

# Cooperative Trajectory Planning for Automated Vehicles

***Citation for published version (APA):***

van Hoek, R. B. A. (2021). *Cooperative Trajectory Planning for Automated Vehicles*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mechanical Engineering]. Eindhoven University of Technology.

***Document status and date:***

Published: 06/07/2021

***Document Version:***

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

***Please check the document version of this publication:***

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

***General rights***

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

***Take down policy***

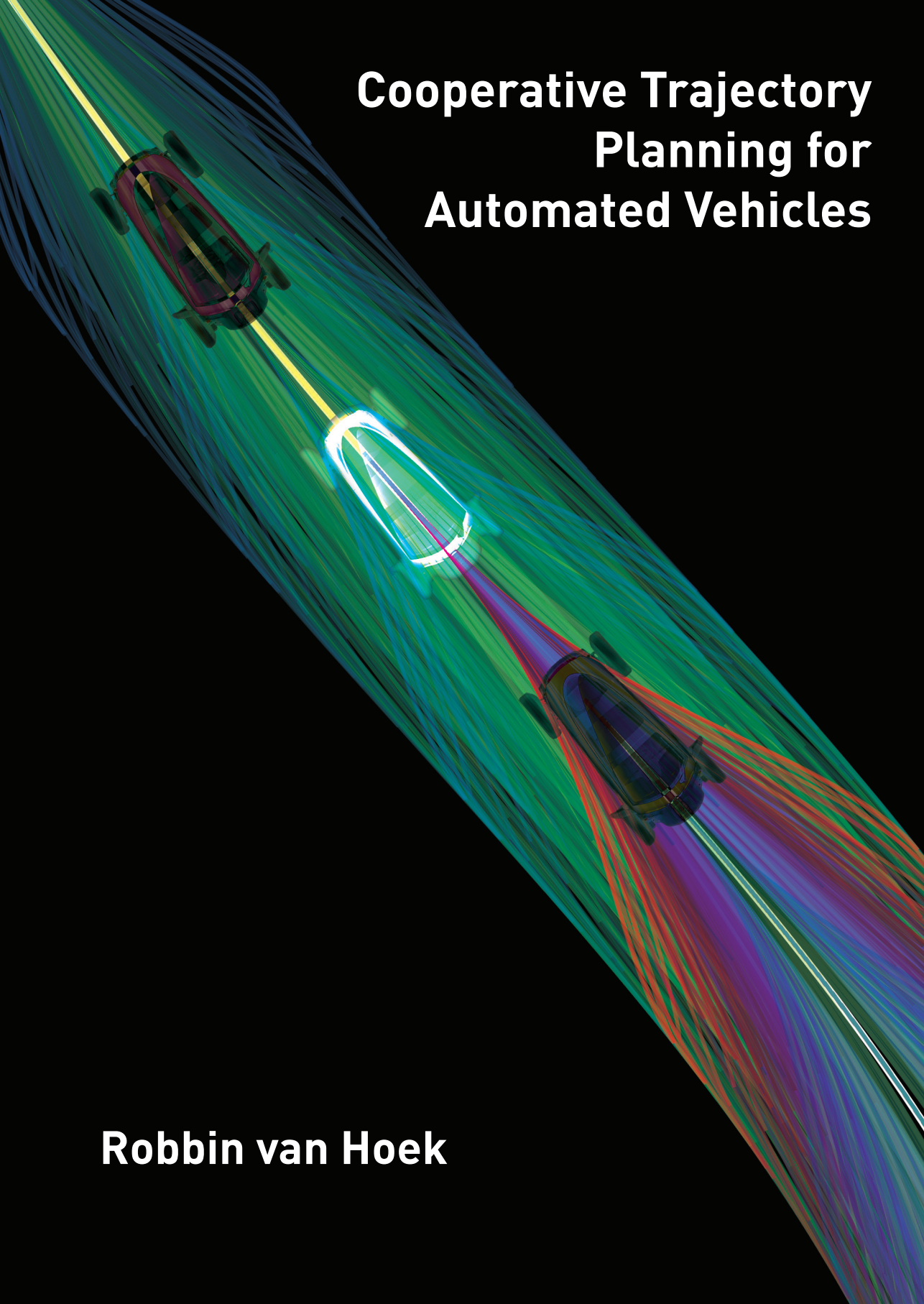
If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# Cooperative Trajectory Planning for Automated Vehicles

**Robbin van Hoek**



# Cooperative Trajectory Planning for Automated Vehicles

Robbin van Hoek



The author has successfully completed the educational program of the Graduate School of the Dutch Institute of Systems and Control (DISC).



The work described in this thesis is carried out in the Dynamics & Control group at the Department of Mechanical Engineering of the Eindhoven University of Technology. The work is supported by the Dutch Organization for Scientific Research (NWO). This work has been carried out NWO as part of the i-CAVE project with project number 14893.

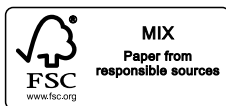
A catalogue record is available from the Eindhoven University of Technology Library

ISBN: 978-90-386-5283-2

Typeset by the author using  $\text{\LaTeX}$  2 $\epsilon$

Reproduction: Gildeprint - Enschede, the Netherlands

Cover design: Bart Thielen (step2design) & Robbin van Hoek





# Cooperative Trajectory Planning for Automated Vehicles

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de  
Technische Universiteit Eindhoven, op gezag van de  
rector magnificus prof.dr.ir. F.P.T Baaijens, voor een  
commissie aangewezen door het College voor  
Promoties, in het openbaar te verdedigen op  
dinsdag 6 juli 2021 om 13:30 uur

door

Robbin Bernard Adrianus van Hoek

geboren te Gilze en Rijen

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr. L.P.H. de Goey
promotor:	prof.dr. H. Nijmeijer
copromotor:	dr.ir. J. Ploeg
leden:	prof.dr.ir. B. Jayawardhana (Rijksuniversiteit Groningen)
	prof.dr.ir. P.W.A. Zegelaar
	dr.ir. R. Tóth
	G. Orosz Ph.D. (University of Michigan)

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

# Summary

Vehicle automation has become an important topic in recent years. It is aimed towards mitigating driver-induced traffic accidents, improving the road capacity of the existing infrastructure as well as reducing fuel consumption. Two major classes of automated vehicles can be distinguished. The first is the class of cooperative vehicles, which use vehicle-to-vehicle (V2V) communication, or vehicle-to-infrastructure (V2I) communication in order to exchange motion data and sometimes meta data such as intention. An example of cooperative vehicles that use inter-vehicle communication is Cooperative Adaptive Cruise Control (CACC), in which vehicles drive at very small inter-vehicle distances and use communicated inputs from preceding vehicles in order to maintain a desired spacing. The spacing policy and controllers of these vehicles aim for string stability, which is the property that disturbances attenuate in upstream direction of the vehicle platoon. This is pivotal in preventing traffic jams and thus for increasing road capacity. However, due to the pre-defined behaviour of vehicle following, the vehicle is only capable of driving in a limited number of scenarios. The second class concerns autonomous vehicles. This type of vehicle uses on-board sensors such as radar, LIDAR and computer vision systems in order to identify the road, other traffic participants, and other relevant features or obstacles. The control algorithms on board these vehicles make use of explicit planning of a vehicle trajectory, such that feasibility and collisions can be checked prior to committing to the planned motion. This allows autonomous vehicles to handle a much wider class of traffic scenarios, compared to cooperative vehicles. However, in contrast to cooperative vehicles, the framework of autonomous vehicles is typically not aimed towards obtaining a string-stable traffic system of multiple autonomous vehicles. Additionally, these classes of vehicles use different automation frameworks: Cooperative vehicles in platoons make use of feedforward and feedback control, whereas autonomous vehicles make use of explicit trajectory planning.

A promising way forward is to integrate the string stability objective and the

communication aspect of the cooperative approach in the framework of autonomous vehicles. This would combine the versatility of the autonomous vehicle with the road capacity benefits of the cooperative vehicle. Such a vehicle would be capable of navigating the roads autonomously, and when encountering other equipped vehicles, decide to start platooning. Additionally, it can decide to overtake vehicles (e.g., heavy duty road vehicles) if these vehicles drive too slow to start a time-efficient platoon, or decide to break up a platoon if the preceding vehicle demonstrates behavior that is undesired for the host vehicle. The development of a unified framework for both cooperative and autonomous vehicles is therefore the main focus of the work presented here.

This framework is designed to plan a trajectory relative to a reference path and reference velocity, which has been shown in literature to provide good results for autonomous vehicles. The trajectory planning is performed in a Frenet frame with respect to a nominal path, such that the vehicle will always follow the general direction of the road. B-splines are used to construct the planned trajectories, as these allow the trajectories to be efficiently communicated between vehicles since only a few parameters are required. This allows the required communication bandwidth to be low. The automated vehicles that utilize this framework simultaneously generate both cooperative as well as autonomous trajectories. This allows them to decide to break up the platoon at any moment if needed and instead utilize the autonomous trajectories. This vehicle could then potentially become, the lead vehicle of the newly formed platoon.

The generation of the autonomous trajectory of the automated vehicle is the lowest cost trajectory of a set of potential candidate trajectories. This set also always includes trajectories that perform emergency braking or lateral deviations to prevent possible collisions. A cost function is used to select the most-comfortable, feasible and collision free trajectory. The cost functions for both cooperative and autonomous trajectories are designed in such a way that the selected trajectories in consecutive planning cycles are similar within the class of B-spline functions. This is referred to as temporal consistency. Temporal consistency is important as the selected trajectories are communicated to following vehicles, which in turn generate the cooperative trajectories based on the received information.

For cooperative trajectories, timing is critical due to the small inter vehicle distance. The trajectories are constructed by means of a closed form computation, such that computation time can be guaranteed to satisfy the real time implementation requirements. Moreover, the communicated trajectories also include the time of construction on a common clock, which is obtained via the Global Positioning System (GPS). As a result, the cooperative vehicles are aware of the time delay between the planning of the preceding vehicle and the planning of the host vehicle, such that they can compensate for it.

A challenge of adopting the trajectory planning framework for cooperative

driving at short inter vehicle distances is the required computation time. The trajectories are typically updated 5 times per second, whereas distance information of the radar is typically updated around 20 times per second. To overcome the drawback of reduced control bandwidth, a method of time scaling is derived. This time scaling algorithm uses the measured inter vehicle distance to scale the time of the selected trajectory. In doing so, it allows the vehicle to slow down if needed, while maintaining the same geometric path as originally planned. Implementing the time scaling mechanism in the framework of the cooperative automated vehicle allows the use of the most up-to-date radar information available.

To validate this theoretical framework, two full-scale prototype demonstrator platforms have been developed in the form of two modified Renault Twizy's. The experiments performed with these demonstrator platforms show that the theoretical framework can be used for cooperative and automated driving at short inter-vehicle distances. Additionally, both the gap closing strategy as well as the time scaling mechanism are validated. The time scaling mechanism also showed promising results in overcoming system faults in the lead vehicle, where the lead vehicle would significantly deviate from its communicated trajectory.

Summarizing, this thesis focusses on the development of a unified framework including both cooperative as well as autonomous vehicles, as well as the experimental validation thereof. The framework allows a single vehicle to act both as an autonomous vehicle, or demonstrate vehicle-following behavior at short inter vehicle distances, while considering string stability. The flexibility of decision making of the autonomous vehicle is retained, such that versatile traffic scenarios can be handled. Experimental validation demonstrates that the developed framework has great potential for further improvement for the next generation of automated vehicles.



# Contents

<b>Summary</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Automated Vehicles	1
1.1.1 Autonomous Vehicles	3
1.1.2 Cooperative Vehicles	5
1.1.3 i-CAVE (Integrated Cooperative Automated VEHICLE)	8
1.2 Challenges in Trajectory Planning for Automated Cooperative Vehicles	9
1.3 Research objectives and Contributions	10
1.4 Outline	11
<b>2 Background</b>	<b>13</b>
2.1 Motion Planning	13
2.1.1 Literature Overview	13
2.1.2 Vehicle model	15
2.1.3 Motion Planning in Frenet Frame	18
2.1.4 Trajectory Generation	22
2.2 Splines	28
2.2.1 Bézier curves	28
2.2.2 B-splines	32
2.3 Reference Trajectory	38
2.3.1 Reference Path	38
2.3.2 Reference Velocity	42
2.4 Summary	45
<b>3 Cooperative Trajectory Planning</b>	<b>47</b>
3.1 Cooperative trajectory objective	48



3.1.1	Temporal Consistency . . . . .	49
3.1.2	String Stability and Collision Avoidance . . . . .	49
3.1.3	Spacing Policy . . . . .	50
3.2	Cooperative trajectory generation using polynomials . . . . .	51
3.2.1	Shift in Arrival Time . . . . .	53
3.3	Cooperative B-spline Trajectory Generation . . . . .	58
3.3.1	B-spline Trajectory Construction . . . . .	58
3.3.2	Design Considerations . . . . .	61
3.3.3	Communication / Planning Delay . . . . .	61
3.3.4	Simulation Results . . . . .	63
3.4	Gap Closing B-spline Trajectories . . . . .	72
3.4.1	Variable Spacing Policy . . . . .	72
3.4.2	Results . . . . .	77
3.5	Summary . . . . .	82
<b>4</b>	<b>Autonomous Planner and Combined Framework</b>	<b>83</b>
4.1	Autonomous Planner . . . . .	84
4.1.1	Longitudinal Trajectory Generation . . . . .	85
4.1.2	Discretized Search Space . . . . .	86
4.1.3	Lateral Trajectory Generation . . . . .	91
4.2	Combined Autonomous and Cooperative Trajectory Planning . . . . .	95
4.3	Summary . . . . .	103
<b>5</b>	<b>String Stability Analysis</b>	<b>105</b>
5.1	$\mathcal{L}_2$ -String Stability . . . . .	106
5.2	Discrete-Time Dynamics of Planning Algorithm . . . . .	108
5.2.1	Lead Vehicle . . . . .	111
5.2.2	Follower Vehicle . . . . .	113
5.3	String Stability in the Frequency Domain . . . . .	114
5.3.1	Coordinate Transformation . . . . .	115
5.3.2	Transfer Function Description . . . . .	117
5.3.3	Evaluation and Discussion . . . . .	120
5.4	Temporally Consistent Lead Vehicle . . . . .	122
5.4.1	Lead Vehicle Minimum Jerk . . . . .	123
5.4.2	Lead Vehicle Velocity Controlled Trajectory Planner . . . . .	130
5.4.3	Planning Delay for Following Vehicles . . . . .	132
5.5	Summary and Discussion . . . . .	137
<b>6</b>	<b>Experimental Implementation</b>	<b>139</b>
6.1	Experimental Setup . . . . .	139
6.2	Implementation Details . . . . .	141
6.2.1	Integration of Radar Information . . . . .	142
6.2.2	Most Important Object (MIO) Identification . . . . .	142
6.2.3	Drivetrain Input . . . . .	145

---

6.2.4	Initial Condition of Planning . . . . .	146
6.3	Lead Vehicle Experiments . . . . .	146
6.4	Cooperative Vehicle Following Experiments . . . . .	148
6.4.1	Planning Delay . . . . .	149
6.4.2	Gap Closing . . . . .	151
6.4.3	Vehicle Following . . . . .	154
6.5	Summary and Discussion . . . . .	159
<b>7</b>	<b>Time Scaling</b>	<b>161</b>
7.1	Scaling Approach . . . . .	161
7.2	Simulation Results . . . . .	164
7.3	Experimental Validation . . . . .	166
7.4	Summary and Discussion . . . . .	167
<b>8</b>	<b>Conclusions and Recommendations</b>	<b>169</b>
8.1	Conclusion . . . . .	170
8.2	Recommendations . . . . .	173
<b>A</b>	<b>Integral of the product of two B-splines</b>	<b>175</b>
<b>B</b>	<b>Planning Algorithm Properties</b>	<b>179</b>
	<b>Dankwoord</b>	<b>197</b>
	<b>Curriculum Vitae</b>	<b>199</b>



# Chapter 1

## Introduction

Vehicle automation has been a promising field of research to improve passenger and driver comfort, road safety, fuel consumption and traffic efficiency. Section 1.1 of this chapter first introduces the classes of automated vehicles and outlines their capabilities and limitations. Section 1.2 then identifies the main challenges in the field of cooperative trajectory planning, a promising solution to bridge the gap between the aforementioned classes of automated vehicles. Section 1.3 addresses these challenges by formulating objectives for this thesis and summarizes the main contributions, after which Section 1.4 presents an outline of the thesis.

### 1.1 Automated Vehicles

This section introduces the subclasses of vehicles within the group of automated vehicles and discusses some technical aspects of them and highlights important features. Vehicle automation has been an ongoing field of study for years that aims to increase road safety, reduce fuel consumption, or simplify the driver's task. The first applications of vehicle automation can be found in commercially available vehicles in the form of Anti-lock Braking Systems, Electronic Stability Control or Cruise Control. More modern applications that can be found in commercially available vehicles nowadays include lane keeping systems and adaptive cruise control. However, these systems are referred to as driver assistance systems, rather than automated vehicles since the driver remains responsible for the behavior of the vehicle. Following the taxonomy of the SAE [1], the term *Automated Vehicle* refers to a vehicle that has automated systems for both longitudinal as well as lateral actuation and where the driver is no longer responsible for the vehicle behavior. This is summarized in Table 1.1, which shows which parts of the Dynamic Driving Task (DDT) are taken care of

Table 1.1: SAE vehicle automation level definitions [1]

Level	Name	Dynamic Driving Task (DDT)		DDT fallback	Operational Design Domain (ODD)
		Motion Control	Object & Event Detection and Response (OEDR)		
Driver performs part or all of the DDT					
0	No driving automation	Driver	Driver	Driver	n/a
1	Driver assistance	Driver & System	Driver	Driver	Limited
2	Partial driving automation	System	Driver	Driver	Limited
ADS performs the entire DDT (while engaged)					
3	Conditional driving automation	System	System	Fallback-ready user	Limited
4	High driving automation	System	System	System	Limited
5	Full driving automation	System	System	System	Unlimited

by the automated system, and for which the driver remains responsible. It can be seen that from level three and up, the driver is no longer responsible for the DDT. Therefore, level three and up the vehicle is referred to as an *automated driving system* (ADS), rather than a *driving automation system*.

