadillac.com/media/us/en/cadillac/news.detail.html/content/Pages/news/us/en/2017/mar/0309-v2v.html>. Retrieved June 29, 2018.
- Candes, E. J., Wakin, M., and Boyd, S. Enhancing sparsity by reweighted  $\ell_1$  minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, 2008.
- Carvalho, A., Gao, Y., Gray, A., Tseng, H. E., and Borrelli, F. Predictive control of an autonomous ground vehicle using an iterative linearization approach. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 2335–2340, 2013.
- Carvalho, A., Gao, Y., Lefevre, S., and Borrelli, F. Stochastic predictive control of autonomous vehicles in uncertain environments. In *Proceedings of the International Symposium on Advanced Vehicle Control*, 2014.
- Carvalho, A., Lefévre, S., Schildbach, G., Kong, J., and Borrelli, F. Automated driving: The role of forecasts and uncertainty – a control perspective. *European Journal of Control*, 24:14–32, 2015.
- Catalin, G. and Nedevschi, S. Object tracking from stereo sequences using particle filter. In *Proceedings of the IEEE Intelligent Computer Communication and Processing*, pages 279–282, 2008.
- Caterpillar Inc. CAT - Command for hauling. URL [https://www.cat.com/en\\_US/by-industry/mining/surface-mining/surface-technology/command/command-for-hauling.html](https://www.cat.com/en_US/by-industry/mining/surface-mining/surface-technology/command/command-for-hauling.html). Retrieved April 24, 2018.
- Cerri, P., Soprani, G., Zani, P., Choi, J., Lee, J., Kim, D., Yi, K., and Broggi, A. Computer vision at the Hyundai autonomous challenge. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 777–783, 2011.
- Chang, K., Li, W., Devlin, P., Shaikhbahai, A., Varaiya, P., Hedrick, J., McMahon, D., Narendran, V., Swaroop, D., and Olds, J. Experimentation with a vehicle platoon control system. In *Proceedings of Vehicle Navigation and Information Systems Conference*, volume 2, pages 1117–1124, 1991.
- Chen, Y., Sundareswaran, V., Anderson, C., Broggi, A., Grisleri, P., Porta, P., Zani, P., and Beck, J. TerraMax: Team Oshkosh urban robot. *Journal of Field Robotics*, 25(10):841–860, 2008. ISSN 1556-4967.
- Choi, J.-W., Curry, R., and Elkaim, G. Path planning based on Bézier curve for autonomous ground vehicles. In *Proceedings of the World Congress on Engineering and Computer Science*, pages 158–166, 2008.

- Chu, K., Lee, M., and Sunwoo, M. Local path planning for off-road autonomous driving with avoidance of static obstacles. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1599–1616, 2012.
- Cirillo, M. From videogames to autonomous trucks: A new algorithm for lattice-based motion planning. In *Proceedings of the Conference on Automated Planning and Scheduling*, pages 6–12, 2016.
- Collares Pereira, G., Svensson, L., Lima, P. F., and Mårtensson, J. Lateral model predictive control for over-actuated autonomous vehicle. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 310–316, 2017.
- Collares Pereira, G., Lima, P. F., Wahlberg, B., Pettersson, H., and Mårtensson, J. Linear time-varying robust model predictive control for discrete-time nonlinear systems. In *Proceedings of the IEEE Conference on Decision and Control*, 2018. To appear.
- Cordts, M., Omran, M., Ramos, S., Rehfeld, R., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.
- Cosbey, A., Mann, H., Maennling, N., Toledoano, P., Geipel, J., and Brauch, M. D. Mining a mirage? Technical report, International Institute for Sustainable Development, 2016.
- Coulter, R. Implementation of the pure-pursuit path tracking algorithm. Technical report, Carnegie Mellon University, 1992.
- Cummins, M. and Newman, P. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.
- Cunningham, A. G., Galceran, E., Eustice, R. M., and Olson, E. Mpdm: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1670–1677, 2015.
- d’Andréa Novel, B., Campion, G., and Bastin, G. Control of nonholonomic wheeled mobile robots by state feedback linearization. *The International Journal of Robotics Research*, 14(6):543–559, 1995.
- Daniel, K., Nash, A., Koenig, S., and Felner, A. Theta\*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research*, 39:533–579, 2010.
- Darms, M., Rybski, P., and Urmson, C. Classification and tracking of dynamic objects with multiple sensors for autonomous driving in urban environments. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1197–1202, 2008.