These automated vehicles can be separated into two classes of vehicles [2]. The first class are autonomous vehicles, which rely on on-board sensors to navigate a large variety of scenarios from urban streets to highways. The second class are the cooperative vehicles. Cooperative driving can be described as *influencing the individual vehicle behaviour to optimize the collective behaviour in terms of road throughput, fuel efficiency or safety* [3]. Typically this is achieved by means of information sharing over a wireless link. While both classes of vehicles make use of some type of automation, the underlying control configurations can be very different.

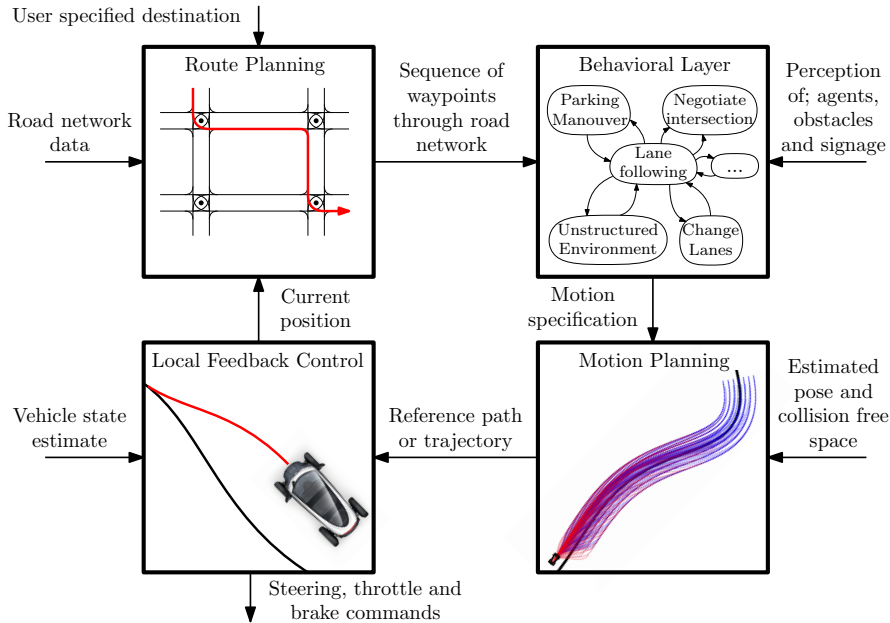


Figure 1.1: Typical layers in the decision making process of a self driving car.

### 1.1.1 Autonomous Vehicles

Autonomous vehicles are a class of vehicles which are capable of navigating an environment autonomously, by means of their on-board sensors. The decision making process of this complex task is often subdivided into hierarchical layers [4]. An overview of these layers is shown in Figure 1.1. The Routing layer chooses what route to follow and which roads to take, in order to get to the destination. The behavioural layer is responsible for interpreting the situation and deciding what type of planner to invoke. A motion planner then generates a feasible motion over a finite time horizon, according to the specification of the behavioural layer. Finally, a low level motion controller regulates the vehicle towards the planned motion.

When the specified motion is described purely geometrically, it is referred to as a *path* and its planner is referred to as a path planner. A separate layer then often determines the velocity with which the path needs to be driven. Examples which utilize such a planner can be found in [5] [6], which make use of Bézier curves to describe geometrical paths that circumvent obstacles and plan a path in narrow corridor. A disadvantage of these path planners is that it is very difficult to take into account dynamic obstacles, as the speed with which a path is executed, determines if a collision will occur or not. Therefore, determining

whether the path itself is feasible, is difficult to do without knowledge about the corresponding velocity. When a motion is parametrized in time, it is referred to as a *trajectory*, and the planning layer is referred to as a trajectory planner [4]. Trajectory planners can explicitly take into account the motion of obstacles, making it relatively straightforward to do collision and constraint checks to determine the feasibility of the trajectory. Both path and trajectory planning has also been studied extensively in the field of robotics, but real-time planning for high speed road vehicles has additional challenges that have led to a separate branch of research.

Two important events that had a major contribution to the development of autonomous vehicles are the DARPA grand challenge of 2005, and the DARPA Urban challenge of 2007. The DARPA grand challenge of 2005 was a competition for driverless cars in which an off-road course of 212 km had to be traversed in under 10 hours. The main challenge was high-speed road finding and obstacle detection and avoidance [7]. Also no dynamic obstacles would have to be handled by the robot vehicles. The winning team Stanford and their vehicle 'Stanley' made use of a two-stage path planner. The first layer was an offline path smoother that translates the waypoints and speed limits to a smoother path and more detailed speed limit. The second layer is an online path planning algorithm and consists of a series of candidate trajectories that laterally swerve or nudge the vehicle around obstacles or in the right direction.

Unlike the DARPA Grand Challenge, the DARPA Urban challenge involved driving in urban scenarios with dynamic obstacles, while all traffic rules needed to be obeyed. The finals consisted of three missions that traversed a 89 km course. The vehicles had to navigate more complex urban scenarios, which include pulling into a parking lot, crossing a four way intersection, merging into traffic, and defensive driving in which another road user is driving head on in the wrong lane. These challenges required sophisticated motion planning algorithms for driving on structured roads, as well as cluttered parking lots. The winning vehicle 'Boss' from Tartan racing used a lattice planner [8] [9]. The Stanford entry 'junior' that received second place, made use of a dynamic programming planner for global route planning and a set of trajectories that include lateral shift to perform normal on-road driving [10]. A separate planner for free-form navigation was used for path finding on parking lots and making u-turns in case of blocked roads [11]. In contrast, the MIT entry 'Talos' makes use of a Rapidly-exploring Random Tree (RRT) for both free-form navigation, as well as on-road driving. In the same competition different strategies were achieved to complete the same objectives. A comprehensive review of the various types of motion planners used in automated vehicles can be found in [12].

One very common solution is to make use of the a priori information of the road, to construct a reference path for the vehicle to follow in general. This is due to driving on a road being typically a fairly structured environment, due to the direction of travel along a given lane of the road. The road provides the vehicle



with a geometrical reference path and a narrow set of constraints in which to look for candidate trajectories [13]. A group of trajectory planners that makes use of this reference path operates by means of sampling. These planners first refine the reference path [14], before deterministically sampling either the configuration space or the input space of a stable closed loop representation of the vehicle [15]. These samples are used to construct candidate trajectories that follow the general direction of the reference path. The generated trajectories can then be checked for feasibility and collisions, after which the most optimal based on a given cost function (e.g., comfort and advancing along the route) is selected for execution. In case of sampling of the configuration space, the trajectories are often constructed in a moving reference frame, fixed to the reference path by means of polynomials [16], [17], [18]. This allows the vehicle to satisfy the direction of travel in the general direction of the road, while constraints on the width of the road are easily applied. Moreover, the quintic polynomials in [18] can be shown to minimize jerk and hence result in comfortable trajectories.

One disadvantage of these polynomial planners, is the difficulty in implementing more complex manoeuvres, such as multiple lateral swerves. A method that overcomes this drawback, makes use of graph searches, to efficiently find trajectories constructed by means of motion primitives. In [19], a spatio-temporal search graph is constructed in which the edges are represented by quintic splines. By including a time variable in the search graph, trajectories that are invalidated by dynamic obstacles can be easily eliminated. In [20] [21], a similar approach is adopted, in which a graph is constructed on-line by means of multiple lateral and longitudinal sampled configurations, that are connected by means of polynomials in the lateral direction, and piecewise continuous accelerations in the longitudinal direction. These papers also describe an adaptive cruise control type of strategy for following vehicles, where the acceleration is not piecewise continuous, but instead derived from a feedback controller.

These autonomous vehicles are highly versatile: Applications for autonomous vehicles include driving through the desert [7], driving through urban areas with also cyclists [18], but also cases of limit handling [22], where the vehicle operates near peak friction. For example, in [23] [24], a controller is presented that is used for the control of a racing vehicle, operating in the highly non-linear region of the tyres, while in [25], a controller is presented for autonomous drifting. An example more directly applicable to on-road driving is given in [26], which presents a model predictive collision avoidance planner. These examples emphasise the high variety of applications for the autonomous vehicle.

### 1.1.2 Cooperative Vehicles

In contrast to autonomous vehicles, cooperative vehicles aim to optimize the collective behavior of a group of vehicles. An enabler for this collective behavior

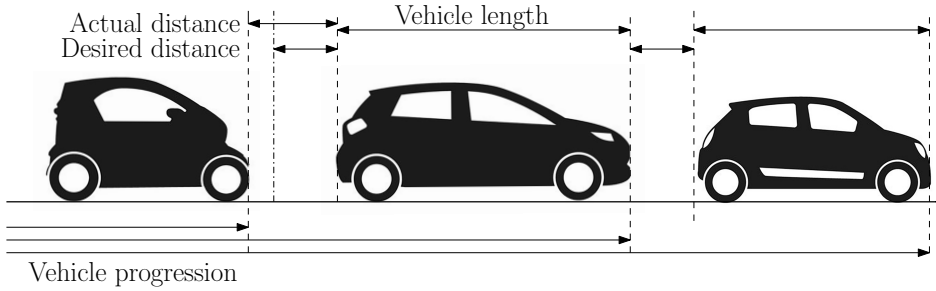


Figure 1.2: Schematic representation of a platoon. Actual inter-vehicle distance is regulated towards the desired inter-vehicle distance.

is communication via a wireless link. Using this wireless link, vehicles can share information about their current state, their intention and their perception, which provides a new information source that is beyond line-of-sight of individual vehicles.

This additional information leads to many new applications that are not feasible for autonomous vehicles. For example, [27] describes the use of vehicle following with short inter-vehicle distances, often referred to as *platooning*. An example of platooning is shown in Figure 1.2, which illustrates the objective of a desired inter-vehicle distance, as well as the actual distance. An important performance criterion in platooning is *string-stability* [28] [29] [30]. String stability refers to the attenuation of the effect of disturbances in the upstream direction of vehicle flow. This attenuation prevents amplification of both the spacing error and control input to the string of vehicles, which are a major contributor to traffic jams. In so-called 'ghost' traffic jams, a vehicle brakes and in response, the following vehicle brakes harder due to a delayed reaction time of the driver or control loop. When this continues upstream, vehicles eventually come to a complete stop and a traffic jam is formed without the road being utilized at full capacity. Another benefit of platooning is the potential to increase fuel efficiency, especially in trucks [31]. Other examples of cooperative driving are found in [32] [33], where platooning is extended to also include lateral vehicle following, where the latter eliminates the problem of corner cutting. In [34] [35], a method is presented that makes use of virtual platoons to handle intersections in a highly efficient manner, by mapping all incoming lanes to a virtual platoon.

Similar to the DARPA Challenge for autonomous vehicles, development in cooperative driving has also been accelerated due to projects and competitions. A number of such projects are listed in [36]. For example, the Californian PATH (Partners for Advanced Transportation Technology) pioneered research in platooning technology [29] [37]. More recently, the Grand Cooperative Driving Challenge (GCDC) of 2011 and 2016 contributed to the development of

cooperative driving technology, and identified directions for future research. The first GCDC was organised in 2011 with the aim to accelerate the introduction of cooperative vehicle technologies [38]. Two scenarios, were executed during the 2011 GCDC. The first was highway driving, in which the steady state behavior was assessed. The second scenario involved a traffic light where a platoon was stopped. This platoon then had to accelerate smoothly when the traffic light went green, after which an approaching second platoon smoothly had to join the accelerating first platoon at the tail. The first GCDC demonstrated that cooperative driving in heterogeneous platoons was indeed possible, using very different control solutions. The GCDC of 2011 was won by Team ANNIEway [39], which considered all preceding vehicles in the platoon as a leader to derive multiple acceleration inputs for the host vehicle, after which the lowest acceleration input was selected for execution.

The second GCDC was held in 2016 [40], aiming to enable more complex cooperative manoeuvres. The first scenario that was tested, was merging of platoons on two highway lanes. This scenario includes manoeuvres such as vehicle following, gap making and changing lanes, which, when combined, led to a zipping manoeuvre. The second scenario, was the cooperative crossing of a T-intersection. In this scenario, vehicles coordinate their longitudinal motion to prevent collisions and avoid a full stop at a T-intersection to increase throughput. To enable these more complex manoeuvres, a hierarchical layer architecture and a more involved message set was used to coordinate the manoeuvres. The GCDC of 2016 was won by Team Halmstad university [41], which used the distance to the preceding vehicle, as well as a communicated acceleration of the preceding vehicle to determine the input to the host vehicle.

Participants of both GCDCs utilized different control strategies to achieve the same objective. Many of these differences can be traced back to how vehicles utilize information of downstream vehicles. These different controller topologies have also been studied extensively in literature. In [42], a Networked Control System modelling framework is used to study the influence of communication delays on the system. In certain scenarios where the cohesion of the platoon is important, such as for example truck platooning, it might be useful to consider bi-directional communication, was was done in [43]. It presents a method in which constraints on acceleration and velocity of vehicles are considered by vehicles in the downstream direction of the constrained vehicle. Other information topologies can also be considered. In [44] for example, the requirements on the information topology and local controllers to achieve string stability for a time-gap spacing policy are studied. Similarly, [45] demonstrates that for a rigid platoon configuration, bi-directional-leader topology is required for linear controllers to achieve a bounded stability margin, independent of platoon size, while [46] shows a similar result for asymmetrical controllers. Other examples where the platoon formation is rigid, are presented in [47]. In [47], the application is not necessarily platooning, but instead focusses on

scenarios in which the vehicles can have an arbitrary fixed formation, potentially spanning multiple lanes. The formation then cooperatively navigates around obstacles, while diverting from the fixed formation as little as possible by means of Model Predictive Control (MPC). In other work, platoons are considered in which not all vehicles are automated. In [48], a Connected Cruise Control (CCC) is presented, in which a heterogeneous set of vehicles is considered. Some of these vehicles can even be operated by humans. Various communication topologies are considered, and it is demonstrated that even if not all vehicles are automated, significant improvements can be achieved in terms of string stability when inserting a number of automated vehicles into traffic.

One of the important considerations in cooperative driving is the dependency on the wireless link. This wireless link will always to some extent be subject to communication delay and packet loss. In order to remain safe, graceful degradation of the system is required, in which the performance degrades in proportion to the failure [49]. Some solutions to this problem have been proposed, which include an estimator for the preceding vehicle [49], or the communication of multiple input steps by means of an MPC controller [50]. In [51], the string stability of such an MPC controller is assessed and verified experimentally.

In conclusion, the cooperative vehicle is capable of potentially lowering fuel consumption, increasing traffic flow and increasing the awareness of connected traffic participants by means of beyond-line-of-sight perception. This comes with the requirement that a wireless link is available for receiving and sharing information. The additional overhead required for cooperative driving depends on the application, but is typically larger than for autonomous vehicles. While the cooperative vehicle adds many advantages over autonomous vehicles, the application is often limited to some sort of vehicle following scenario.

### 1.1.3 i-CAVE (Integrated Cooperative Automated Vehicle)

Although the autonomous vehicle and cooperative vehicle are different classes of vehicles, with different control topologies, an overlap exists between the two. Both classes are equipped to actuate all vehicle controls and obtain information of the environment. A research program that focusses on this overlap is the i-CAVE program. i-CAVE (Integrated Cooperative Automated Vehicle) is a Dutch scientific program in which a Cooperative Dual Mode Automated Transport (C-DMAT) system is researched and designed [52]. This program is subdivided into six research projects being; Sensing and Mapping [53] [54] [55], Cooperative Vehicle Control [56] [57] [58], Dynamic Fleet Management [59], Communication [60], Human Factors [61] [62] [63] and Architecture Functional Safety [64]. The research from these six groups are integrated on a full-scale demonstrator platform [65]. The work in this thesis is part of the research programme i-CAVE Cooperative vehicle Control with project number

14893, which is (partly) financed by the Dutch Research Council (NWO).

## 1.2 Challenges in Trajectory Planning for Automated Cooperative Vehicles

Clearly, both the autonomous vehicle as well as the cooperative vehicle have appealing attributes. While the autonomous vehicle is highly versatile due to its trajectory planner, the cooperative vehicle is capable of improving overall traffic throughput by ensuring string-stable behavior. The cooperative vehicle in contrast is not suitable for a high variety of scenarios, due to the limitations of not having a trajectory planner. Ideally, the versatility of the autonomous vehicle would be combined with the interaction properties of the cooperative vehicle, in particular string stability. Although some research includes vehicle following in the trajectory planning framework (e.g., [18] [20]), string stability is not taken into account explicitly. Many trajectory planners specifically aimed towards highway driving are reviewed in [66]. However, none of them make use of cooperation, making string stable trajectory planning a clear research direction.

Also a vehicle that is capable of driving cooperatively, yet decide to overtake vehicles or break from the platoon when needed remains a challenge. An attempt at the development of such a system is presented in [67], which illustrated an MPC based platooning scenario, with lateral movement of the platoon to overtake other road users generated by Artificial Potential Fields (APF), in which the vehicles are allowed to switch between platooning and single vehicle cruising, although string stability was not explicitly analysed. Moreover, the MPC based trajectories would require a high communication bandwidth in practice.

The issue above is also closely related to challenges in communication. A traffic system in which cooperative vehicle are required cooperate with both autonomous vehicles, as well as other cooperative vehicles, poses challenges in terms of communication. As a result, the cooperative vehicle needs to operate with information in both the format of an autonomous vehicle as well as that of a cooperative vehicle. Additionally, the braking and forming of platoons in an ad-hoc fashion requires that the system works well without a priori knowledge of the information communication topology. Finally, to reduce the probability of dropped packets in the wireless link, data packets need to be small [68], hence solutions that utilize less communication bandwidth are preferred.

Although the notion of string-stability is well covered in literature, string-stability for trajectory planners has not been thoroughly covered. Specifically, when a trajectory planner plans a motion in continuous time in a receding horizon manner, both continuous-time dynamics (the planned trajectory) and discrete-time dynamics (updating of the planned trajectories at given time intervals) come into play. To the best of the authors knowledge, this

has not yet been covered in literature.

One drawback of trajectory planning over traditional feedback control is the computational burden. On the same hardware, a trajectory planner updates at significantly lower update rates. This might cause issues when driving at very short inter vehicle distances. In such cases, safe vehicle following can only be achieved if the update rate is sufficiently high to compensate for unexpected disturbances. Overcoming this drawback is important for safe vehicle following at short distances.

Finally, the switching between cooperative and autonomous trajectories should occur smoothly, be predictable, and be tunable by means of user parameters. When implemented correctly, a well implemented cost function in combination with a framework that is capable of planning multiple different types of trajectories, prevents the need for a behavioral layer to determine which type of motion planner to invoke.

A solution to the aforementioned challenges is to develop a method of cooperative trajectory planning. When string-stable vehicle following trajectories can be generated, they can be incorporated in the trajectory planner framework of the autonomous vehicle. This results in a versatile vehicle, that is capable of string-stable vehicle following. To obtain a traffic system in which autonomous vehicle and cooperative vehicles can cooperate, both driving modes should broadcast the same information. This allows for ad-hoc platoon formation, without the need of additional coordination. Ideally, only predecessor information is sufficient, as this simplifies implementation drastically. This allows the vehicle to simply join a platoon, or decide to break the platoon and perform autonomous manoeuvres, such as overtaking of slower vehicles. Additionally, this ad-hoc approach of forming and breaking of platoons, means that such a framework would scale well for longer platoons, as only information of the direct predecessor is required.

### 1.3 Research objectives and Contributions

Based on the aforementioned challenges, this thesis aims to contribute to the analysis and control of automated vehicles, that make use of explicit trajectory planning. In particular, automated vehicles are of interest that, that make use of a communication link with other vehicles, in order to share intended motions over a certain time horizon. Hence, the following objectives are defined:

- The design of a unified trajectory planning framework for automated and cooperative vehicles. A vehicle equipped with the aforementioned framework can operate both autonomously and in a cooperative fashion, using the same control topology, while broadcasting the same information in both modes.

- Ensure that cooperative trajectories are string-stable, such that the macroscopic traffic system benefits from the increased throughput.
- Experimental validation of the developed framework on full-scale vehicles.

The contributions of this thesis, based on the research objectives are as follows:

1. The development of a class of trajectory planning algorithms for cooperative vehicles. These cooperative vehicles make use of V2V communication, which is used to communicate their planned trajectories. The bandwidth of this communication is taken into account, by making use of parametrized trajectories.
2. The development of an autonomous trajectory planner, specifically designed in support of the cooperative trajectory planner. Additionally, the combination of both planners is considered and merged in a single framework. This combination allows the autonomous vehicle to maintain the versatility of the autonomous vehicle, while being capable of benefiting from V2V communication.
3. Next to the development of a cooperative trajectory planning method, this thesis also focusses on a string-stability analysis of the aforementioned algorithm. An analysis of  $\mathcal{L}_2$  string stability of a string of vehicles equipped with the cooperative trajectory planning algorithms is considered. Moreover, this study also considers the delay between time at which the preceding vehicle and the host vehicle plan their corresponding trajectories.
4. The development of a time-scaling mechanism to overcome the computational drawback of trajectory planning for vehicle following at short inter vehicle distances.
5. Two full-scale demonstrator platforms have been developed within the i-CAVE project to validate the theoretical results. The two Renault Twizy's that are used for this purpose have been equipped with sensing and actuating capabilities. These vehicles have been used to validate all aspects of the developed cooperative trajectory planning, including target tracking by means of a radar, the communication of the message used to broadcast the intention of the vehicle as well as the top-level supervisory layer.

## 1.4 Outline

The outline of this thesis is as follows. Chapter 2 describes the framework of the autonomous vehicle, the in-plane model of the vehicle, planning in the Frenet frame and a construction of a reference path and velocity. Additionally, it



provides the reader with basic information of splines, which is required in the following chapters. Chapter 3 describes the trajectory planning framework for the automated cooperative vehicle, specifically addressing the construction of longitudinal cooperative trajectories. A method for comfortable gap closing in vehicle following scenarios is also provided. Chapter 4 then described the construction of autonomous trajectories, as well as the construction of the lateral part for the cooperative trajectories. Additionally, it describes how both cooperative as well as autonomous trajectories can be integrated in a single framework and how the automated vehicle can switch between both modes. Chapter 5 then describes the  $\mathcal{L}_2$ -string stability analysis of the cooperative vehicle following trajectories. Additionally, necessary conditions for string stability are derived for a class of trajectory planning algorithms. Chapter 6 describes the experimental implementation of the trajectory planning algorithm on the full-scale vehicle setup. Chapter 7 will present a time-scaling method that is used to partially overcome the control bandwidth limitation of trajectory planning in cooperative vehicle following. Finally, Chapter 8 will provide conclusions on the presented work, and recommendations for future research directions.

# Chapter 2

## Background

This chapter presents some preliminary information that is used throughout the thesis. It starts with describing the motion planning framework that is extensively used in literature, and is partly adopted in this work as well. After that, the mathematical description of Bézier curves and B-splines is presented. Finally, the reference trajectory that is used in the trajectory planning framework is described. This reference trajectory consists of a Bézier spline and a reference velocity.

### 2.1 Motion Planning

To navigate through complex environments, automated vehicles make use of motion planning. The motion planning problem for automated vehicles can be summarized as follows: *Given a cost function, find the optimal, collision free and kinematically-feasible trajectory, that satisfy the vehicles constraints in real-time*, where the cost function can include many attributes of the trajectory depending on the application. Finding such a trajectory can be challenging in the presence of dynamic obstacles. This section discusses literature related various solutions to the motion planning problem for automated vehicles, after which details of the specific trajectory planning approach that is selected is presented.

#### 2.1.1 Literature Overview

Path and trajectory planning is a wide field of research, which can be subdivided in various layers with increasing complexity, ranging from route planning to kino-dynamic trajectory planning. The type of planning that is suitable for each planning algorithm is dependent on the application and the vehicle model that is used. In what follows, a general overview of literature is presented, without a specific vehicle model in mind.

Optimal route planning was pioneered by Dijkstra in 1959 [69]. Several years later, heuristics were added to speed up this algorithm in what became known as A\* [70] [71]. Dijkstra's algorithm and A\* solve a shortest-path problem, which is formulated in a connected graph. A graph consists of nodes and edges, in which the nodes can for example represent destinations, while the edges relate to routes connecting the destinations, each with a given distance or cost. Dijkstra's algorithm can actually be used for any planning problem, which can be represented by a graph. The nodes in the graph can also represent the possible (or discretized) configurations of the system, while the edges represent transitions between the configurations with corresponding costs. The optimal configuration trajectory is then given by the sequence of edges returned by the algorithm, which corresponds to the lowest cost transition.

These algorithms were also used in applications of robotics, where robots need to find a path and move through cluttered environments. However, in these cases, the environment is often only partially known, such that the graph needs to be updated online. To this extend D\* [72] was developed, which allowed for dynamic execution. Additionally, when using the A\* and D\* in a gridded representation of the environment, each node is only connected to its 8 direct neighbours, which might result in spatially suboptimal paths if the goal is in a certain place with respect to the origin. A solution to this issue is presented in [73], which includes different connectivity structures between nodes, which the authors have named basic Theta\* and Phi\*.

However, in robotics, simply finding a path is often not sufficient, as the path might not satisfy the kinodynamic constraints of the robot. While this might not be an issue for finding a route for the autonomous vehicle, it does become relevant when planning the local motion of the vehicle in for example a lane change. Although the A\* algorithm and similar algorithms might be used to solve kinodynamic planning problems by discretizing the state space of the robot, the computational complexity quickly blows up for higher dimensional state spaces. A method to overcome this problem is the Probabilistic Road Map (PRM) approach [74], which first samples the free configurations space (the space spanned by the generalized coordinates of the system) in a random manner to create nodes, after which these nodes (or configurations) are connected by means of the dynamics of the robot to assign cost to the edges. Finally, a graph search method such as A\* is used to find the optimal path. This method is complete in the sense that it finds a path if one exists when the number of samples increase. However, connecting two configurations that are close might be difficult in case of non-holonomic constraints.

A method that overcomes this issue, is the Rapidly expanding Random Trees (RRTs), which apply system inputs to drive the system closer towards randomly selected configurations [75] [76], as opposed to exact convergence to the configuration in the Probabilistic Road Map approach. These system inputs that are used to drive the system towards a sample can for example be provided by

an LQR controller, as was done in [77]. Due to the non-holonomic property of road vehicles, RRTs have been applied to trajectory planners for vehicles [78] [79] [80] [81]. A comprehensive comparison between RRT, PRM and A\* path planning for parking in automated vehicles can be found in [82]. A disadvantage of the RRTs is the probabilistic nature of the samples, and the fact that the found path is often non-optimal. The path obtained by RRT\* [83] [84] converges to the global optimal path, by reconnecting old nodes to new samples and potentially decrease the cost of reaching a certain node. This ensures that the returned cost converges to the global optimal cost the more the configuration space is sampled, but the number of samples that are required can be large. Other comparable methods are SST (Stable Sparse RRT) [85], in which a steering function is not needed.

These methods typically attempt to search for a global optimal solution. However, for on-road driving, typically the information regarding other road users is incomplete, or changes over time. Hence, the trajectory needs to be continuously updated on-line. Consequently, finding the global optimum to the planning problem might become to time-consuming to perform on-line. Instead, often *local* motion planning is used instead, in which the planners attempt to find the optimal trajectory towards some intermediate goal state. Additionally, reference information is often known a priori for on-road driving. Hence, the path finding for which these algorithms are used is not strictly necessary, as the general path is already known. This information can either be provided by Highly Automated Driving (HAD) maps, in the form of the centerline of the road, or it can be a refined version that is generated in real-time, such as in [14], [16]. This information can be used to achieve human-like driving on roads, by generating candidate trajectories that follow the general direction of the road. This method was already used in the first DARPA challenge by many of the teams. The information of the road can be used to sample intermediate goal states, as the vehicle is expected to drive along the road. This is also done extensively in literature. See for example [20] and [13] for graph search approached related to reference path. Additionally, [86], [18] and [87] present methods in which terminal states related to a reference path which are then connected to the initial state by means of smooth and kinematically-feasible paths.

### 2.1.2 Vehicle model

For the purpose of trajectory planning it is important to consider the dynamics and resulting constraints that these trajectories should satisfy. Note that for a planned trajectory to be feasible, it is required to start from the current vehicle state, as the vehicle cannot change states instantaneously. The vehicles considered in this thesis are typical road vehicles. Although extensive and complex vehicle models can be derived which can include pitch and roll of the

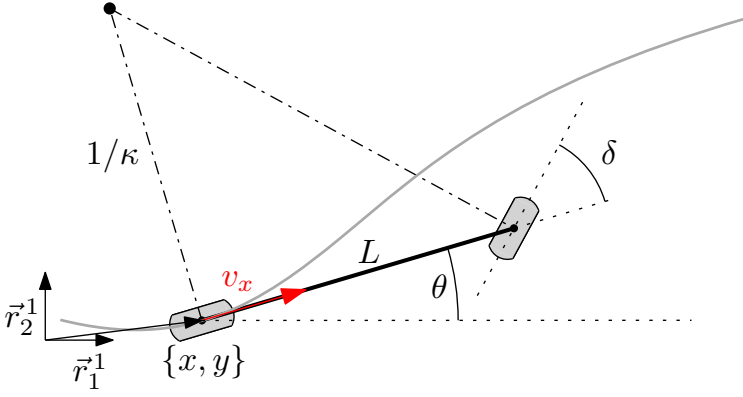


Figure 2.1: Kinematic bicycle model for in-plane dynamics

chassis, vertical wheel dynamics and (non-linear) tyre dynamics [88], most of these can be considered too detailed for the purpose of trajectory planning. Instead, a simplified model is used that is limited to the in-plane vehicle dynamics that are relevant for trajectory planning.

### Planar dynamics

A kinematic bicycle model is used as simplified model for the planar dynamics of the road vehicle. This bicycle model is illustrated in Figure 2.1, which shows the coordinates  $x$  and  $y$  expressed in the world fixed frame  $\vec{r}^1$ . The curvature  $\kappa$  of the path followed by the rear axle can be changed by means of steering angle  $\delta$ , and also depends on the wheelbase  $L$ . Finally the curvature changes the heading angle  $\theta$  whenever there is motion of the vehicle (e.g.,  $v_x \neq 0$ ). This can be captured by the following planar dynamics for vehicle  $i$

$$\kappa_i(t) = \frac{\tan \delta_i(t)}{L_i}, \quad (2.1a)$$

$$\dot{\theta}_i(t) = \kappa_i(t) v_{x,i}(t), \quad (2.1b)$$

$$\dot{x}_i(t) = v_{x,i}(t) \cos \theta_i(t), \quad (2.1c)$$

$$\dot{y}_i(t) = v_{x,i}(t) \sin \theta_i(t), \quad (2.1d)$$

in which  $L_i$  represents the wheel base,  $\delta_i(t)$  the steering angle and  $\kappa_i(t)$  the curvature of the spatial path followed by the center point of the rear axle. The steering angle  $\delta_i(t)$  and forward velocity  $v_{x,i}(t)$  can be considered the inputs to these planar dynamics.

### Internal vehicle dynamics

In addition to the in-plane dynamics, the vehicle is limited in terms of adjusting the steering wheel angle  $\delta$  and forward velocity  $v_x$ . These internal dynamics are modelled as following. The rate of change of the steering wheel cannot be changed instantaneously and is instead modelled by means of first order dynamics

$$\dot{\delta}_i(t) = \frac{1}{\eta_{\delta,i}} (u_{\delta,i}(t) - \delta_i(t)) \quad (2.2)$$

where  $\eta_{\delta,i}$  represents the time constant and  $u_{\delta,i}(t)$  the input.

In addition, the drive-line dynamics introduce continuity in the longitudinal acceleration, such that it cannot be changed instantaneously. The same model as used in [27], [50], [51], [3] can be adopted, which represents the experimental demonstrator platform that is used in later chapters. This model uses first-order dynamics for the drive-line

$$\dot{s}_{x,i}(t) = v_{x,i}(t), \quad (2.3a)$$

$$\dot{v}_{x,i}(t) = a_{x,i}(t), \quad (2.3b)$$

$$\dot{a}_{x,i}(t) = \frac{1}{\eta_{x,i}} (u_{x,i}(t) - a_{x,i}(t)), \quad (2.3c)$$

in which  $s_{x,i}(t)$  is the curvilinear distance travelled by the center of the rear axle,  $v_{x,i}(t)$  the longitudinal velocity,  $a_{x,i}(t)$  is the longitudinal acceleration,  $u_{x,i}(t)$  the commanded longitudinal acceleration, and  $\eta_{x,i}$  the time constant of the drive-line for vehicle  $i$ .

### Planning consequences

To ensure that planned trajectories are feasible, they should be initialized from the current vehicle states. As a result of the internal dynamics, these vehicle states include the current Cartesian position, longitudinal velocity, longitudinal acceleration, heading and steering angle.

To see that it is sufficient to require that the states are continuous up to the steering input and acceleration consider the following coordinate transformation. Define new augmented inputs to the system

$$\nu_{j,i}(t) := \frac{1}{\eta_{j,i}} (u_{j,i}(t) - a_{j,i}(t)), \quad j \in \{x, \delta\}, \quad (2.4)$$

such that the steering dynamics can be represented by a simple integrator

$$\dot{\delta}_i(t) = \nu_{\delta,i}(t). \quad (2.5)$$

Similarly a triple integrator system is obtained for the longitudinal dynamics.

$$\dot{s}_{x,i}(t) = v_{x,i}(t), \quad (2.6a)$$

$$\dot{v}_{x,i}(t) = a_{x,i}(t), \quad (2.6b)$$

$$\dot{a}_{x,i}(t) = \nu_{x,i}(t), \quad (2.6c)$$

in which  $s_{x,i}(t)$ , represents the curvilinear distance of the vehicle, and  $v_{x,i}(t)$  and  $a_{x,i}(t)$  define the velocity and acceleration in the  $\vec{r}_1^3$  direction. When the planned trajectories are initialized in these states, the trajectories in the original coordinates can also be reconstructed.

Additionally note that properties of the planned trajectories can be used to compensate for dynamics, since the input  $u_{x,i}(t)$  to the original system can be computed as  $u_{x,i}(t) = \eta_{x,i}\nu_{x,i}(t) + a_{x,i}(t)$ , where  $\nu_{x,i}$  can be substituted by the planned jerk. Similarly,  $u_{\delta,i}(t) = \eta_{\delta,i}\nu_{\delta,i}(t) + \delta_i(t)$ , where  $\nu_{\delta,i}(t)$  can be substituted by the planned steering rate. This shows a benefit of explicitly planning trajectories, rather than using a feedback strategy, as the drive-line and steering dynamics can be compensated.

Finally, as an additional benefit, this method also prevents jerky inputs due to replanning. Consider a trajectory planner that does not initialize its trajectories in the current acceleration and steering angle. In these cases, the reference steering angle and reference acceleration that the low-level controllers should track are discontinuous, which is uncomfortable for the passengers.

### 2.1.3 Motion Planning in Frenet Frame

The a priori information of the road can be used to define a new coordinate frame in which the trajectory planning problem is formulated. Note that only the reference path itself is not sufficient for on-road driving, as an obstacle might prevent the vehicle from continuing. Therefore, a method is required, that allows the vehicle to deviate from the reference path. A moving reference frame can be attached to the spatial reference path. This reference frame is referred to as a Frenet frame, and is illustrated in Figure 2.2. The figure illustrates a reference path  $\mathbf{X}_c(s) = [q_{c,x}(s), q_{c,y}(s)]$  in inertial reference frame  $\vec{r}^1$ , with components  $\vec{r}_1^1$ ,  $\vec{r}_2^1$ , where  $s$  is the curvilinear distance or covered arc length. This reference path can be regarded as being the centerline of the road, with sufficient continuity properties. A moving Frenet frame  $\vec{r}^2$ , with components  $\vec{r}_1^2$ ,  $\vec{r}_2^2$  is then defined at the projection of the vehicle onto the reference path, with  $\vec{r}_1^2$  pointing in the direction of the reference path.