- De Luca, A., Oriolo, G., and Samson, C. *Feedback control of a nonholonomic car-like robot*, volume 229, pages 171–253. Springer Berlin Heidelberg, 1998.
- de Wit, C. C. and Sørdalen, O. J. Exponential stabilization of mobile robots with nonholonomic constraints. *IEEE Transactions on Automatic Control*, 37(11): 1791–1797, 1992.
- de Wit, C. C., Khennouf, H., Samson, C., and Sørdalen, O. J. Nonlinear control design for mobile robots. In *Recent trends in mobile robots*, pages 121–156. World Scientific, 1993.
- Di Cairano, S., Tseng, H., Bernardini, D., and Bemporad, A. Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain. *IEEE Transactions on Control Systems Technology*, 21(4): 1236–1248, 2013.
- Dickmanns, E., Behringer, R., Dickmanns, D., Hildebrandt, T., Maurer, M., Thomanek, F., and Schiehlen, J. The seeing passenger car VaMoRs-P. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 68–73, 1994.
- Dickmanns, E. D. Vehicles capable of dynamic vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 97, pages 1577–1592, 1997.
- Dickmanns, E. D. and Graefe, V. Dynamic monocular machine vision. *Machine vision and applications*, 1(4):223–240, 1988.
- Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- Dubins, L. 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):497–516, 1957.
- Dubiński, J. Sustainable development of mining mineral resources. *Journal of Sustainable Mining*, 12(1):1–6, 2013.
- Eggert, J., Damerow, F., and Klingelschmitt, S. The foresighted driver model. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 322–329, 2015.
- Eilers, S., Mårtensson, J., Pettersson, H., Pillado, M., Gallegos, D., Tobar, M., Johansson, K. H., Ma, X., Friedrichs, T., and Borojeni, S. S. COMPANION – Towards co-operative platoon management of heavy-duty vehicles. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 1267–1273, 2015.
- Elfes, A. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

- Englund, C., Chen, L., Ploeg, J., Semsar-Kazerooni, E., Voronov, A., Bengtsson, H. H., and Didoff, J. The grand cooperative driving challenge 2016: Boosting the introduction of cooperative automated vehicles. *IEEE Wireless Communications*, 23(4):146–152, 2016.
- Enzweiler, M. and Gavrila, D. A multilevel mixture-of-experts framework for pedestrian classification. *IEEE Transactions on Image Processing*, 20(10):2967–2979, 2011.
- Erlien, S. M., Funke, J., and Gerdes, J. C. Incorporating non-linear tire dynamics into a convex approach to shared steering control. In *Proceedings of the IEEE American Control Conference*, pages 3468–3473, 2014.
- Erlien, S. M., Fujita, S., and Gerdes, J. C. Shared steering control using safe envelopes for obstacle avoidance and vehicle stability. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):441–451, 2016.
- Ernst and Young. Who's in the driving seat? How the rise of autonomous vehicles will transform the relationship between man and car. Technical report, 2015.
- European Commission. Roadmap to a single european transport area. Technical report, 2011. URL [http://ec.europa.eu/transport стратегии/facts-and-figures/index\\_en.htm](http://ec.europa.eu/transport стратегии/facts-and-figures/index_en.htm). Retrieved April 12, 2018.
- European Commission. Attitudes of europeans towards urban mobility. Technical report, 2013.
- European Commission. EU transport in figures – Statistical pocketbook 2017. Technical report, 2017.
- European Environment Agency. Electric vehicles in Europe. Technical report, 2016.
- European Road Safety Observatory. Traffic safety basic facts – heavy goods vehicles and buses. Technical report, 2016.
- European Telecommunications Standards Institute. Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band. Technical report, 2012.
- Eurostat. Passenger cars in the EU. Electronic, 2017. URL [http://ec.europa.eu/eurostat/statistics-explained/index.php/Passenger\\_cars\\_in\\_the\\_EU](http://ec.europa.eu/eurostat/statistics-explained/index.php/Passenger_cars_in_the_EU). Retrieved April 12, 2018.
- Evestedt, N. *Sampling Based Motion Planning for Heavy-Duty Autonomous Vehicles*. Licentiate thesis, Linköping University, Linköping, Sweden, 2016.
- Evestedt, N., Axehill, D., Trincavelli, M., and Gustafsson, F. Sampling recovery for closed-loop rapidly expanding random tree using brake profile regeneration. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 101–106, 2015.

- Evestedt, N., Ward, E., Folkesson, J., and Axehill, D. Interaction aware trajectory planning for merge scenarios in congested traffic situations. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 465–472, 2016.
- Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., and Hrovat, D. Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology*, 15(3):566–580, 2007a.
- Falcone, P., Tufo, M., Borrelli, F., Asgari, J., and Tseng, H. E. A linear time-varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2980–2985, 2007b.
- Falcone, P., Borrelli, F., Tseng, E., Asgari, J., and Hrovat, D. A hierarchical model predictive control framework for autonomous ground vehicles. In *Proceedings of the IEEE American Control Conference*, pages 3719–3724, 2008a.
- Falcone, P., Borrelli, F., Tseng, H., Asgari, J., and Hrovat, D. Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation. *Int J Robust Nonlinear Control*, 18(8):862–875, 2008b.
- Falcone, P., Tseng, E., Borrelli, F., Asgari, J., and Hrovat, D. MPC-based yaw and lateral stabilisation via active front steering and braking. *Vehicle System Dynamics*, 46(S1):611–628, 2008c.
- Fassbender, D., Heinrich, B. C., and Wuensche, H.-J. Motion planning for autonomous vehicles in highly constrained urban environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4708–4713, 2016.
- Federal Highway Administration. The dream of an automated highway. Technical report, U. S. Department of Transportation, 2007.
- Ferguson, D. and Stentz, A. Field D\*: An interpolation-based path planner and replanner. In *Robotics research*, pages 239–253. Springer, 2007.
- Ferguson, D., Howard, T. M., and Likhachev, M. Motion planning in urban environments. *Journal of Field Robotics*, 25(11-12):939–960, 2008.
- Ferreau, H., Kirches, C., Potschka, A., Bock, H., and Diehl, M. qpOASES: A parametric active-set algorithm for quadratic programming. *Math Program Comp*, 6(4):327–363, 2014.
- Fisher, B. S. and Schnittger, S. Autonomous and remote operation technologies in the mining industry: Benefits and cost. Technical report, BAEconomics, 2012.