The planning problem is now defined as determining the curvilinear distance  $s(t)$ , of the moving Frenet frame, and the lateral distance  $\ell(t)$ , of the vehicle to the Frenet frame, parametrized to time  $t$ . A trajectory then refers to the combination of longitudinal and lateral trajectories,  $\{s(t), \ell(t)\}$ . In particular, the longitudinal and lateral coordinates together with their time derivatives form

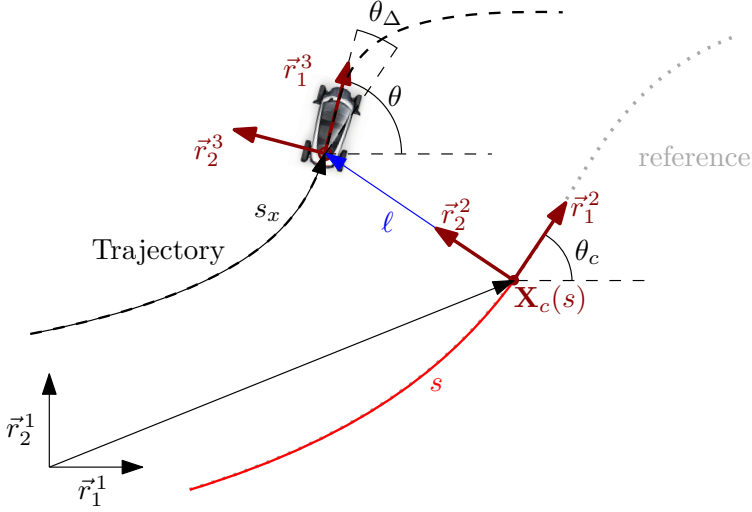


Figure 2.2: Reference path described by  $\mathbf{X}_c(s)$ , deviated path constructed via lateral offset  $\ell$  in the moving Frenet frame  $\vec{r}^2(s)$ , which is parametrized to curvilinear distance,  $s$ .

state trajectories  $\mathbf{S}(t) := [s(t), \dot{s}(t), \ddot{s}(t)]^\top$  and  $\mathbf{\Lambda}(t) := [\ell(t), \dot{\ell}(t), \ddot{\ell}(t)]^\top$ . The objective of the trajectory planner is to find feasible, collision free and comfortable trajectories,  $\mathbf{S}(t), \mathbf{\Lambda}(t)$ , starting from initial state  $\mathbf{S}(t_0), \mathbf{\Lambda}(t_0)$ , over a time horizon  $t \in [t_0, t_0 + T]$ , while considering dynamic obstacles.

Another possibility is to sample the intermediate goal states by means of this Frenet frame, yet still plan the trajectories in the original Cartesian coordinate frame. A comparison between these approaches is illustrated in Figure 2.3. Both planning approaches in the example make use of quintic polynomials, as they can be used to minimize jerk [18] [89]. These polynomials are initialized in the current state  $\mathbf{S}(t_0), \mathbf{\Lambda}(t_0)$ , which results in continuity in position, velocity and acceleration (and curvature and direction). The Frenet frame is used to select 7 different laterally off-setted intermediate goal states that are aligned with the reference path. These goal states are identical in both examples. Clearly, by planning completely in the Frenet frame, the constructed trajectories follow the curves of the road, whereas the planning in Cartesian coordinates cuts corners. Although no constraints have been applied in the example, application of spatial constraints is easier in the Frenet frame.



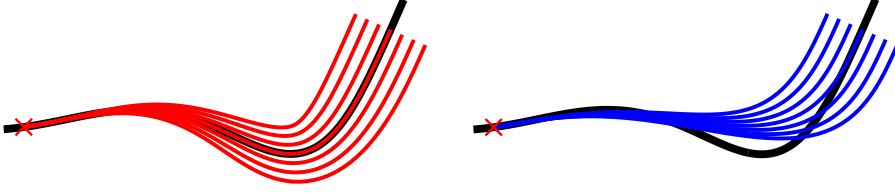


Figure 2.3: Comparison of planning in the Frenet frame vs. planning in the Cartesian frame. Left: Planning in Frenet frame by means of quintic polynomials in  $s(t)$  and  $\ell(t)$  for 7 intermediate goal states that are laterally off setted with respect to the reference path. Right: Planning in Cartesian frame by means of quintic polynomials in  $x_{h,x}(t)$  and  $x_{h,y}(t)$ .

### Mapping of Coordinates

The coordinates  $[s(t), \dot{s}(t), \ddot{s}(t), \ell(t), \dot{\ell}(t), \ddot{\ell}(t)]$  in the Frenet frame  $\vec{r}^2(s)$  based on reference path  $\mathbf{X}_c(s)$ , need to be mapped to coordinates  $[q_x(t), q_y(t), \theta(t), \kappa(t), v_x(t), a_x(t)]$ , where  $q_x(t)$ ,  $q_y(t)$ ,  $\theta(t)$  are position and orientation of the vehicle frame  $\vec{r}^3$  with respect the global Cartesian coordinate frame  $\vec{r}^1$ ,  $\kappa := \frac{d}{ds}\theta$  is curvature of the vehicles trajectory,  $s_x(t)$ ,  $v_x(t)$  and  $a_x(t)$  are the arc length, velocity and acceleration of the vehicle in the  $\vec{r}_1^3$  direction and velocity in  $\vec{r}_2^3$  is zero due to the kinematic description of the motion. These can be computed by the following relations [18]. Position coordinates are found by means of superposition

$$\begin{bmatrix} q_x \\ q_y \end{bmatrix} = \mathbf{X}_c(s) + \begin{bmatrix} \cos \theta_c(s) & -\sin \theta_c(s) \\ \sin \theta_c(s) & \cos \theta_c(s) \end{bmatrix} \begin{bmatrix} 0 \\ \ell \end{bmatrix}. \quad (2.7)$$

Given the curvature  $\kappa_c := \frac{d}{ds}\theta_c$ , and heading angle  $\theta_c$  of the reference line, the following relation can be identified for  $\dot{s}$  and  $\dot{\ell}$

$$\dot{s}(1 - \kappa_c \ell) = v_x \cos \theta_\Delta, \quad (2.8)$$

$$\dot{\ell} = v_x \sin \theta_\Delta, \quad (2.9)$$

$$\theta_\Delta = \theta - \theta_c.$$

where  $\theta_\Delta$  represents the heading difference between the vehicle frame and Frenet frame, and the  $-\kappa_c \ell$  term accounts for the rotation of the Frenet frame when it moves along the reference line. Using these relations, the vehicle velocity  $v_x$  can be written as

$$v_x = \sqrt{(1 - \kappa_c \ell)^2 \dot{s}^2 + \dot{\ell}^2}. \quad (2.10)$$

The derivative of  $\ell$  with respect to  $s$  is found via

$$\ell' = \frac{d}{ds}\ell = \frac{dt}{ds} \frac{d}{dt}\ell = \frac{\dot{\ell}}{\dot{s}} = (1 - \kappa_c \ell) \tan \theta_\Delta. \quad (2.11)$$

To find the remaining coordinates, a derivative operator with respect to  $s$  is defined, that makes use of the covered arc length  $s_x$  of the trajectory of the vehicle frame. By substituting (2.8) the following can be written for operator of the derivative with respect to covered arc length of the reference line

$$\frac{d}{ds} = \frac{ds_x}{ds} \frac{d}{ds_x} = \frac{ds_x}{dt} \frac{dt}{ds} \frac{d}{ds_x} = \frac{v_x}{\dot{s}} \frac{d}{ds_x} = \frac{1 - \kappa_c \ell}{\cos(\theta_\Delta)} \frac{d}{ds_x}. \quad (2.12)$$

With the definitions of curvatures  $\kappa_c$  and  $\kappa$ , the derivative of heading difference is written as

$$\frac{d}{ds} \theta_\Delta = \frac{1 - \kappa_c \ell}{\cos(\theta_\Delta)} \frac{d}{ds_x} \theta - \frac{d}{ds} \theta_c = \frac{1 - \kappa_c \ell}{\cos(\theta_\Delta)} \kappa - \kappa_c.$$

Then for the second derivative of  $\ell$  with respect to  $s$  is found by differentiating (2.11) again

$$\begin{aligned} \ell'' &= \frac{d^2}{ds^2} \ell = \frac{d}{ds} ((1 - \kappa_c \ell) \tan(\theta_\Delta)), \\ &= (1 - \kappa_c \ell) \frac{d}{ds} \tan(\theta_\Delta) - \tan(\theta_\Delta) \frac{d(\kappa_c \ell)}{ds}, \\ &= \frac{(1 - \kappa_c \ell)}{\cos^2(\theta_\Delta)} \left( \frac{1 - \kappa_c \ell}{\cos(\theta_\Delta)} \kappa - \kappa_c \right) - \tan(\theta_\Delta) (\kappa'_c \ell + \kappa_c \ell'), \end{aligned} \quad (2.13)$$

where

$$\frac{d}{ds} \tan(\theta_\Delta) = \frac{1}{\cos^2(\theta_\Delta)} \left( \frac{1 - \kappa_c \ell}{\cos(\theta_\Delta)} \kappa - \kappa_c \right),$$

was used. By assuming the vehicle travels in the direction of the road, the heading difference can be said to satisfy  $|\theta_\Delta| < \frac{\pi}{2}$ . Consequently, by combining (2.8) with  $v_x \geq 0$  (vehicle is not moving backwards in vehicle frame) and  $\dot{s} \geq 0$  (vehicle is not moving backwards along the reference line) it can be seen that  $(1 - \kappa_c \ell) > 0$ . Hence, (2.11) and (2.13) can be solved to find  $\theta$  and  $\kappa$  of the trajectory. Finally acceleration in the vehicle frame  $a_x$  is found by first determining the time derivative of (2.8)

$$\begin{aligned} v_x &= \dot{s} \frac{1 - \kappa_c \ell}{\cos \theta_\Delta}, \\ a_x &= \dot{v}_x = \ddot{s} \frac{1 - \kappa_c \ell}{\cos \theta_\Delta} + \dot{s} \frac{ds}{dt} \frac{d}{ds} \left( \frac{1 - \kappa_c \ell}{\cos \theta_\Delta} \right) = \ddot{s} \frac{1 - \kappa_c \ell}{\cos \theta_\Delta} \\ &\quad + \frac{\dot{s}^2}{\cos \theta_\Delta} \left( (1 - \kappa_c \ell) \tan \theta_\Delta \left( \frac{1 - \kappa_c \ell}{\cos \theta_\Delta} \kappa - \kappa_c \right) - (\kappa'_c \ell + \kappa_c \ell') \right), \end{aligned} \quad (2.14)$$

where it is used that

$$\frac{d}{ds} \frac{1}{\cos \theta_\Delta} = \frac{\tan \theta_\Delta}{\cos \theta_\Delta} \left( \frac{1 - \kappa_c \ell}{\cos \theta_\Delta} \kappa - \kappa_c \right).$$

Finally to relate the time derivatives of  $\ell$  to the spatial derivatives of  $\ell$ , the following relations can be found.

$$\dot{\ell} = \frac{ds}{dt} \frac{d}{ds} \ell = \dot{s} \ell', \quad (2.15)$$

$$\ddot{\ell} = \frac{d}{dt} \dot{\ell} = \ddot{s} \ell' + \dot{s}^2 \ell''. \quad (2.16)$$

### Implications of continuity

Consider again the vehicle model presented in Section 2.1.2, which requires continuity in curvature  $\kappa(t)$  and acceleration  $a_x$ . This continuity requirement in turn imposes requirements on the continuity of the reference path  $\mathbf{X}_c(s)$ , and the planned trajectory  $\ell(t)$  and  $s(t)$  in the Frenet frame, by means of the mapping  $[\mathbf{S}_i(t), \Lambda_i(t), \mathbf{X}_c(s)] \rightarrow [s_{x,i}(t), v_{x,i}(t), a_{x,i}(t), \delta_i(t)]$  described above, in which  $s(t)$  can be used to obtain  $X_c(s(t))$ . Note that the mapping to  $\delta_i(t)$  was not explicitly given, but can be derived by rewriting (2.13) to find expression for  $\kappa(t)$

$$\kappa(t) = \left( (\ell'' + \tan(\theta_\Delta) (\kappa'_c \ell + \kappa_c \ell')) \frac{\cos^2(\theta_\Delta)}{(1 - \kappa_c \ell)} + \kappa_c \right) \frac{\cos(\theta_\Delta)}{1 - \kappa_c \ell}, \quad (2.17)$$

and combining it with (2.1a). Consequently the requirement of continuity in  $\kappa(t)$  imposes the requirement that the reference path  $\mathbf{X}_c(s)$  is thrice continuously differentiable and  $\ell(s)$  is twice continuously differentiable.

Similarly by considering (2.14), it can be seen that in order for  $a_x(t)$  in the vehicle frame to be continuous, again  $\mathbf{X}_c(s)$  should be three times and  $\ell(s)$  is twice continuously differentiable. However, in order to compute the input to the system (2.6), the jerk  $\dot{a}_x(t)$  should also be continuous, hence  $\mathbf{X}_c(s)$  should be four times and  $\ell(s)$  is three times continuously differentiable. In practice however, the contributions to motion in the vehicle frame  $[s_{x,i}(t), v_{x,i}(t), a_{x,i}(t)]$ , of the lateral trajectories  $\Lambda(t)$  are much smaller than that of  $\mathbf{S}(t)$ .

### 2.1.4 Trajectory Generation

To navigate a high variety of scenarios, the trajectory planner should be capable of diverting from the reference trajectory. This is useful when a lane is blocked or other road vehicles obstruct the reference path. Moreover, the planner should be capable of doing this in real-time, as the automated vehicle cannot come to a standstill and consider its options. An approach that is often presented in literature, is to generate a wide variety of candidate trajectories, and selecting

the best one for the current scenario for execution. After generating the trajectories, all trajectories are checked for feasibility and a collision check is performed. This approach makes it very easy to take into account dynamic obstacles. During trajectory generation, obstacles are not considered. The collision check with the trajectory is easily performed afterwards. In [18], a set of terminal states and arrival times are selected, to connect the current vehicle states to these goal states by means of quintic polynomials. Because no inequality constraints are considered during the trajectory construction phase, a closed form expression can be found to generate these trajectories. As a result, no iterative optimization is required, which is beneficial for real-time implementation, since the time to compute a trajectory will always have a hard upper bound.

### Polynomial Trajectories

Following the notation used in [18], a trajectory  $\xi(t) = [\xi(t), \dot{\xi}(t), \ddot{\xi}(t)]^\top$  related to a generalized coordinate  $\xi(t)$  that connects initial state  $\xi_0 = \xi(0)$  to terminal state  $\xi_\tau = \xi(\tau)$ , can be constructed by means of a quintic polynomial as

$$\xi(t) = \mathbf{M}_1(t - t_0)\mathbf{c}_{012} + \mathbf{M}_2(t - t_0)\mathbf{c}_{345}, \quad t_0 \leq t \leq t_0 + \tau, \quad (2.18)$$

$$\mathbf{M}_1(t) = \begin{bmatrix} 1 & t & t^2 \\ 0 & 1 & 2t \\ 0 & 0 & 2 \end{bmatrix}, \quad \mathbf{M}_2(t) = \begin{bmatrix} t^3 & t^4 & t^5 \\ 3t^2 & 4t^3 & 5t^4 \\ 6t & 12t^2 & 20t^3 \end{bmatrix}. \quad (2.19)$$

With  $\mathbf{M}_2(0) = \mathbf{0}_3$ , the first coefficient can be determined by the initial conditions and the final coefficient by the terminal condition by means of

$$\begin{aligned} \mathbf{c}_{012} &= [\mathbf{c}_0 \quad \mathbf{c}_1 \quad \mathbf{c}_2]^\top = \mathbf{M}_1(0)^{-1}\xi_0, \\ \mathbf{c}_{345} &= [\mathbf{c}_3 \quad \mathbf{c}_4 \quad \mathbf{c}_5]^\top = \mathbf{M}_2(\tau)^{-1}(\xi_\tau - \mathbf{M}_1(\tau)\mathbf{c}_{012}), \quad \tau > 0 \end{aligned}$$

where  $\mathbf{c}_{012}$  and  $\mathbf{c}_{345}$  represent the polynomial coefficients. For a quartic polynomial, with  $\mathbf{c}_5 = 0$ , the polynomial  $\xi(t)$  and  $\mathbf{c}_{34}$  are computed as

$$\begin{aligned} \xi(t) &= \mathbf{M}_1(t - t_0)\mathbf{c}_{012} + \mathbf{M}_2(t - t_0) \begin{bmatrix} I_2 \\ 0 \end{bmatrix} \mathbf{c}_{34}, \quad t_0 \leq t \leq t_0 + \tau, \quad (2.20) \\ \mathbf{c}_{34} &= \begin{bmatrix} \mathbf{c}_3 \\ \mathbf{c}_4 \end{bmatrix} = \left( \begin{bmatrix} 0 & I_2 \end{bmatrix} \mathbf{M}_2(\tau) \begin{bmatrix} 0 \\ I_2 \end{bmatrix} \right)^{-1} \left( \begin{bmatrix} 0 & I_2 \end{bmatrix} \xi_\tau - \mathbf{M}_1(\tau)\mathbf{c}_{012} \right), \end{aligned}$$

where only the terminal velocity and acceleration are constrained, and the position state is arbitrary. This is useful for example in cases of highway driving, where the terminal position is not important, but instead a certain velocity is more important. These polynomial trajectories can then be used to construct trajectories, that connect the current vehicle state  $\mathbf{S}(t_0)$ ,  $\mathbf{\Lambda}(t_0)$ , to a terminal state  $\mathbf{S}(\tau)$ ,  $\mathbf{\Lambda}(\tau)$ .

A diverse set of terminal states is created by means of set of  $N_{\dot{s}} + N_s$  longitudinal, and  $N_\ell$  lateral terminal states. For the longitudinal trajectories, quartic polynomials are used to connect to goal states of a given  $\dot{s}(\tau) = \dot{s}_j$ ,  $j \in [1, \dots, N_{\dot{s}}]$  and  $\ddot{s}(\tau) = 0$ , which are useful for highway driving with a certain velocity. Alternatively, quintic polynomials are used to connect to goal states with a specified goal position  $s(\tau) = s_j$ ,  $j \in [1, \dots, N_s]$ ,  $\dot{s}(\tau) = 0$  and  $\ddot{s}(\tau) = 0$ , which are useful for a precision stop when stopping behind another vehicle, at a traffic light or other scenarios that require a full stop.

The lateral trajectories are constructed by means of quintic polynomials in such a way that they follow the direction of travel of the road lane. By selecting goal states for a given lateral position  $\ell$ , and  $\dot{\ell} = 0$ ,  $\ddot{\ell} = 0$ , the trajectories always continue to follow the direction of the road after a lateral swerve. Note that due to the non-holonomic constraints of road vehicles, the state  $\ell(t)$  cannot vary when  $s(t)$  does not. In [17], a solution is proposed in which for low velocities (e.g.,  $\dot{s}(t)$  below a given threshold), quintic polynomials parametrised in curvilinear distance  $s$  (e.g.,  $\ell(s(t))$ ) instead of time  $t$  (e.g.,  $\ell(t)$ ) are used for the lateral trajectories. This solution prevents that a large portion of the trajectories is invalidated due to the high curvatures, resulting from large changes in  $\ell$  and small changes in  $s$  (driving slow). Note that in these cases the initial state  $\ell(s(t_0))$ ,  $\ell'(s(t_0))$ ,  $\ell''(s(t_0))$  is connected to terminal states  $\ell(s(\tau)) = \ell_l$ ,  $l \in \{1 \dots, N_\ell\}$ ,  $\ell'(s(\tau)) = 0$ ,  $\ell''(s(\tau)) = 0$ .

Finally, by including  $N_\tau$  arrival times  $\tau_k$ ,  $k \in \{1, \dots, N_\tau\}$ , both fast and highly energetic trajectories as well as slower more gradual changing trajectories are included. The faster trajectories can potentially be used to evade nearby obstacles or quickly reduce velocity when needed, while the slower trajectories with lower acceleration and jerk are more comfortable. Each of the  $(N_{\dot{s}} + N_s) \times N_\ell$  combinations of trajectories is generated for various arrival times  $\tau_k$ . Trajectories where  $\tau_k > T$  are truncated to  $t_0 \leq t \leq t_0 + T$ , and trajectories where  $\tau_k < T$  are extended by means of extrapolation with constant velocity such that all trajectories are defined on the same time horizon  $t_0 \leq t \leq t_0 + T$ .

A set of these trajectories is shown in Figure 2.4. It can be seen that all the trajectories indeed follow the general direction of the road. These trajectories are generated by means of the following set of arrival times and terminal states

$$\dot{s}_j = \frac{j}{N_{\dot{s}}} v_c \quad j \in \{1, \dots, N_{\dot{s}}\}, \quad (2.21)$$

$$\ell_l = \left( \frac{2(l-1)}{N_\ell-1} - 1 \right) \ell_{\max} \quad l \in \{1, \dots, N_\ell\}, \quad (2.22)$$

$$\tau_k = 2T \left( \frac{2}{3} \right)^{(k-1)} \quad k \in \{1, \dots, N_\tau\}, \quad (2.23)$$

where  $\ell_l$  takes on linearly spaced values in  $[-\ell_{\max}, \ell_{\max}]$ , with  $\ell_{\max}$  the maximum lateral offset based on the width of the road. For the velocity,  $\dot{s}_j$  takes

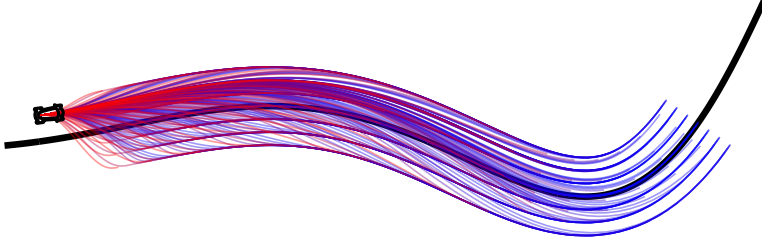


Figure 2.4: Visualization of a set of generated trajectories with a set of terminal conditions for  $N_\tau = 6$ ,  $N_s = 7$ ,  $N_s = 0$ ,  $N_\ell = 7$ , along a spatial reference path.

on linearly spaced values related to the reference velocity  $v_c$ , of which the construction is discussed at the end of this chapter. In the example, no specific positioning trajectories are used ( $N_s = 0$ ) as no stopping is required in this scenario.

### Feasibility and Collision Check

The generated trajectories can be subsequently checked for feasibility and collisions. Feasibility of the trajectory in terms of minimum turn radius, maximum acceleration, maximum velocity and maximum equivalent power  $v(t)a(t)$  can easily be checked by means of the computed signals  $\kappa(t), v(t), a(t)$  by solving (2.13) for  $\kappa(t)$ , computing  $v(t)$  via (2.10) and  $a(t)$  via (2.14).

Multiple options exist for collision checks. In [90], a method is presented in which the dimensions of the host vehicle and obstacles are covered in circles, which allows for an easy and computationally inexpensive collision check. However, it does introduce some conservatism, as the circles always extend past the polygon outline of the vehicle. This can be seen in Figure 2.5b, which illustrates the circles that are used to cover the polygon of a vehicle. Clearly, the area covered by the circles is larger than that of the vehicle, resulting in conservatism in the collision check. Another option is the separating axis theorem presented in [91]. This theorem states that two convex polygons do not intersect if and only if there exist an axis that separates them. Additionally, it states that the vertices of each polygon can be used as a candidate for the separating axis, such that a finite number of axes can be used to perform the collision check. This ensures that no conservatism is introduced by the collision check, and that the vehicle can potentially navigate very narrow passages. An illustration of the separating axis theorem is shown in Figure 2.5a. It can be seen that  $\vec{r}_1^a$  and  $\vec{r}_2^a$  are not a separating axis between bounding box  $a$  and  $b$ , as projecting the geometry of both  $a$  and  $b$  on these vectors results in overlap. In contrast, projecting the geometry on vector  $\vec{r}_1^b$  does not result in overlap, such that it is a separating axis, and the polygons can be concluded to not overlap.

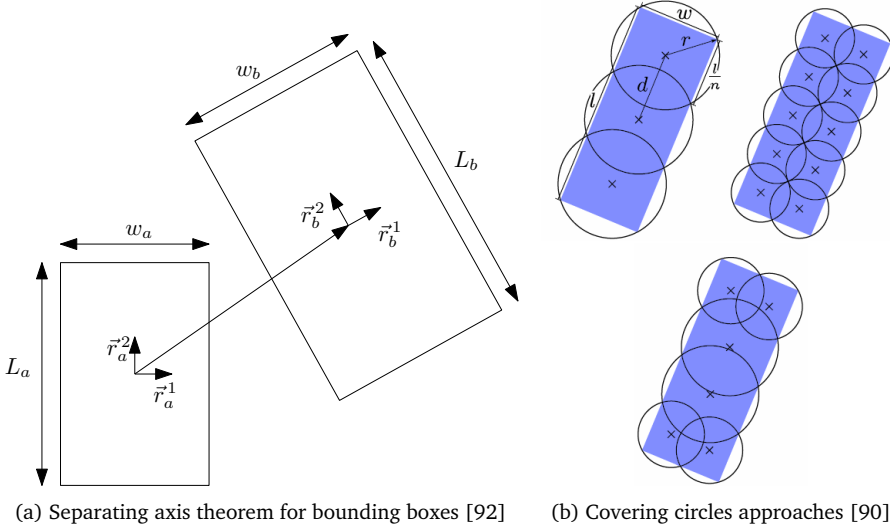


Figure 2.5: Different approaches for Collision checks

### Cost Function

The optimal trajectory with respect to a certain cost function can then be selected from the remaining feasible and collision free trajectories. A suitable cost function should be chosen to represent desirable vehicle behavior. Desirable behavior naturally includes safe driving (e.g., keeping sufficient distance from other road users and obstacles), progression along the road, and lane centering. Additionally, the cost function in combination with the candidate trajectories should result in temporal consistent behavior, such that the trajectories selected for execution in consecutive planning cycles coincide over the part of the time horizon on which both are defined.

In [17] [18], a simple cost function is used, with cost terms for arrival time, terminal velocity or position, and terminal lateral position. The trajectories that plan towards a given terminal velocity with quartic polynomials, as well as trajectories that plan towards a given terminal position with quintic polynomials have respective cost functions

$$J_{\dot{s}} = Q_{\tau}\tau + \frac{1}{2}Q_{\dot{s}}(\dot{s}(\tau) - \dot{s}_{\text{ref}}(\tau))^2 + \frac{1}{2}Q_{\ell}\ell(\tau)^2, \quad (2.24)$$

$$J_s = Q_{\tau}\tau + \frac{1}{2}Q_s(s(\tau) - s_{\text{ref}}(\tau))^2 + \frac{1}{2}Q_{\ell}\ell(\tau)^2. \quad (2.25)$$

Note that these cost functions indeed achieve part of the desired behavior. Progression along the road with a given desired reference velocity is prioritised

over speeding beyond the legal limit and driving slower than the reference velocity. Deviation from the lane center ( $\ell = 0$ ) is also penalized, but only with respect to a single lane, which might not be the desired behavior in case of highway driving with multiple lanes. Temporal consistency can also be achieved, but requires some additional bookkeeping. Note that the quintic and quartic polynomials are actually optimal in terms of minimal jerk for a given terminal state, as shown in [18]. Due to the principle of optimality, temporal consistency is then achieved if the same terminal state is selected. This requires that the  $\tau$ ,  $\dot{s}(\tau)/s(\tau)$ ,  $\ell$  corresponding to the selected trajectory, should be included in the set of terminal states in the consecutive plan step. However, to ensure that these trajectories are also selected in the next plan step, trajectories with lower arrival times  $\tau$  should be selected. Consequently, higher energetic trajectories are prioritized over more gradual and comfortable trajectories, which is usually undesirable from a comfort perspective. Also note that maintaining a safe distance is not necessarily included in this simple cost function. Hence, a 'near-miss' trajectory might pass the collision check, while it is not suitable for safe execution.

Alternatively, more complex and intricate cost functions can be used to select the optimal trajectories. In [93], a manually designed non-linear vertices-based cost function is used to evaluate the trajectories. In contrast to the cost function of [18], here distance towards the other road users is included and made dependent on the velocity. This helps to pass other road users at safer distances, and maintain safer distances to the preceding vehicle that depend on the velocity of both host and preceding vehicle. Additionally, it penalizes acceleration to improve passenger comfort.

A different cost function is presented in [94], [95], which uses Artificial Potential Fields (APF) as cost functions, to promote lane centering and maintaining a safe distance to other vehicles. This artificial potential field approach provides an elegant solution to perform lane centering on multiple lanes. A local minimum of cost in the lateral direction is located in the center of each lane. Some potential around the road markers that separate the lanes, ensures that a vehicle prioritizes changing lanes over driving on top of a road marker. Additionally, higher potentials on the shoulders of the road prevent that the vehicle drives off the road completely. Similarly, a Gaussian potential field is included around other road vehicles, such that trajectories that maintain a larger distance while overtaking are preferred over trajectories that pass other vehicles at small distances (e.g., a few centimetres). The disadvantage of these cost functions, is that temporal consistency is by no means ensured.



## 2.2 Splines

Splines are piecewise polynomials used for the description of curves [96]. These curves are often parametric and map a parameter to  $\mathbb{R}^n$ . Splines have been widely used for many applications, such as Computer Aided Design, Approximation and Interpolation. Splines have also been used in trajectory planning, both for spatial path description in  $\mathbb{R}^2$  [5] [6] [97] [98] [99], as well as state trajectories [100] [101] [102] [103] [104].

### 2.2.1 Bézier curves

Bézier curves were pioneered by Pierre Bézier [105] and Paul de Casteljau [106] who used them for the design of bodywork for Renault and Citroën respectively. A Bézier curve of degree  $p$  maps parameter  $u \in [0, 1]$  to  $\mathbf{C}(u) \in \mathbb{R}^n$ , and is constructed by means of control points  $\mathbf{P}_j \in \mathbb{R}^n$ ,  $j \in [0, \dots, p]$  [107]

$$\begin{aligned} \mathbf{C}(u) &= \sum_{j=0}^p b_{j,p}(u) \mathbf{P}_j = \mathbf{b}_p(u) \mathcal{P}, \quad 0 \leq u \leq 1, \\ \mathbf{b}_p(u) &= [b_{0,p}(u) \quad b_{1,p}(u) \quad \dots \quad b_{p,p}(u)], \\ \mathcal{P} &= [\mathbf{P}_0 \quad \mathbf{P}_1 \quad \dots \quad \mathbf{P}_p]^\top, \end{aligned} \quad (2.26)$$

where  $b_{j,p}(u)$  is the *Bernstein basis polynomial*

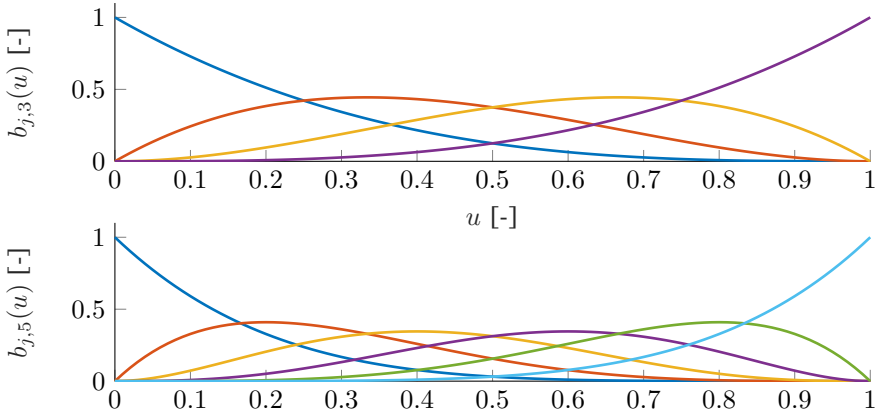
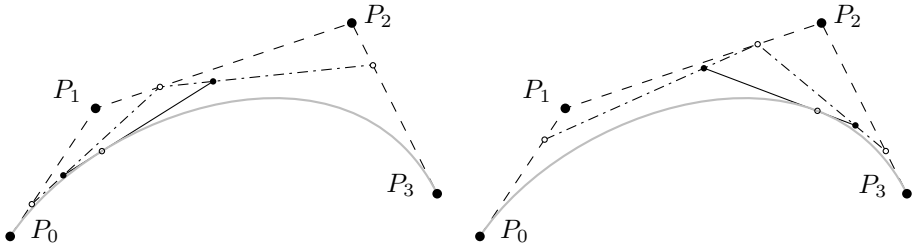
$$b_{j,p}(u) = \binom{p}{j} (1-u)^{(p-j)} u^j = \frac{p!}{j!(p-j)!} u^j (1-u)^{p-j}, \quad j \in \{0, \dots, p\}. \quad (2.27)$$

For example, a Bézier curve of degree 3 requires control points  $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$  to construct  $\mathbf{C}(u) = (1-u)^3 \mathbf{P}_0 + 3u(1-u)^2 \mathbf{P}_1 + 3u^2(1-u) \mathbf{P}_2 + u^3 \mathbf{P}_3$ . Basis functions  $b_{j,3}(u)$ ,  $j \in \{0, \dots, 3\}$  and  $b_{j,5}(u)$ ,  $j \in \{0, \dots, 5\}$  are illustrated in Figure 2.6.

An example of a Bézier curve of degree  $p = 3$  is shown in Figure 2.7, which illustrates the construction of the curve geometrically by means of the Casteljau Algorithm. For any  $u \in [0, 1]$ , a point at  $u$  times the length of the segments is constructed on each of the line segments. These points are then again connected to the subsequent point on the next line segment, after which a point is added again at  $u$  times the length of the new line segment. This process is repeated until only one point remains, which coincides with  $\mathbf{C}(u)$ . The line segments connecting the control points are referred to as the control polygon and determines the shape of the spline.

Important properties of Bézier curves are as follows [107] :

1. The curve coincides with the first and with the final control point, i.e.,  $\mathbf{C}(0) = \mathbf{P}_0$  and  $\mathbf{C}(1) = \mathbf{P}_n$ .

Figure 2.6: Basis functions,  $b_{j,p}$  for  $p = 3$  and  $p = 5$ .Figure 2.7: Graphical representation of the Casteljau algorithm for the construction of Bézier curves by means of their control points, for  $u = \frac{1}{4}$  (left) and  $u = \frac{3}{4}$  (right)

2. The direction vector of the tangents to the curves at beginning and end of the curve coincide with the directions of  $\mathbf{P}_1 - \mathbf{P}_0$  and  $\mathbf{P}_n - \mathbf{P}_{n-1}$ .
3. The curves lie in the convex hull of the control points.
4. The curves are invariant under rotations, translations and scaling. Rotations, translations and scaling of the control points result in a similar rotation, translation and scaling of the curve.

The basis functions of these Bézier curves, in the form of Bernstein polynomials, have the following properties for  $0 \leq u \leq 1$ :

1.  $b_{j,p}(u) \geq 0$ .
2.  $\sum_{j=0}^p b_{j,p}(u) = 1$ .
3.  $b_{j,p}(u)$  has a maximum at  $u = j/p$ .
4. basis functions can be recursively defined  
 $b_{j,p}(u) = (1-u)b_{j,p-1}(u) + ub_{j-1,p-1}(u)$ , where  $b_{j,p-1}(u) \equiv 0$  if  $j < 0$  or  $j > p$ .