- Forbes. Self-driving cars are coming, 2014. URL <https://www.forbes.com/sites/peterdiamandis/2014/10/13/self-driving-cars-are-coming/>. 457a23934b86. Retrieved June 6, 2018.
- Ford Motor Company. URL <https://corporate.ford.com/innovation/autonomous-2021.html>. Retrieved April 23, 2018.
- Frasch, J. V., Gray, A., Zanon, M., Ferreau, H. J., Sager, S., Borrelli, F., and Diehl, M. An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles. In *Proceedings of the IEEE European Control Conference*, pages 4136–4141, 2013.
- Funke, J. and Gerdes, J. C. Simple clothoid lane change trajectories for automated vehicles incorporating friction constraints. *Journal of Dynamic Systems, Measurement, and Control*, 138(2):021002–021002–9, 2016.
- Funke, J., Theodosis, P., Hindiyeh, R., Stanek, G., Kritatakirana, K., Gerdes, J. C., Langer, D., Hernandez, M., Muller-Bessler, B., and Huhnke, B. Up to the limits: Autonomous Audi TTS. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 541–547, 2012.
- Funke, J., Brown, M., Erlien, S. M., and Gerdes, J. C. Collision avoidance and stabilization for autonomous vehicles in emergency scenarios. *IEEE Transactions on Control Systems Technology*, 25(4):1204–1216, 2017.
- Galceran, E., Cunningham, A. G., Eustice, R. M., and Olson, E. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction. In *Robotics: Science and Systems*, pages 2290–2297, 2015.
- Gao, Y., Gray, A., Frasch, J., Lin, T., Tseng, E., Hedrick, J., and Borrelli, F. Spatial predictive control for agile semi-autonomous ground vehicles. In *Proceedings of the International Symposium on Advanced Vehicle Control*, 2012.
- Garcia, C., Prett, D., and Morari, M. Model predictive control: Theory and practice – A survey. *Automatica*, 25(3):335–348, 1989.
- Ge, D., Jiang, X., and Ye, Y. A note on the complexity of  $\ell_p$  minimization. *Mathematical Programming*, 129(2):285–299, 2011.
- General Motors Company. URL <https://getcruise.com/>. Retrieved April 23, 2018.
- Gerdts, M., Karrenberg, S., Müller-Beßler, B., and Stock, G. Generating locally optimal trajectories for an automatically driven car. *Optimization and Engineering*, 10(4):439, 2009.

- Glaser, S., Vanholme, B., Mammar, S., Gruyer, D., and Nouveliere, L. Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction. *IEEE Transactions on Intelligent Transportation Systems*, 11(3):589–606, 2010.
- Goldbeck, J., Hürtgen, B., Ernst, S., and Kelch, L. Lane following combining vision and DGPS. *Image Vision Comput.*, 18(5):425–433, 2000.
- González, D., Pérez, J., Milanés, V., and Nashashibi, F. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1135–1145, 2016.
- Goodwin, G. C. and Sin, K. S. *Adaptive filtering prediction and control*. Courier Corporation, 2014.
- Götte, C., Keller, M., Rösmann, C., Nattermann, T., Haß, C., Glander, K.-H., Seewald, A., and Bertram, T. A real-time capable model predictive approach to lateral vehicle guidance. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 1908–1913, 2016.
- Graf Plessen, M., Bernardini, D., Esen, H., and Bemporad, A. Spatial-based predictive control and geometric corridor planning for adaptive cruise control coupled with obstacle avoidance. *IEEE Transactions on Control Systems Technology*, pages 1–13, 2017.
- Grant, M. and Boyd, S. CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, 2014.
- Gu, T., Snider, J., Dolan, J. M., and Lee, J.-W. Focused trajectory planning for autonomous on-road driving. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 547–552, 2013.
- Guanetti, J., Kim, Y., and Borrelli, F. Control of connected and automated vehicles: State of the art and future challenges. *Annual Reviews in Control*, 2018.
- Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2016. URL <http://www.gurobi.com>.
- Hart, P. E., Nilsson, N. J., and Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- Harvard Business Review. Driverless cars will change auto insurance. Here's how insurers can adapt, 2017. URL <https://hbr.org/2017/12/driverless-cars-will-change-auto-insurance-heres-how-insurers-can-adapt>. Retrieved April 25, 2018.