### Bézier Derivatives

The derivative of a Bézier curve is again a Bézier curve. This can be recognised by noting that  $\frac{db_{j,p}(u)}{du} = p(b_{j-1,p-1}(u) - b_{j,p-1}(u))$ , with  $b_{-1,p-1}(u) \equiv b_{p,p-1}(u) \equiv 0$ . Using this property the derivative of a Bézier curve

$$\begin{aligned} \mathbf{C}^{(1)}(u) &\equiv \frac{d}{du} \mathbf{C}(u) = \sum_{j=0}^p p(b_{j-1,p-1}(u) - b_{j,p-1}(u)) \mathbf{P}_j \\ &= \sum_{j=0}^{p-1} b_{j,p-1}(u) p(\mathbf{P}_{j+1} - \mathbf{P}_j), \end{aligned}$$

such that the derivative of a Bézier curve is again a Bézier curve, with control points  $p(\mathbf{P}_{j+1} - \mathbf{P}_j)$ ,  $j \in \{0, \dots, p-1\}$ . The derivative of a curve is referred to as its hodograph. The hodograph corresponding with the  $d$ -th derivative of the curve is denoted as

$$\mathbf{C}^{(d)}(u) = \sum_{j=0}^{p-d} b_{j,p-d}(u) \mathbf{P}_j^{(d)} = \mathbf{b}_{j,p-d}(u) \mathcal{P}^{(d)}, \quad (2.28)$$

with  $\mathbf{P}_j^{(d)}$ ,  $j \in \{0, \dots, p-d\}$  the control points of this derivative curve. To find these control points of the hodograph, the following relation can be used

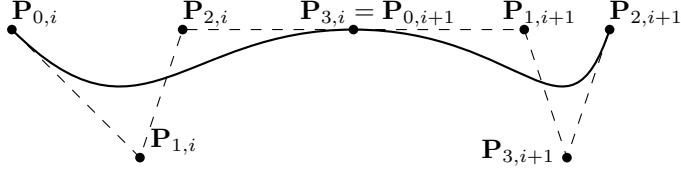
$$\mathcal{P}^{(d)} = \underbrace{\Delta_{d \ d-1} \Delta_{d-1 \ d-2} \Delta_{21} \Delta_{10}}_{\Delta_{d0}} \mathcal{P}, \quad (2.29)$$

$$\Delta_{d \ d-1} = (p+1-d) \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}, \quad (2.30)$$

where the matrix  $\Delta_{d0} \in \mathbb{R}^{n+1-d \times n+1}$  is used to transform the control points of the curve to the control points of the  $d$ -th derivative of that curve.

### Bézier Spline with Parametric Continuity

Several Bézier curves can be connected to each other to form a Bézier spline, constructed of segments where each segment is created by a Bézier curve. The control points of segment  $i$  are denoted  $\mathbf{P}_{j,i}$ ,  $j \in \{0, \dots, p\}$ . These connected segments are then continuous up to the  $(p-1)$ -th derivative denoted by  $\mathcal{C}^{p-1}$  if

Figure 2.8: Bézier spline with continuity  $\mathcal{C}^2$ 

the following constraints are satisfied:

$$\mathcal{C}^0 : \begin{aligned} \mathbf{C}_i(1) &= \mathbf{C}_{i+1}(0) \\ \mathbf{P}_{p,i} &= \mathbf{P}_{0,i+1}, \end{aligned} \quad (2.31)$$

$$\mathcal{C}^1 : \begin{aligned} \frac{d}{du} \mathbf{C}_i(1) &= \frac{d}{du} \mathbf{C}_{i+1}(0) \\ (\mathbf{P}_{p,i} - \mathbf{P}_{p-1,i}) &= (\mathbf{P}_{1,i+1} - \mathbf{P}_{0,i+1}), \end{aligned} \quad (2.32)$$

$$\mathcal{C}^2 : \begin{aligned} \frac{d^2}{du^2} \mathbf{C}_i(1) &= \frac{d^2}{du^2} \mathbf{C}_{i+1}(0) \\ (\mathbf{P}_{p,i} - 2\mathbf{P}_{p-1,i} + \mathbf{P}_{p-2,i}) &= (\mathbf{P}_{2,i+1} - 2\mathbf{P}_{1,i+1} + \mathbf{P}_{0,i+1}). \end{aligned} \quad (2.33)$$

For Bézier curves of higher degree, higher degrees of continuity up until  $\mathcal{C}^{p-1}$  can be enforced in an analogous way. An example of a Bézier spline of degree  $p = 3$  is depicted in Figure 2.8, which shows that the control polygon of one segment is related to the control polygon of the next segment in the connection point of the two Bézier segments.

### Bézier Spline with Geometric Continuity

Parametric continuity is thus the result of each segment being continuous with respect to the parameter  $u$ . In practice, this parameter  $u$  often has no physical representation, which makes a parametric continuity constraint unnecessarily restrictive [108]. Instead, geometric continuity of degree  $k$   $\mathcal{G}^k$  is defined as continuity in the derivatives with respect to arc length  $s$ , rather than parameter  $u$  [109].

$$\frac{d^j}{ds^j} \mathbf{C}_{i+1}(0) = \frac{d^j}{ds^j} \mathbf{C}_i(0), \quad j \in \{0, \dots, k\}, \quad (2.34)$$

where arc length  $s$  is defined as

$$s_i(u) = \int_0^u \left\| \frac{d}{dx} \mathbf{C}_i(x) \right\|_2 dx.$$

This means that instead of (2.32) and (2.33), the tangent and curvature of the spline can be made continuous. To see that this yields additional degrees of freedom, observe that the tangent at the beginning and end of the Bézier curve

is determined only by the direction of the line segments spanned by  $\mathbf{P}_{0,i}$ ,  $\mathbf{P}_{1,i}$  and  $\mathbf{P}_{p-1,i}$ ,  $\mathbf{P}_{p,i}$ , where the length of these segments is not important. This can be generalized to so called *beta-constraints* [110], which for continuity up to  $\mathcal{G}^4$  are given by

$$\mathcal{G}^0 : \quad \mathbf{C}_{i+1}(0) = \mathbf{C}_i(1), \quad (2.35)$$

$$\mathcal{G}^1 : \quad \frac{d}{du} \mathbf{C}_{i+1}(0) = \beta_{1,i} \frac{d}{du} \mathbf{C}_i(1), \quad (2.36)$$

$$\mathcal{G}^2 : \quad \frac{d^2}{du^2} \mathbf{C}_{i+1}(0) = \beta_{1,i}^2 \frac{d^2}{du^2} \mathbf{C}_i(1) + \beta_{2,i} \frac{d}{du} \mathbf{C}_i(1), \quad (2.37)$$

$$\mathcal{G}^3 : \quad \frac{d^3}{du^3} \mathbf{C}_{i+1}(0) = \beta_{1,i}^3 \frac{d^3}{du^3} \mathbf{C}_i(1) + 3\beta_{1,i}\beta_{2,i} \frac{d^2}{du^2} \mathbf{C}_i(1) + \beta_{3,i} \frac{d}{du} \mathbf{C}_i(1), \quad (2.38)$$

$$\begin{aligned} \mathcal{G}^4 : \quad \frac{d^4}{du^4} \mathbf{C}_{i+1}(0) = & \beta_{1,i}^4 \frac{d^4}{du^4} \mathbf{C}_i(1) + 6\beta_{1,i}^2\beta_{2,i} \frac{d^3}{du^3} \mathbf{C}_i(1) \\ & + (4\beta_{1,i}\beta_{3,i} + 3\beta_{2,i}^2) \frac{d^2}{du^2} \mathbf{C}_i(1) + \beta_{4,i} \frac{d}{du} \mathbf{C}_i(1), \end{aligned} \quad (2.39)$$

where  $\beta_{1,i} > 0$  and  $\beta_{2,i}$ ,  $\beta_{3,i}$ ,  $\beta_{4,i}$  are arbitrary and represent the additional degrees of freedom. These additional degrees of freedom could for example be used to improve peak curvature, or to minimize some cost function. This can be generalized for higher degrees of geometric continuity as shown in [111], but these are not relevant for the application here, as  $\mathcal{G}^4$  is typically sufficient.

Where parametric continuity requires that the first and last line segment of two control polygons have the same length and are collinear, for geometric continuity only the collinear requirement remains. This is illustrated in Figure 2.9, which compares  $\mathcal{C}^2$  and  $\mathcal{G}^2$  continuity for a given set points that the Bézier spline must pass through. In this example,  $\beta_{j,i} = 0$ ,  $j > 1, \forall i$ . A specific choice for  $\beta_{1,i}$  is made, where the connecting part of the control polygons ( $\|\mathbf{P}_{1,i} - \mathbf{P}_{0,i}\|_2$  and  $\|\mathbf{P}_{p,i} - \mathbf{P}_{p-1,i}\|_2$ ) scale with the distance between the start and end of the Bézier segment ( $\|\mathbf{P}_{p,i} - \mathbf{P}_{0,i}\|_2$ ), that are usually known a priori as the points through which the spline should pass. This is achieved by setting  $\beta_{1,i} = \frac{\|\mathbf{P}_{p,i+1} - \mathbf{P}_{0,i+1}\|_2}{\|\mathbf{P}_{p,i} - \mathbf{P}_{0,i}\|_2}$ . From the figure it can be seen that using this specific choice for  $\beta_{1,i}$ , the lines that form the control polygon, scale with the overall dimension of the corresponding spline segment, making it a natural choice that in many cases results in lower peak curvatures.

### 2.2.2 B-splines

The Bézier curve is actually somewhat limited in terms of design freedom. The order of the Bézier curve is directly determined by the number of control points. This can be seen by the construction process of the Bézier curve. A B-spline [112]

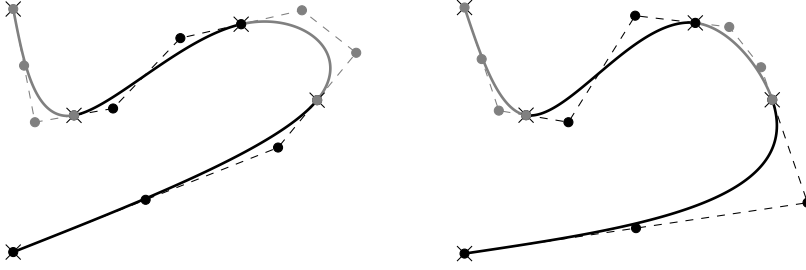


Figure 2.9: Comparison of continuity of Bézier curve of degree  $p = 3$ , interpolating points  $\mathbf{P}_{0,i}$ ,  $i \in \{0, 1, 2, 3\}$  and  $\mathbf{P}_{3,3}$ , where  $\mathbf{P}_{0,i+1} = \mathbf{P}_{3,i}$ ,  $i \in \{0, 1, 2\}$ . Pass-through points ( $\times$ ), Control polygon ( $- \bullet -$ ), Bézier curve ( $—$ ) Left: parametric continuity  $\mathcal{C}^2$ . Right: geometric continuity  $\mathcal{G}^2$ , with  $\beta_{1,i} = \|\mathbf{P}_{p,i+1} - \mathbf{P}_{0,i+1}\| \|\mathbf{P}_{p,i} - \mathbf{P}_{0,i}\|^{-1}$ ,  $\beta_{j,i} = 0$ ,  $j > 1, \forall i$

(or Basis-spline) is a more general description for a spline. In a B-spline, the number of control points and the degree of the spline are not directly linked due to the use of a knot vector, which is used to determine the basis functions. Given a non-decreasing knot vector  $\mathbf{U} = [u_0, \dots, u_{n+p}]$  where  $u_i \leq u_{i+1}$ , a degree  $p$  of the B-spline, and  $n + 1$  control points, the basis functions of a B-spline are recursively defined up to degree  $p$  as

$$N_{j,0}(u) = \begin{cases} 1 & \text{if } u_j \leq u < u_{j+1} \\ 0 & \text{otherwise} \end{cases}, \quad j \in \{0, \dots, n+p-1\}, \quad (2.40)$$

$$N_{j,k}(u) = \frac{u - u_j}{u_{j+k} - u_j} N_{j,k-1}(u) + \frac{u_{j+k+1} - u}{u_{j+k+1} - u_{j+1}} N_{j+1,k-1}(u), \quad j \in \{0, \dots, n+p-k\}. \quad (2.41)$$

Note that the basis functions have properties  $0 \leq N_{j,k}(u) \leq 1$  and  $\sum_j N_{j,k}(u) = 1$ ,  $\forall u$ . Additionally it should be noted that, in the case of repeated knots (e.g.,  $u_i = u_{i+1}$ ), the fraction in (2.41) can result in  $\frac{0}{0}$ . In the definition of B-splines, these terms are interpreted as zero.

To construct a B-spline  $\mathbf{C}(u)$ ,  $u \in [u_0, u_{n+p+1}]$  of degree  $p$ , the basis functions of degree  $p$  are simply multiplied by  $n + 1$  coefficients or ‘control points’  $\mathcal{P} = [\mathbf{P}_0 \ \mathbf{P}_1 \ \dots \ \mathbf{P}_n]^\top$  i.e.,

$$\mathbf{C}(u) = \sum_{j=0}^n N_{j,p}(u) \mathbf{P}_j = \mathcal{B}_{n,p}(u) \mathcal{P}, \quad (2.42)$$

$$\mathcal{B}_{n,p}(u) = [N_{0,p}(u) \ N_{1,p}(u) \ \dots \ N_{n,p}(u)]. \quad (2.43)$$

Generally, the B-splines do not pass through their control points at the beginning and the end, like Bézier curves do. To achieve this property, a clamped knot vector

must be used. Clamped knot vectors have  $p + 1$  repeated knots in the beginning and at the end, such that  $u_0 = u_1 = \dots = u_p$  and  $u_n = u_{n+1} = \dots = u_{n+p}$ , a uniformly distributed clamped knot vector distributes the remaining internal knots evenly over the interval spanned by the first and last knot:

$$\mathbf{U} = [\underbrace{u_0 \dots u_0}_{p+1} \underbrace{u_{p+1} \dots u_{n-1}}_{n-p} \underbrace{u_n \dots u_n}_{p+1}],$$

$$u_j = \frac{j-p}{n-p+1}(u_{n+p} - u_0), \quad j = \{p+1, \dots, n\}. \quad (2.44)$$

B-splines basis functions defined on a clamped knot vector have as property that  $N_{0,p}(0) = 1$ ,  $N_{j,p}(0) = 0$ ,  $j > 0$ , such that the spline coincides with the first control point, and  $N_{n,p}(u_{n+p+1}) = 1$ ,  $N_{j,p}(u_{n+p+1}) = 0$ ,  $j < n$ , such that the spline coincides with the final control point. Note that if a clamped knot vector with no internal knots is used and  $u_0 = 0$ ,  $u_{p+n} = 1$ , the B-spline reduces to a Bézier curve, as the corresponding basis functions are identical. This demonstrates that the Bézier curve is actually a special case of the B-spline. Due to the convenient property of a clamped knot vector, primarily B-splines with clamped knot vectors is used in the remainder of this thesis. Figure 2.10 shows a B-spline defined on a clamped knot vector.

**Lemma 2.1.** *The basis function  $N_{j,p}(u)$  defined on the clamped knot vector  $\mathbf{U}$  is only supported (i.e., non-zero) on the domain  $u \in [u_j, u_{j+p+1}]$ .*

*Proof.* Using (2.41) it can be seen that  $N_{j,p}(u)$  is the linear combination of  $N_{j,p-1}(u)$  and  $N_{j+1,p-1}(u)$ . As a result it is only supported on the domain on which either  $N_{j,p-1}(u)$  or  $N_{j+1,p-1}(u)$  is supported. Continuing this recursion it can be seen that  $N_{j,p}(u)$  depends only on the zero-order functions  $N_{k,0}(u)$ ,  $j \in \{j, \dots, j+p+1\}$  and in turn can only be supported on the domain of these functions. Using (2.40) it can be seen that these functions  $N_{k,0}$  are supported on the domain  $u = [u_k, u_{k+1}]$ , which completes the proof.  $\square$

## Derivatives and Primitives

Just like the derivative of a Bézier curve being a Bézier curve, the derivative of a B-spline is also again a B-spline. As such, the spline  $\mathbf{C}(u)$  as well as its derivatives can be expressed as a function of the original control points. Using

$$\frac{d^d}{du^d} N_{j,p} = \frac{p+1-d}{u_{j+p+1-d} - u_j} N_{j,p-d}(u) - \frac{p+1-d}{u_{j+p+2-d} - u_{j+1}} N_{j+1,p-d}(u), \quad (2.45)$$

the control points of the  $d$ -th derivative of the B-spline defined on a clamped knot vector are found similarly to that of the Bézier curve in (2.30), by means of the

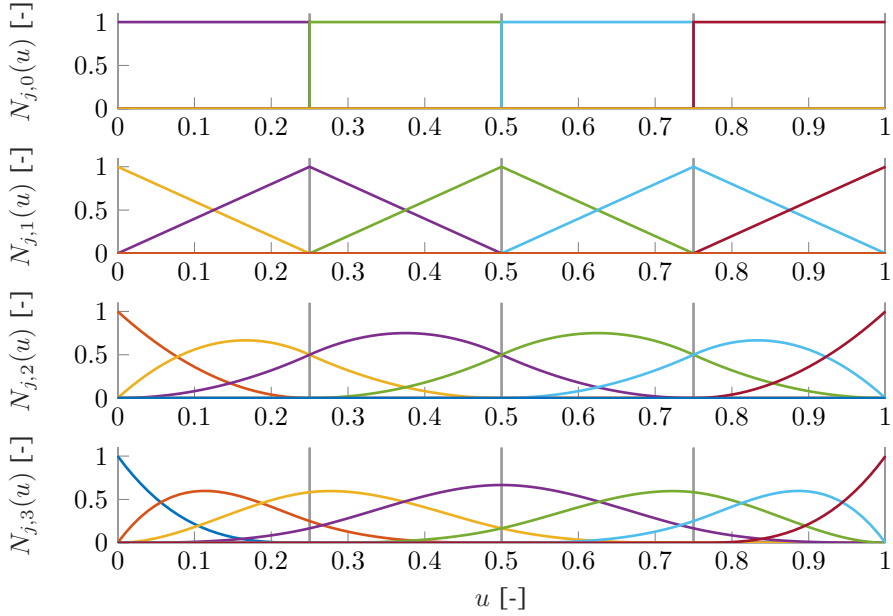


Figure 2.10: Basis functions for  $p = 3$ ,  $N_{0,k}$  (—),  $N_{1,k}$  (—),  $N_{2,k}$  (—),  $N_{3,k}$  (—),  $N_{4,k}$  (—),  $N_{5,k}$  (—),  $N_{6,3}$  (—) with  $k \leq p$ , knots (—) on knot vector  $\mathbf{U} = [0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1]$ .