- Heald, M. Rational approximations for the fresnel integrals. *Mathematics of Computation*, 44:459–461, 1985.
- Hedrick, J., Tomizuka, M., and Varaiya, P. Control issues in automated highway systems. *IEEE Control Systems Magazine*, 14(6):21–32, 1994.
- hwan Jeon, J., Karaman, S., and Frazzoli, E. Anytime computation of time-optimal off-road vehicle maneuvers using the RRT\*. In *Proceedings of the IEEE Conference on Decision and Control and IEEE European Control Conference*, pages 3276–3282, 2011.
- IEEE-Standards Association. IEEE Standard for information technology – Amendment 6: Wireless access in vehicular environments. Technical report, 2010.
- International Transport Forum. Managing the transition to driverless road freight transport. Technical report, Organisation for Economic Co-operation and Development, 2017a.
- International Transport Forum. ITF transport outlook 2017. Technical report, Organisation for Economic Co-operation and Development, 2017b.
- Jiang, D. and Delgrossi, L. IEEE 802.11p: Towards an international standard for wireless access in vehicular environments. In *Proceedings of the IEEE Vehicular Technology Conference*, pages 2036–2040, 2008.
- Jiang, Z.-P. and Nijmeijer, H. Tracking control of mobile robots: A case study in backstepping. *Automatica*, 33(7):1393–1399, 1997.
- Jo, K., Kim, J., Kim, D., Jang, C., and Sunwoo, M. Development of autonomous car – Part I: Distributed system architecture and development process. *IEEE Transactions on Industrial Electronics*, 61(12):7131–7140, 2014.
- Jo, K., Kim, J., Kim, D., Jang, C., and Sunwoo, M. Development of autonomous car – Part II: A case study on the implementation of an autonomous driving system based on distributed architecture. *IEEE Transactions on Industrial Electronics*, 62(8):5119–5132, 2015.
- Kammel, S., Ziegler, J., Pitzer, B., Werling, M., Gindele, T., Jagzent, D., Schröder, J., Thuy, M., Goebel, M., and Hundelshausen, F. Team AnnieWAY’s autonomous system for the 2007 DARPA Urban Challenge. *Journal of Field Robotics*, 25(9):615–639, 2008.
- Kanayama, Y. and Hartman, B. Smooth local path planning for autonomous vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1265–1270, 1989.
- Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T. A stable tracking control method for an autonomous mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 384–389, 1990.

- Karaman, S. and Frazzoli, E. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011.
- Karaman, S., Walter, M. R., Perez, A., Frazzoli, E., and Teller, S. Anytime motion planning using the RRT. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1478–1483, 2011.
- Kato, S., Tsugawa, S., Tokuda, K., Matsui, T., and Fujii, H. Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications. *IEEE Transactions on Intelligent Transportation Systems*, 3(3):155–161, 2002.
- Katrakazas, C., Quddus, M., Chen, W.-H., and Deka, L. 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, 2015.
- Katriniok, A. and Abel, D. LTV-MPC approach for lateral vehicle guidance by front steering at the limits of vehicle dynamics. In *Proceedings of the IEEE Decision and Control and IEEE European Control Conference*, pages 6828–6833, 2011.
- Katriniok, A., Maschuw, J. P., Christen, F., Eckstein, L., and Abel, D. Optimal vehicle dynamics control for combined longitudinal and lateral autonomous vehicle guidance. In *Proceedings of the IEEE European Control Conference*, pages 974–979, 2013.
- Kim, E., Kim, J., and Sunwoo, M. Model predictive control strategy for smooth path tracking of autonomous vehicles with steering actuator dynamics. *International Journal of Automotive Technology*, 15(7):1155–1164, 2014.
- Kirchner, A. and Heinrich, T. Model based detection of road boundaries with a laser scanner. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 93–98, 1998.
- Kogan, D. and Murray, R. Optimization-based navigation for the DARPA Grand Challenge. In *Proceeding of the IEEE Conference on Decision and Control*, 2006.
- Kolmanovsky, I. and Gilbert, E. G. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering*, 4(4):317–367, 1998.
- Komatsu Ltd. Autonomous haul system. URL <http://www.komatsu.com.au/AboutKomatsu/Technology/Pages/AHS.aspx>. Retrieved April 24, 2018.
- Kong, J., Pfeiffer, M., Schildbach, G., and Borrelli, F. Kinematic and dynamic vehicle models for autonomous driving control design. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1094–1099, 2015.

- KPMG. Self-driving cars: The next revolution. Technical report, Center for Automotive Research, Transportation Systems Analysis Group, 2012.
- Kritayakirana, K. and Gerdès, J. C. Using the centre of percussion to design a steering controller for an autonomous race car. *Vehicle System Dynamics*, 50(1): 33–51, 2012a.
- Kritayakirana, K. and Gerdès, J. C. Autonomous vehicle control at the limits of handling. *International Journal of Vehicle Autonomous Systems*, 10(4):271–296, 2012b.
- Kühn, W. *Fundamentals of road design*, volume 20. WIT Press, 2013.
- Kuwata, Y., Teo, J., Fiore, G., Karaman, S., Frazzoli, E., and How, J. P. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 17(5):1105–1118, 2009.
- Kvasnica, M., Grieder, P., Baotić, M., and Morari, M. Multi-parametric toolbox (MPT). In *International Workshop on Hybrid Systems: Computation and Control*, pages 448–462. Springer, 2004.
- Labakhua, L., Nunes, U., Rodrigues, R., and Leite, F. S. Smooth trajectory planning for fully automated passengers vehicles: Spline and clothoid based methods and its simulation. In *Informatics in Control Automation and Robotics*, pages 169–182. Springer, 2008.
- LaValle, S. M. Rapidly-exploring random trees: A new tool for path planning. Technical report, Iowa State University, 1998.
- LaValle, S. M. *Planning algorithms*. Cambridge University Press, 2006.
- Lefèvre, S., Vasquez, D., and Laugier, C. A survey on motion prediction and risk assessment for intelligent vehicles. *Robomech Journal*, 1(1):1, 2014.
- Levinson, J., Montemerlo, M., and Thrun, S. Map-based precision vehicle localization in urban environments. In *Robotics: Science and Systems*, volume 4, page 1, 2007.
- Lewis, F. L. and Syrmos, V. L. *Optimal control*. John Wiley & Sons, 1995.
- Li, L., Huang, W.-L., Liu, Y., Zheng, N.-N., and Wang, F.-Y. Intelligence testing for autonomous vehicles: A new approach. *IEEE Transactions on Intelligent Vehicles*, 1(2):158–166, 2016.
- Liang, K.-Y. *Fuel-Efficient Heavy-Duty Vehicle Platoon Formation*. PhD thesis, KTH Royal Institute of Technology, Stockholm, 2016.

- Liebner, M., Klanner, F., Baumann, M., Ruhhammer, C., and Stiller, C. Velocity-based driver intent inference at urban intersections in the presence of preceding vehicles. *IEEE Intelligent Transportation Systems Magazine*, 5(2):10–21, 2013.
- Lim, C., Baumgarten, R., and Colton, S. Evolving behaviour trees for the commercial game defcon. In *Applications of evolutionary computation*, pages 100–110. Springer, 2010.
- Lima, P. F., Trincavelli, M., Mårtensson, J., and Wahlberg, B. Clothoid-based speed profiler and control for autonomous driving. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 2194 – 2199, 2015a.
- Lima, P. F., Trincavelli, M., Mårtensson, J., and Wahlberg, B. Clothoid-based model predictive control for autonomous driving. In *Proceedings of the IEEE European Control Conference*, pages 2983 – 2990, 2015b.
- Lima, P. F., Nilsson, M., Trincavelli, M., Mårtensson, J., and Wahlberg, B. Experimental evaluation of SA-MPC on the high-speed test track, 2016a. URL <http://doi.org/10.5281/zenodo.1292422>.
- Lima, P. F., Nilsson, M., Trincavelli, M., Mårtensson, J., and Wahlberg, B. Experimental evaluation of SA-MPC on the precision test track, 2016b. URL <http://doi.org/10.5281/zenodo.1292426>.
- Lima, P. F., Trincavelli, M., Nilsson, M., Mårtensson, J., and Wahlberg, B. Experimental evaluation of economic model predictive control for an autonomous truck. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 710–715, 2016c.
- Lima, P. F., Mårtensson, J., and Wahlberg, B. Stability conditions for linear time-varying model predictive control in autonomous driving. In *Proceedings of IEEE Conference on Decision and Control*, pages 2775–2782, 2017a.
- Lima, P. F., Nilsson, M., Trincavelli, M., Mårtensson, J., and Wahlberg, B. Spatial model predictive control for smooth and accurate steering of an autonomous truck. *IEEE Transactions on Intelligent Vehicles*, 2(4):238–250, 2017b.
- Lima, P. F., Oliveira, R., Mårtensson, J., and Wahlberg, B. Minimizing long vehicles overhang exceeding the drivable surface via convex path optimization. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 1374–1381, 2017c.
- Lima, P. F., Collares Pereira, G., Mårtensson, J., and Wahlberg, B. Progress maximization model predictive controller. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, 2018a. To appear.

- Lima, P. F., Collares Pereira, G., Mårtensson, J., and Wahlberg, B. Experimental validation of model predictive control stability for autonomous driving. *Control Engineering Practice*, 2018b. To appear.
- Lima, P. F., Collares Pereira, G., Mårtensson, J., and Wahlberg, B. Experimental evaluation LTV-MPC stability for autonomous driving, 2017d. URL <http://doi.org/10.5281/zenodo.1292428>.
- Lindner, F., Kressel, U., and Kaelberer, S. Robust recognition of traffic signals. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 49–53, 2004.
- Liniger, A., Domahidi, A., and Morari, M. Optimization-based autonomous racing of 1:43 scale RC cars. *Optimal Control Applications and Methods*, 36(5):628–647, 2015.
- Ljungqvist, O., Evestedt, N., Cirillo, M., Axehill, D., and Holmer, O. Lattice-based motion planning for a general 2-trailer system. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 819–824, 2017.
- Löfberg, J. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.
- Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.
- Magnusson, M. *The three-dimensional normal-distributions transform: An efficient representation for registration, surface analysis, and loop detection*. PhD thesis, Örebro Universitet, 2009.
- Manz, M., von Hundelshausen, F., and Wuensche, H.-J. A hybrid estimation approach for autonomous dirt road following using multiple clothoid segments. In *Proceeding of the IEEE International Conference on Robotics and Automation*, pages 2410–2415, 2010.
- Marzbani, H., Jazar, R., and Fard, M. Better road design using clothoids. In *Sustainable Automotive Technologies 2014*, pages 25–40. Springer International Publisher, 2015. ISBN 978-3-319-17998-8.
- Marzinotto, A., Colledanchise, M., Smith, C., and Ogren, P. Towards a unified behavior trees framework for robot control. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 5420–5427, 2014.
- Mattingley, J. and Boyd, S. CVXGEN: A code generator for embedded convex optimization. *Optimization and Engineering*, 13:1–27, 2012.
- Maurer, M., Behringer, R., Furst, S., Thomanek, F., and Dickmanns, E. D. A compact vision system for road vehicle guidance. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 3, pages 313–317, 1996.