hodograph transformation

$$\mathcal{P}^{(d)} = \underbrace{\Delta_{d-1}(\mathbf{U}) \Delta_{d-1-d-2}(\mathbf{U}) \Delta_{21}(\mathbf{U}) \Delta_{10}(\mathbf{U})}_{\Delta_{d0}} \mathcal{P}, \quad (2.46)$$

$$\delta_i = \frac{p+1-d}{u_{i+p+d} - u_{i+d}}, \quad (2.47)$$

$$\Delta_{d-1}(\mathbf{U}) = \begin{bmatrix} \delta_0 & & & \\ & \delta_1 & & \\ & & \ddots & \\ & & & \delta_{n-d} \end{bmatrix} \begin{bmatrix} -1 & 1 & & \\ & -1 & 1 & \\ & & \ddots & \ddots \\ & & & -1 & 1 \end{bmatrix}, \quad (2.48)$$

in which  $\mathcal{P}^{(d)}$  denotes the control points associated with the  $d$ -th derivative. Note that  $\Delta_{d0} \in \mathbb{R}^{n+1-d \times n+1}$ . Hence, for each derivative, the number of control points is reduced by one. The derivative spline is then computed by means of

$$\mathbf{C}^{(d)}(u) = \mathcal{B}_{n-d,p-d}(u) \mathcal{P}^{(d)}. \quad (2.49)$$

Also note that the degree of the  $d$ -th derivative spline is  $p-d$ . The basis functions corresponding to the derivative splines are defined on a knot vector that has one



of the clamped knots on either side removed, such that the multiplicity of the clamped knots is again one lower than the degree of the spline. The resulting knot vector corresponding to the  $d$ -th derivative spline, denoted by  $\mathbf{U}^{(d)}$ , is then equal to

$$\mathbf{U}^{(d)} = [\underbrace{u_0 \dots u_0}_{p+1-d} \underbrace{u_{p+1} \dots u_{n-1}}_{n-p} \underbrace{u_n \dots u_n}_{p+1-d}], \quad (2.50)$$

where the number of knots has changed with respect to the original knot vector of (2.44). It can be shown that  $N_{i+1,p-1}(u)$  evaluated on the original knot vector ( $\mathbf{U}$ ) is equal to  $N_{i,p-1}(u)$  on the new knot vector ( $\mathbf{U}^{(d)}$ ). Note that  $\Delta_{d0}$  in (2.46) depends on the knot vector  $\mathbf{U}$ , due to the dependency on knots  $u_j$  in (2.47). Similarly, the primitive of a B-spline is again a B-spline. The coefficients of the primitive spline can be found by

$$\mathcal{P}^{(-d)} = \underbrace{\Xi_d^{(d-1)}(\mathbf{U}) \Xi_{(d-1)(d-2)}(\mathbf{U}) \Xi_{21}(\mathbf{U}) \Xi_{10}(\mathbf{U})}_{\Xi_{d0}} \mathcal{P}, \quad (2.51)$$

$$\Xi_d^{(d-1)} = \begin{bmatrix} 0 & \dots & \dots & \dots & 0 \\ \xi_0 & \ddots & & & \vdots \\ \xi_0 & \xi_1 & \ddots & & \vdots \\ \xi_0 & \xi_1 & \xi_2 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ \xi_0 & \xi_1 & \xi_2 & \dots & \xi_{n+d} \end{bmatrix}, \quad \xi = \frac{u_{j+p+1} - u_j}{p+d}, \quad (2.52)$$

and the primitive is computed as

$$\mathbf{C}^{(-d)}(u) = \mathcal{B}_{n+d,p+d}(u) \mathcal{P}^{(-d)}. \quad (2.53)$$

Note that in order to define basis functions up until degree  $p+d$ , additional knots are required. Similarly to how knots are removed for the derivative of a spline, clamped knots are added for the primitive of a spline at the start and end of the knot vector as

$$\mathbf{U}^{(-d)} = [\underbrace{u_0 \dots u_0}_{p+1+d} \underbrace{u_{p+1} \dots u_{n-1}}_{n-p} \underbrace{u_n \dots u_n}_{p+1+d}], \quad (2.54)$$

where the number of knots is again different from the original knot vector (2.44). More details of integrating B-splines can be found in [113].

**Lemma 2.2.** *Given a B-spline  $C(u)$  of degree  $p$  defined on the clamped knot vector  $\mathbf{U}$  without internal knots with multiplicity greater than one, the value at  $u = 0$  of the  $d$ -th derivative only depends on the first  $d+1$  control points  $P_i$ ,  $j \in \{0, \dots, d\}$ .*

*Proof.* The clamped knot vector  $\mathbf{U}$ , has  $p + 1$  repeated knots at the start and beginning, such that  $u_j = 0$ ,  $j \in \{0, \dots, p\}$  and  $u_j = 1$ ,  $j \in \{n, \dots, n + p\}$ . For the remaining knots it holds that  $u_j < u_{j+1}$ ,  $j \in \{p, \dots, n - 1\}$ .

Hence, following (2.40), it can be seen that  $N_{p,0}(0) = 1$  and  $N_{j,0}(0) = 0$ ,  $j \neq p$ .

Then using (2.41), it follows that the only non-zero basis function at  $u = 0$  is  $N_{p-j,j}(0) = 1$ ,  $j \in \{0, \dots, p\}$ . Consequently, the spline value of the spline  $C^{(d)}(u)$  is equal to the control point  $P_0^{(d)}$  following (2.49).

Then following (2.46) and using the structure in (2.48), it is clear that  $P_0^{(d)}$  depends on the control points  $P_i$ ,  $j \in \{0, \dots, d\}$ , which completes the proof.  $\square$

### Control Polygon for 1-D Graph

Traditionally, splines are often used for drawing shapes in 2D (or 3D), where both the control points  $\mathbf{P}_i$  as well as the spline  $\mathbf{C}(u)$  are 2D (or 3D), as for example was done in Figure 2.9. For these cases, a control polygon can be drawn by simply connecting the control points  $\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n$  by means of a line. The control polygon dictates the shape of the spline, as B-splines have the *strong* convex hull property. This follows from the B-spline  $\mathbf{C}(u)$ ,  $u \in [u_i, u_{i+1})$  being in the convex hull of the subset of control points  $\{\mathbf{P}_j \mid j \in \{i, i - 1, \dots, i - p\}\}$ , rather than the convex hull of all control points.

A similar control polygon can also be constructed in the parameter space (e.g., in a plot of each dimension of  $\mathbf{C}(u)$  plotted against parameter  $u$ ). This is useful if  $\mathbf{C}(u)$  is one dimensional (e.g.,  $C(u)$ ), as it allows to visualize the effect of each of the control points on the graph of  $C(u)$ . The coordinate in  $u$  corresponding to each control point is called a Greville abscissa [114] and is found via

$$\mu_j = \frac{1}{p} \sum_{k=1}^p u_{j+k}, \quad j \in \{0, \dots, n\}. \quad (2.55)$$

An example of such a control polygon for a spline is shown in Figure 2.11, which shows the spline and control polygon of a 1D B-spline. It can be seen that, similar to the control polygon in  $\mathbb{R}^n$ , the spline starts and ends in the first control point, and the spline is tangent to the first and last segment of the control polygon at the start and end of the spline. The slope of the control polygon in the parameter space for a clamped knot vector ( $u_0 = u_1 = \dots = u_p = 0$ ) is

$$\frac{P_1 - P_0}{\mu_1 - \mu_0} = \frac{P_1 - P_0}{\frac{1}{p} \sum_{k=1}^p (u_{1+k} - u_k)} = \frac{P_1 - P_0}{\frac{1}{p} (u_{1+p} - u_p)}, \quad (2.56)$$

which, for a clamped knot vector is indeed equal to the derivative of the spline which follows from (2.49) and (2.47) as

$$\frac{d}{du} C(0) = P_0^{(1)} = \frac{p}{u_{p+1} - u_1} (P_1 - P_0). \quad (2.57)$$

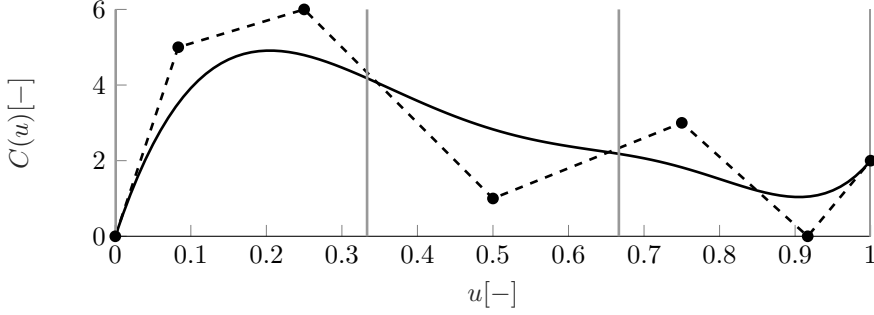


Figure 2.11: B-spline  $C(u) \in \mathbb{R}^1$  (—) and control polygon (- • -) constructed by Greville abscissae and control point  $\{\mu_j, P_j\}$ . Knots  $u_i, i \in \{0, \dots, n + p\}$  (—).

Hence, it can be seen that the Greville abscissae are indeed suitable to construct the control polygon.

## 2.3 Reference Trajectory

A reference trajectory can be used as a nominal trajectory to be used by automated vehicles. A reference trajectory is constructed by means of two elements, the spatial reference path and a reference velocity. This section discusses how to construct the reference path by means of Bézier splines, and the construction of the reference velocity related to that reference path, for the use in automated vehicles.

### 2.3.1 Reference Path

The reference path is the spatial reference for the automated vehicle. This spatial reference can be obtained a priori by means of Highly Automated Driving (HAD) maps or from historic data, or can be generated in real-time based on local perception [14]. To ensure that the vehicle is actually capable of following the reference path, the path should satisfy certain requirements. These requirements are related to the limitations of the vehicle. For a given forward velocity, the yaw rate of the vehicle is determined by the steering angle. As the vehicle is not capable of instantaneously changing the steering angle, the vehicle is also incapable of instantaneously changing the yaw rate of the vehicle, which in turn implies that the curvature cannot be changed instantaneously. Note that, even if the steering angle could be changed instantaneously, vehicle dynamics induced by tyre dynamics still prohibits instantaneous changes in yaw rate. Hence, for a non-zero forward velocity, curvature should be at least continuous. This improves the tracking of the path by the automated vehicle [115].

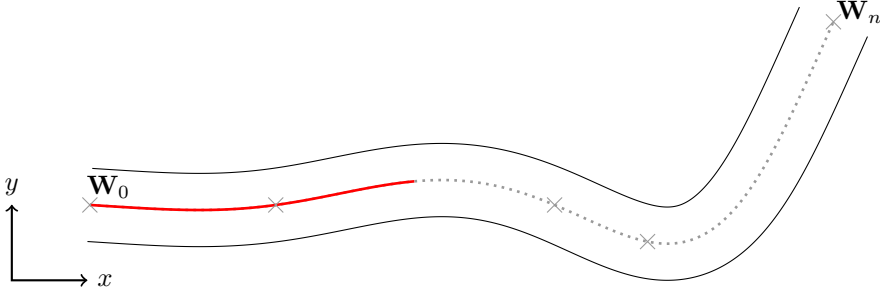


Figure 2.12: Centerline of the road fitted based on lane information. Centerline reference (.....), Waypoints ( $\times$ ), Lane Boundary (—), curvilinear distance  $s$  (—)

Moreover, as steering wheel torque is bounded, the rate at which the steering wheel turns also cannot change instantaneously, which implies that the curvature of the path should not only be continuous, but also continuously differentiable with respect to arc length.

### Bézier Reference Fitting

The spatial reference path  $\mathbf{X}_c(s)$  is generated by means of *way-points*. An example is given in Figure 2.12. These way-points are points in space which the reference path should pass through. The path could be interpolated by either a B-spline or Bézier spline. The advantage of a B-spline with respect to a Bézier curve (separating  $p$  from  $n$ ), is not relevant here, as a Bézier spline can do the same. The added benefit of the Bézier spline, is that it allows to easily use both geometric and parametric continuity. Finding the current state  $\mathbf{S}_i(t)$ ,  $\mathbf{\Lambda}_i(t)$  is then found by projecting on this Bézier spline [116]. As the Bézier splines pass through their control points  $\mathbf{P}_{0,i}$  and  $\mathbf{P}_{p,i}$ , it is natural to define these control points to be equal to these way-points. The remaining control points of the Bézier segments are constrained by means of geometric continuity requirements. The curvature of the path should be continuous differentiable with respect to length, such that the spline is at least  $\mathcal{G}^3$  continuous.

Given  $N+1$  way-points  $\mathbf{W}_i$ ,  $i \in \{0, \dots, N\}$ , an open Bézier spline (as opposed to a closed curve in which first and last segments are connected to form a loop) of degree  $p$  with  $\mathcal{G}^{p-1}$  continuity can be constructed by means of  $N$  Bézier segments of degree  $p$ , as shown in Section 2.2.1. Control points  $\mathbf{P}_{j,i}$ ,  $j \in \{0, \dots, p\}$ ,  $i \in \{0, \dots, N-1\}$  can then be constructed. Note that each Bézier segment contains  $p+1$  control points. For  $N$  Bézier segments, this results in  $N(p+1)$  degrees of freedom for the curve fitting process. As the Bézier curve passes through  $\mathbf{P}_{0,i}$  and  $\mathbf{P}_{p,i}$  and  $\mathcal{G}^0$  should be satisfied, the first and last control points of each segment

are constrained as

$$\mathbf{P}_{0,i} = \mathbf{W}_i, \quad i \in \{0, \dots, N-1\}, \quad (2.58)$$

$$\mathbf{P}_{p,i} = \mathbf{W}_{i+1}, \quad i \in \{0, \dots, N-1\}, \quad (2.59)$$

which fixes  $2N$  degrees of freedom. Note that  $\mathcal{G}^0$  continuity is then already implied, as  $\mathbf{P}_{p,i} = \mathbf{P}_{0,i+1}$ . Then to satisfy  $\mathcal{G}^k$ ,  $k \in \{1, \dots, p-1\}$

$$\beta_{1,i}^k \frac{d^k}{du^k} \mathbf{C}_i(1) = \frac{d^k}{du^k} \mathbf{C}_{i+1}(0), \quad i \in \{0, \dots, N-2\}, \quad (2.60)$$

which fixes  $(N-1)(p-1)$  degrees of freedom. Note that  $\beta_{j,i} = 0$ ,  $j > 1, \forall i$  is chosen for simplicity. The remaining  $p-1$  degrees of freedom correspond to the first and last segments. For open curves, the higher derivatives at the start and end of the curve can be used to fix the remaining degrees of freedom as

$$\frac{d^k}{du^k} \mathbf{C}_0(0) = 0, \quad k \in \{1, \dots, \lceil (p-1)/2 \rceil\}, \quad (2.61)$$

$$\frac{d^k}{du^k} \mathbf{C}_{N-1}(1) = 0, \quad k \in \{1, \dots, \lfloor (p-1)/2 \rfloor\}, \quad (2.62)$$

where in case of even  $p$ , an additional constraint is assigned to the start of the spline. For a closed Bézier spline (i.e.,  $\mathbf{W}_0 = \mathbf{W}_N$ ), the final constraints can instead be set to

$$\beta_{1,i}^k \frac{d^k}{du^k} \mathbf{C}_N(1) = \frac{d^k}{du^k} \mathbf{C}_0(0), \quad k \in [1, \dots, p-1]. \quad (2.63)$$

These constraint equations can be solved for a given  $\beta_{1,i}$  as a linear system to find the control points  $\mathbf{P}_{j,i}$ ,  $j \in \{0, \dots, p\}$ ,  $i \in \{0, \dots, N-1\}$  for both the closed spline as well as the open spline. A MATLAB toolbox that achieves this process of fitting Bézier curves is made available at [117].

### Continuity Comparison

Although  $\mathcal{G}^3$  continuity is required, higher orders of continuity could result more preferred paths, for example by a lower peak curvature. The above process can also be used to find Bézier splines of higher-order continuity. A comparison has been performed on a real world test circuit used for autonomous driving. Figure 2.13 shows a comparison of fits with various geometric continuity properties and corresponding degrees of Bézier splines. It can be seen that the  $\mathcal{G}^3$  and  $\mathcal{G}^5$  continuity fits have significantly higher curvatures than the  $\mathcal{G}^4$  and  $\mathcal{G}^2$  fits. This result, combined with the requirement that the continuity is at least  $\mathcal{G}^3$ , motivates the choice for  $\mathcal{G}^4$  fitting for the reference path.