- Mayne, D., Rawlings, J., Rao, C., and Scokaert, P. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.
- Mayne, D. Q., Seron, M. M., and Raković, S. V. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.
- McCrae, J. and Singh, K. Sketch-based interfaces and modeling (SBIM): Sketching piecewise clothoid curves. *Computers and Graphics*, 33(4):452–461, 2009.
- McKinsey. Self-driving car technology: When will the robots hit the road?, 2017. URL <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/self-driving-car-technology-when-will-the-robots-hit-the-road>. Retrieved June 6, 2018.
- Micaelli, A. and Samson, C. *Trajectory tracking for unicycle-type and two-steering-wheels mobile robots*. PhD thesis, Institut National de Recherche en Informatique et en Automatique, 1993.
- Mining Magazine. Feeling tired? Technical report, 2013.
- MIT Technology Review. Mining 24 hours a day with robots, 2016. URL <https://www.technologyreview.com/s/603170/mining-24-hours-a-day-with-robots/>. Retrieved June 6, 2018.
- Montemerlo, M. et al. Junior: The Stanford entry in the Urban Challenge. *Journal of Field Robotics*, 25(9):569–597, 2008.
- Montés, N., Mora, M. C., and Tornero, J. Trajectory generation based on rational Bezier curves as clothoids. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 505–510, 2007.
- Montés, N., Herraez, A., Armesto, L., and Tornero, J. Real-time clothoid approximation by rational Bézier curves. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2246–2251, 2008.
- Nevada State Law. AB 511 Section 8. 2011.
- Nilsson, J., Falcone, P., Ali, M., and Sjöberg, J. Receding horizon maneuver generation for automated highway driving. *Control Engineering Practice*, 41:124–133, 2015.
- Nilsson, N. J. A mobile automaton: An application of artificial intelligence techniques. Technical report, Artificial Intelligence Center - SRI International, 1969.
- Nothdurft, T., Hecker, P., Ohl, S., Saust, F., Maurer, M., Reschka, A., and Bohmer, J. Stadtpilot: First fully autonomous test drives in urban traffic. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 919–924, 2011.

- Oliveira, R., Cirillo, M., Mårtensson, J., and Wahlberg, B. Combining lattice-based planning and path optimization in autonomous heavy-duty vehicle applications. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2018a. To appear.
- Oliveira, R., Lima, P. F., Cirillo, M., Mårtensson, J., and Wahlberg, B. Curvature and sharpness continuous Dubins-like paths. In *Proceedings of the IEEE European Control Conference*, 2018b. To appear.
- Oriolo, G., De Luca, A., and Vendittelli, M. WMR control via dynamic feedback linearization: Design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, 10(6):835–852, 2002.
- Pacejka, H. *Tyre and vehicle dynamics*. Elsevier, 2nd edition, 2005.
- Paden, B., Čáp, M., Yong, S. Z., Yershov, D., and Frazzoli, E. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55, 2016.
- Park, H. and Gerdes, J. C. Optimal tire force allocation for trajectory tracking with an over-actuated vehicle. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1032–1037, 2015.
- Parreira, J. *An Interactive Simulation Model to Compare an Autonomous Haulage Truck System with a Manually-Operated System*. PhD thesis, University of British Columbia, Vancouver, Canada, 2013.
- Perrollaz, M., Spalanzani, A., and Aubert, D. Probabilistic representation of the uncertainty of stereo-vision and application to obstacle detection. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 313–318, 2010.
- Petrov, P. and Nashashibi, F. Modeling and nonlinear adaptive control for autonomous vehicle overtaking. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1643–1656, 2014.
- Pew Research Center. Automation in everyday life. Technical report, 2017.
- Pivtoraiko, M., Knepper, R., and Kelly, A. Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics*, 26(3):308–333, 2009.
- Plessen, M. G., Lima, P. F., Mårtensson, J., Bemporad, A., and Wahlberg, B. Trajectory planning under vehicle dimension constraints using sequential linear programming. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 108–113, 2017.
- Pluymers, B., Rossiter, J. A., Suykens, J. A. K., and De Moor, B. The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty. In *Proceedings of the IEEE American Control Conference*, pages 804–809, 2005.

- Pomerleau, D. and Jochem, T. Rapidly adapting machine vision for automated vehicle steering. *IEEE expert*, 11(2):19–27, 1996.
- Port of Rotterdam. The robot is coming, 2016. URL <https://www.portofrotterdam.com/en/cargo-industry/50-years-of-containers/the-robot-is-coming>. Retrieved April 12, 2018.
- Qian, X., Altché, F., Bender, P., Stiller, C., and de La Fortelle, A. Optimal trajectory planning for autonomous driving integrating logical constraints: An MIQP perspective. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 205–210, 2016.
- Rajamani, R. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.
- Reeds, J. and Shepp, L. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393, 1990.
- Rice, R. S. Measuring car-driver interaction with the g-g diagram. Technical report, SAE Technical Paper, 1973.
- Richalet, J., Testud, J., and Papon, J. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5):413–428, 1978.
- Robinson, T., Chan, E., and Coelingh, E. Operating platoons on public motorways: An introduction to the SARTRE platooning programme. In *Proceedings of the World Congress on Intelligent Transport Systems*, volume 1, page 12, 2010.
- Rosolia, U., De Bruyne, S., and Alleyne, A. G. Autonomous vehicle control: A nonconvex approach for obstacle avoidance. *IEEE Transactions on Control Systems Technology*, 25(2):469–484, 2017.
- Ross, P. Robot, you can drive my car. *IEEE Spectrum*, 51(6):60–90, 2014.
- SAE. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *SAE International*, 2016.
- Samson, C. Velocity and torque feedback control of a nonholonomic cart. In *Advanced Robot Control*, pages 125–151. Springer, 1991.
- Samson, C. 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, 1995.
- Samson, C. and Ait-Abderrahim, K. Feedback control of a nonholonomic wheeled cart in cartesian space. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1136–1141, 1991.

- Sánchez-Reyes, J. and Chacón, J. M. Polynomial approximation to clothoids via s-power series. *Computer-Aided Design*, 35(14):1305–1313, 2003.
- Sandvik. Sanvik automation. URL <https://www.rocktechnology.sandvik/en/products/automation>. Retrieved April 24, 2018.
- Scheuer, A. and Fraichard, T. Planning continuous-curvature paths for car-like robots. In *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1304–1311, 1996.
- Schlechtriemen, J., Wirthmueller, F., Wedel, A., Breuel, G., and Kuhnert, K.-D. When will it change the lane? A probabilistic regression approach for rarely occurring events. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1373–1379, 2015.
- Schneider, L., Cordts, M., Rehfeld, R., Pfeiffer, D., Enzweiler, M., Franke, U., Pollefeyt, M., and Roth, S. Semantic stixels: Depth is not enough. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 110–117, 2016.
- Schreiber, M., Knoppel, C., and Franke, U. Laneloc: Lane marking based localization using highly accurate maps. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 449–454, 2013.
- Schwarting, W., Alonso-Mora, J., and Rus, D. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:187–210, 2018.
- Securing America’s Future Energy. Heavy-duty innovation: Energy, automation, and technology in the trucking sector. Technical report, 2017.
- Shladover, S., Desoer, C., Hedrick, J., Tomizuka, M., Walrand, J., Zhang, W., McMahon, D., Peng, H., Sheikholeslam, S., and McKeown, N. Automated vehicle control developments in the PATH program. *IEEE Transactions on Vehicular Technology*, 40(1):114–130, 1991.
- Society of Automotive Engineers. Dedicated short range communications (DSRC) message set dictionary. Technical report, 2017.
- Takahashi, A. and Ninomiya, Y. Model-based lane recognition. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 201–206, 1996.
- Talbot, A. *The railway transition spiral*. Engineering News Publishing Company, 1904.
- Tesla Inc. Autopilot. URL <https://www.tesla.com/autopilot>. Retrieved April 24, 2018.

- The Economist. If autonomous vehicles rule the world – From horseless to driverless, 2015. URL <http://worldif.economist.com/article/12123/horseless-driverless>. Retrieved April 13, 2018.
- The Guardian. Truck drivers like me will soon be replaced by automation. You're next, 2017. URL <https://www.theguardian.com/commentisfree/2017/nov/17/truck-drivers-automation-tesla-elon-musk>. Retrieved April 25, 2018.
- The Guardian. Tesla car that crashed and killed driver was running on Autopilot, firm says, 2018a. URL <https://www.theguardian.com/technology/2018/mar/31/tesla-car-crash-autopilot-mountain-view>. Retrieved April 24, 2018.
- The Guardian. Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian, 2018b. URL <https://www.theguardian.com/technology/2018/mar/19/uber-self-driving-car-kills-woman-arizona-tempo>. Retrieved April 24, 2018.
- Thorpe, C., Hebert, M., Kanade, T., and Shafer, S. Vision and navigation for the Carnegie-Mellon NavLab. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):362–373, 1988.
- 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., 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. Stanley: The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9):661–692, 2006.
- Times, F. Trucks headed for a driverless future, 2018. URL <https://www.ft.com/content/7686ea3e-e0dd-11e7-a0d4-0944c5f49e46>. Retrieved April 25, 2018.
- Timings, J. P. and Cole, D. J. Minimum maneuver time calculation using convex optimization. *Journal of Dynamic Systems, Measurement, and Control*, 135(3):031015, 2013.
- Torrey, I. and Ford, W. Cost of congestion to the trucking industry: 2017 Update. 2017.
- Tsugawa, S., Yatabe, T., Hirose, T., and Matsumoto, S. An automobile with artificial intelligence. In *Proceedings of the international joint conference on Artificial Intelligence*, pages 893–895, 1979.
- Tsugawa, S., Kato, S., and Aoki, K. An automated truck platoon for energy saving. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4109–4114, 2011.

- Turri, V., Carvalho, A., Tseng, E., Johansson, K., and Borrelli, F. Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 378–383, 2013.
- Ulbrich, S. and Maurer, M. Probabilistic online POMDP decision making for lane changes in fully automated driving. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 2063–2067, 2013.
- United Nations. Paris agreement. Technical report, 2015. URL [https://unfccc.int/sites/default/files/english\\_paris\\_agreement.pdf](https://unfccc.int/sites/default/files/english_paris_agreement.pdf). Retrieved July 16, 2018.
- United Nations Population Division. World population prospects: The 2017 revision. Technical report, 2017.
- Urmson, C., Bagnell, J., Baker, C., Hebert, M., Kelly, A., Rajkumar, R., Rybski, P., Scherer, S., Simmons, R., and Singh, S. Tartan racing: A multi-modal approach to the DARPA Urban Challenge. Technical Report CMU-RI-TR-, Carnegie Mellon University, 2007.
- Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M., Dolan, J., Duggins, D., Galatali, T., and Geyer, C. Autonomous driving in urban environments: Boss and the Urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.
- Van Brummelen, J., O’Brien, M., Gruyer, D., and Najjaran, H. Autonomous vehicle perception: The technology of today and tomorrow. *Transportation Research Part C: Emerging Technologies*, 2018.
- van de Hoef, S. *Coordination of Heavy-Duty Vehicle Platooning*. PhD thesis, KTH Royal Institute of Technology, Stockholm, 2018.
- Van Loan, C. Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control*, 23(3):395–404, 1978.
- van Nunen, E., Kwakkernaak, R., Ploeg, J., and Netten, B. D. Cooperative competition for future mobility. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1018–1025, 2012.
- Varaiya, P. Smart cars on smart roads: Problems of control. *IEEE Transactions on Automatic Control*, 38(2):195–207, 1993.
- Vatavu, A. and Nedevschi, S. Real-time modeling of dynamic environments in traffic scenarios using a stereo-vision system. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 722–727, 2012.
- Velenis, E., Tsotras, P., and Lu, J. Modeling aggressive maneuvers on loose surfaces: The cases of trail-braking and pendulum-turn. In *Proceedings of the IEEE European Control Conference*, pages 1233–1240, 2007.

- Verge, T. It's 2018, so where are the self-driving cars?, 2018. URL <https://www.theverge.com/2018/1/30/16950726/state-of-autonomous-car-self-driving-ces-detroit-2018>. Retrieved April 13, 2018.
- Vivo, G. The SAFESPOT integrated project: An overview. In *Proceeding of the IEEE Intelligent Vehicles Symposium*, pages 14–14, 2007.
- Volkswagen AG. Autonomous Driving at the Push of a Button. URL <https://www.volkswagenag.com/en/news/stories/2017/03/autonomous-driving-at-the-push-of-a-button.html>. Retrieved May 26, 2018.
- Volvo Cars AB. Drive Me, a. URL <https://www.volvcars.com/intl/buy/explore/intellisafe/autonomous-driving/drive-me>. Retrieved May 26, 2018.
- Volvo Cars AB. Intellisafe Autopilot, b. URL <https://www.volvcars.com/bh/about/our-innovation-brands/intellisafe/intellisafe-autopilot>. Retrieved June 29, 2018.
- Vox. 5 big challenges that self-driving cars still have to overcome, 2016. URL <https://www.vox.com/2016/4/21/11447838/self-driving-cars-challenges-obstacles>. Retrieved July 6, 2018.
- Wallace, R. S., Stentz, A., Thorpe, C. E., Moravec, H. P., Whittaker, W., and Kanade, T. First results in robot road-following. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1089–1095, 1985.
- Walton, D. J. and Meek, D. S. A controlled clothoid spline. *Computers & Graphics*, 29(3):353–363, 2005.
- Wang, L. Z., Miura, K. T., Nakamae, E., Yamamoto, T., and Wang, T. J. An approximation approach of the clothoid curve defined in the interval  $[0, \pi/2]$  and its offset by free-form curves. *Computer-Aided Design*, 33(14):1049–1058, 2001.
- Ward, E. and Folkesson, J. Multi-classification of driver intentions in yielding scenarios. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 678–685, 2015.
- Ward, E. and Folkesson, J. Vehicle localization with low cost radar sensors. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 864–870, 2016.
- Waymo. URL <https://waymo.com/>. Retrieved June 29, 2018.
- Wei, J., Dolan, J. M., and Litkouhi, B. Autonomous vehicle social behavior for highway entrance ramp management. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 201–207, 2013.

- Werling, M. and Liccardo, D. Automatic collision avoidance using model-predictive online optimization. In *Proceedings of the IEEE Conference on Decision and Control*, pages 6309–6314, 2012.
- Werling, M., Gindele, T., Jagszent, D., and Groll, L. A robust algorithm for handling moving traffic in urban scenarios. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1108–1112, 2008.
- Wired. Audi’s robotic car climbs Pikes Peak, 2010. URL <https://www.wired.com/2010/11/audis-robotic-car-climbs-pikes-peak/>. Retrieved April 24, 2018.
- Wired. Robotcars could add USD 7 trillion to the global economy, 2017. URL <https://www.wired.com/2017/06/impact-of-autonomous-vehicles/>. Retrieved April 23, 2018.
- Wolcott, R. W. and Eustice, R. M. Visual localization within LiDAR maps for automated urban driving. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 176–183, 2014.
- World Health Organization. Global status report on road safety. Technical report, 2015.
- Wurm, K., Hornung, A., Bennewitz, M., Stachniss, C., and Burgard, W. Octomap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, 2010.
- Xu, W., Wei, J., Dolan, J. M., Zhao, H., and Zha, H. A real-time motion planner with trajectory optimization for autonomous vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2061–2067, 2012.
- Yoon, Y., Shin, J., Kim, H. J., Park, Y., and Sastry, S. Model-predictive active steering and obstacle avoidance for autonomous ground vehicles. *Control Engineering Practice*, 17(7):741–750, 2009.
- Ziegler, J., Bender, P., Dang, T., and Stiller, C. Trajectory planning for Bertha – a local, continuous method. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 450–457, 2014a.
- Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang, T., Franke, U., Appenrodt, N., and Keller, C. Making Bertha drive 2014 – An autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine*, 6(2):8–20, 2014b.