Similarly,  $\mathcal{G}^4$  continuity fitting is compared with  $\mathcal{C}^4$  continuity fitting on the same real world track in Figure 2.14. The advantages of  $\mathcal{G}^k$  fitting with

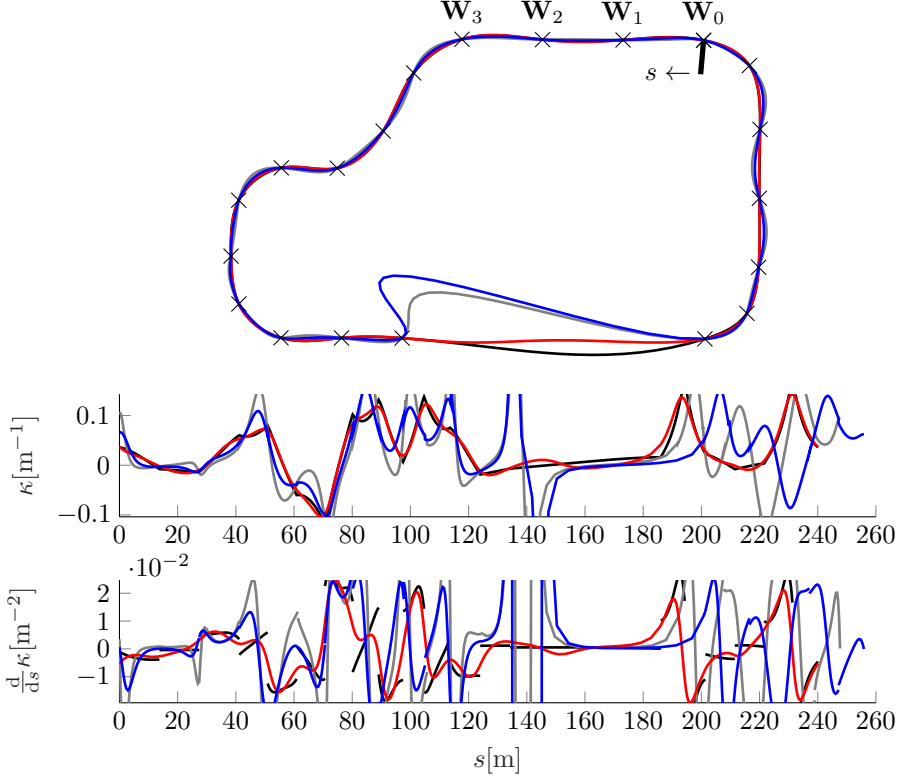


Figure 2.13: Comparison of various geometrical continuity fits with  $\beta_{1,i} = \frac{\|\mathbf{P}_{p,i+1} - \mathbf{P}_{0,i+1}\|}{\|\mathbf{P}_{p,i} - \mathbf{P}_{0,i}\|} (\|\mathbf{P}_{p,i} - \mathbf{P}_{0,i}\|)^{-1}$ ,  $\beta_{j,i} = 0$ ,  $j > 1, \forall i$ , Way-points (x),  $\mathcal{G}^2$  fit (—),  $\mathcal{G}^3$  fit (—),  $\mathcal{G}^4$  fit (—),  $\mathcal{G}^5$  fit (—)

$\beta_{1,i} = \frac{\|\mathbf{P}_{p,i+1} - \mathbf{P}_{0,i+1}\|}{\|\mathbf{P}_{p,i} - \mathbf{P}_{0,i}\|}$  over  $\mathcal{C}^4$  fitting, become clear when observing the lower right segment of the fitted track. Since the segment on the straight part and the next segment in the curve significantly differ in length, the  $\mathcal{C}^4$  fit has an issue in matching the control polygon segment lengths. This issue is not present in the  $\mathcal{G}^4$  fit, as the length of the line segment is scaled by the approximate length of the Bézier segment. The resulting  $\mathcal{G}^4$  fit has significantly lower peak curvatures and curvature derivatives, resulting in a more comfortable track to drive. This again supports the decision to use geometric continuity rather than parametric continuity.

It should be noted that the presented method utilizes spline interpolation. Another option is to use spline approximation. Approximation could be used to include a point cloud of way-points which the spline should approximately describe. A cost function can then be constructed that penalizes Euclidean

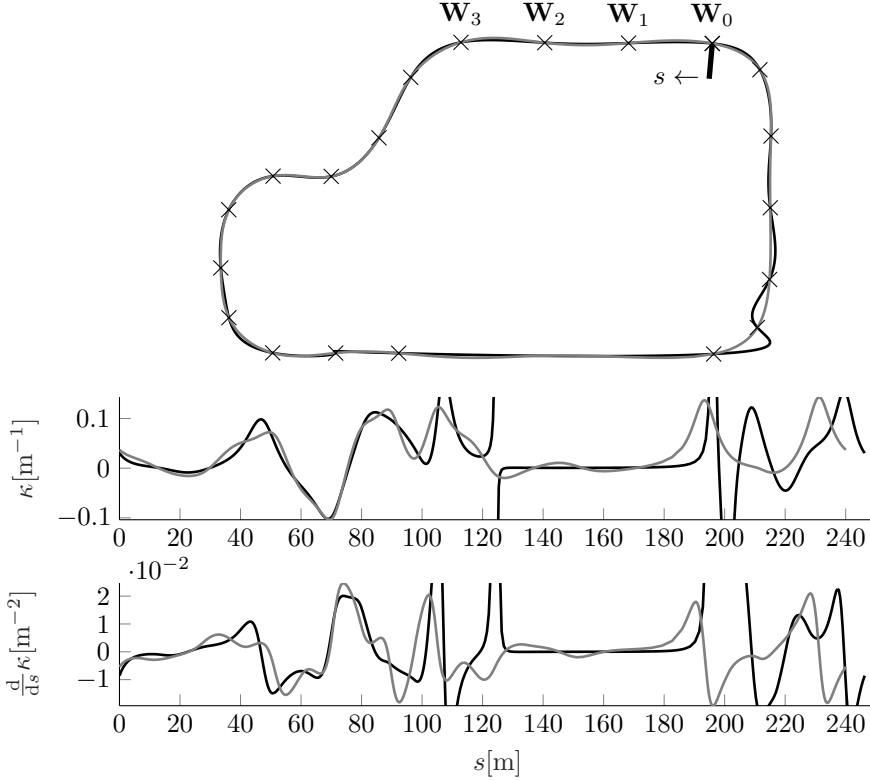


Figure 2.14: Comparison of  $C^4$  and  $G^4$  with  $\beta_{1,i} = \|\mathbf{P}_{p,i+1} - \mathbf{P}_{0,i+1}\| (\|\mathbf{P}_{p,i} - \mathbf{P}_{0,i}\|)^{-1}$ ,  $\beta_{j,i} = 0, j > 1, \forall i$ , Way-points ( $\times$ ),  $C^4$  fit (—),  $G^4$  fit (---)

distance to these points, and could possibly also contain a term penalizing the curvature of the curve. Note that this approach does not change anything fundamental in the trajectory planner framework, and as such can simply be seen as an extension of the work presented here.

### 2.3.2 Reference Velocity

For a given spatial reference path  $\mathbf{X}_c(s)$ , a reference velocity  $v_c(s)$  needs to be constructed. The resulting trajectory should satisfy certain requirements. It should traverse the path as fast as possible while satisfying certain user and environmental constraints. The problem of minimum time velocity profiles for road vehicles is well studied in literature, for point mass models [118], half car models [119], as well as more complex vehicle models [120][121]. These models accurately capture the limits of the vehicle's dynamics. However, For

typical on-road driving, the vehicle will not be operating near the limit. Therefore, a simple point mass model suffices for the application of generating a reference velocity.

In order to drive a fixed spatial path in a comfortable and legal way, certain considerations have to be taken into account. Naturally, each road segment has a certain corresponding maximum legal velocity. Additionally, user comfort should be taken into account in curves, such that accelerations should not be too high. A speed profile can be constructed by considering both the legal driving velocity, as well as the maximum user specified planar acceleration.

To traverse a fixed spatial path with curvature  $\kappa_c(s)$  along arc length  $s$ , in minimum time, given a planar acceleration limit  $\|\ddot{x}(s(t)), \ddot{y}(s(t))\|_2 \leq a_{\max}$ ,  $\forall s$  and maximum legal velocity  $v_{\text{leg}}(s)$ , the maximum velocity  $v_c(s)$  along the curve can be computed as following. First, all values of arc length  $s$  for which the  $v_c(s)$  might become a local minimum are identified. These are either local maxima of curvature, or changes (often discontinuous) in the legal speed limit.

The local maxima of  $\arg \min |\kappa_c(s)|$  are identified, and denoted by  $s \in \mathcal{S}_{\text{curv}}$ . In these points the velocity is upper bounded by  $v_c^2 \kappa_c$  due to the acceleration bound. Next, potential a local minimum of  $v_c(s)$ , can also be at a local minimum of  $v_{\text{leg}}(s)$ . Typically, the legal velocity for any given road is piecewise constant for certain sections of the road. For open curves, the legal velocity at the final known point of the considered distance horizon is set to 0. This ensures that if the vehicle reaches the end of the known reference path, before this reference

---

**Algorithm 2.1** Reference Velocity Algorithm

---

```

procedure VMAX( $\kappa_c(s), v_{\text{leg}}(s), \mathcal{S}_{\text{loc}}$ )
     $v_{\max}(s_k) \leftarrow \min \left[ \sqrt{\frac{a_{\max}}{\kappa_c(s_k)}}, v_{\text{leg}}(s_k) \right], \quad \forall s_k \in \mathcal{S}_{\text{loc}} \quad \triangleright$  Initialize  $v_{\max}(s_k)$ 
    while  $\mathcal{S}_{\text{loc}} \neq \emptyset$  do
         $s_k = \arg \min v_{\max}(s_k), \forall s_k \in \mathcal{S}_{\text{loc}} \quad \triangleright$  use point with lowest maximum
        velocity
        Construct  $v_{\text{acc}}(s), s \in [s_k, s_{k+1}]$ 
        Construct  $v_{\text{dec}}(s), s \in (s_{k-1}, s_k]$ 
         $\triangleright$  Update maximum velocity for neighbouring points
         $v_{\max}(s_{k-1}) \leftarrow \min[v_{\max}(s_{k-1}), v_{\text{dec}}(s_{k-1})]$ 
         $v_{\max}(s_{k+1}) \leftarrow \min[v_{\max}(s_{k+1}), v_{\text{acc}}(s_{k+1})]$ 
         $\triangleright$  Note that for closed curves  $s_1 = s_N$  while updating neighbours
         $\mathcal{S}_{\text{loc}} \leftarrow \mathcal{S}_{\text{loc}} \setminus s_k \quad \triangleright$  Remove  $s_k$  from  $\mathcal{S}_{\text{loc}}$ 
    end while
     $v_c(s) = \min[v_{\text{leg}}(s), v_{\text{acc}}(s), v_{\text{dec}}(s)]$ 
    return  $v_c(s)$ 
end procedure

```

---



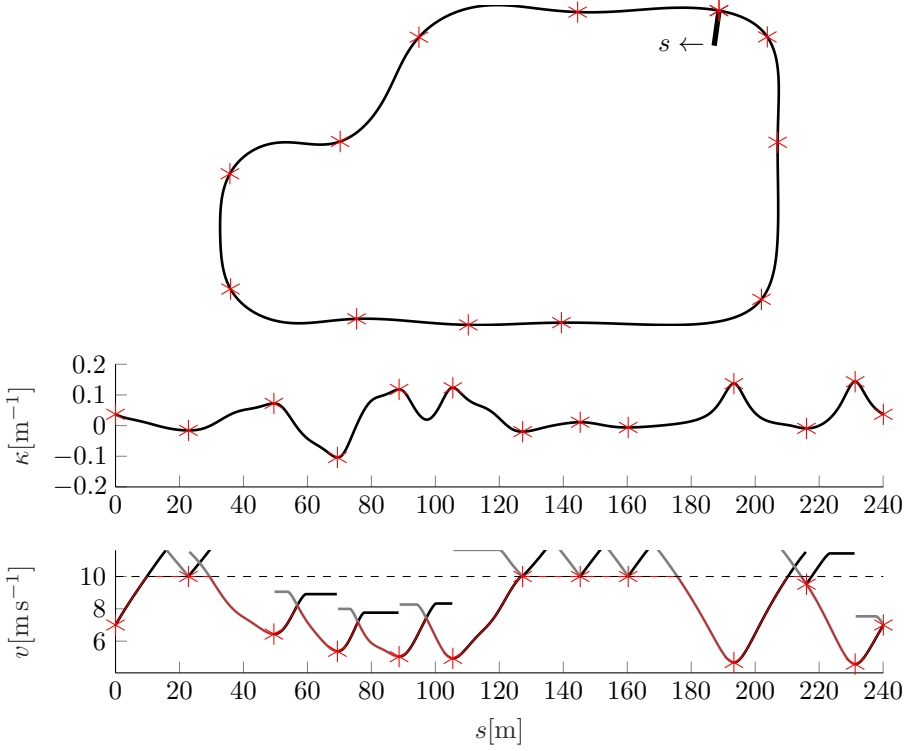


Figure 2.15: Construction of reference velocity  $v_c(s)$  (—), constructed by means of deceleration profiles  $v_{\text{dec}}(s)$  (—), acceleration profiles  $v_{\text{acc}}(s)$  (—), and maximum legal velocity  $v_{\text{leg}}(s)$  (---). points of potential minimal velocity  $S_{\text{loc}}$  (\*).

path is updated, it will come to a standstill. The locations where the legal velocity is discontinuous can also become local minima of  $v_c(s)$ . Both locations where  $v_{\text{leg}}(s)$  has a local minimum, as well as locations where  $v_{\text{leg}}(s)$  is discontinuous are denoted by  $s \in S_{\text{leg}}$ . Finally the set where the velocity profile can attain a local minimum is the union of  $S_{\text{leg}}$  and  $S_{\text{curv}}$ ,  $S_{\text{loc}} = S_{\text{curv}} \cup S_{\text{leg}}$ . The  $N$  elements in  $s_k \in S_{\text{loc}}$ ,  $k \in \{1, \dots, N\}$  are ordered such that  $s_k \leq s_{k+1}$ . Then Algorithm 2.1 is used to determine  $v_c(s)$ .

Starting with the point  $\arg \min v_{\text{max}}(s_k)$  with lowest velocity, a decreasing velocity profile  $v_{\text{dec}}(s)$  and an increasing velocity profile  $v_{\text{acc}}(s)$  are computed from and towards the point  $\{s_k, v_{\text{max}}(s_k)\}$  by means of forward and backwards integration. The acceleration profile is constructed by initializing  $v_{\text{acc}}(s_k) = v_{\text{max}}(s_k)$ . Then by taking into account the lateral acceleration required to traverse the current curvature  $\kappa_c(s)$  with velocity  $v_{\text{acc}}(s)$ , the velocity

profile  $v_{\text{acc}}(s)$ ,  $s \in [s_k, s_{k+1})$  is computed by means of the longitudinal acceleration  $a_x(s) = \sqrt{a_{\text{max}}^2 - \kappa_c^2 v_{\text{acc}}^2(s)}$ . This norm of this longitudinal acceleration combined with lateral acceleration, corresponding to the velocity and curvature, is then equal to the specified acceleration constraint  $a_{\text{max}}$ . Equivalently,  $v_{\text{dec}}(s)$ ,  $s \in (s_{k-1}, s_k]$  is constructed by means of backwards integration. If at some point  $a_{\text{max}}^2 < \kappa_c^2 v_{\text{acc}}^2(s)$  then  $a_x(s) := 0$ . This process is illustrated in Figure 2.15, where a spatial path is shown, together with the corresponding curvature  $\kappa_c(s)$ .

## 2.4 Summary

This chapter has presented background information on three important topics. The motion planning problem for the automated vehicle is reduced to a problem of planning trajectories  $s(t)$  and  $\ell(t)$  on the time horizon  $t \in [t_0, t_0 + T]$ , where in  $s$  and  $\ell$  are defined by means of the Frenet frame with respect to reference path  $\mathbf{X}_c(s)$ . A suitable solution for this motion planning problem is to create a set of candidate trajectories, which are tested for feasibility and collisions. A cost function is used to select the optimal trajectories from the feasible and collision free trajectories.

Additionally, mathematical background is provided for the description of Bézier and B-spline curves and splines. This included basic mathematical operations such as differentiation (in the form of hodographs) and integration. The joining of multiple Bézier segments was discussed with both parametric (e.g.,  $C^n$ ) as well as geometric (e.g.,  $\mathcal{G}^n$ ) continuity was discussed.

These Bézier splines are then used to construct spatial reference path  $\mathbf{X}_c(s)$ . A comparison was shown between various types of continuities, and a Bézier spline with a specific case of  $\mathcal{G}^4$  continuity was selected to be a suitable candidate for the spatial reference path. A reference velocity  $v_c(s)$  is then constructed by means of Algorithm 2.1, while satisfying the user- or system specified acceleration limit  $a_{\text{max}}$  and abiding the speed limit  $v_{\text{leg}}(s)$ .



# Chapter 3

## Cooperative Trajectory Planning

Cooperative trajectories attempt to realize a certain collective behavior. In the scope of this thesis, that collective behavior is specifically vehicle following. By sharing information between vehicles, safe vehicle following at short inter vehicles distances can be achieved. In this chapter, several strategies for the construction of cooperative trajectories are presented. One of these strategies, that utilizes B-splines to construct trajectories forms the basis of a framework for trajectory planning for cooperative vehicles. The chapter is structured as follows. Section 3.1, presents the objectives and requirements of these cooperative trajectories. In Section 3.2, a method using polynomials is used, which connect a initial state to a terminal state, and uses a shift in arrival time to prevent string-instability in scenarios where a platoon of vehicles transitions from one steady state to another. However, this solution shows to only work well in the absence of disturbances, making it unsuitable for implementation. Then in Section 3.3, the main contribution, based on [122], is presented in the form of the construction of cooperative trajectories by means of B-splines. A numerical study is performed to assess the behavior of a string of vehicles, using this trajectory planner, in terms of string stability. In Section 3.4, a modification based on [56] is presented that improves the passenger comfort in cases of gap closing.

Regardless of the approach taken for trajectory planning, the implementation has to be updated on-line in a receding horizon fashion. As discussed in Section 2.1.4, by not considering inequality constraints in the trajectory construction, a hard limit for the time required to compute a trajectory is found, which is beneficial for real-time implementation of a receding horizon approach. In the remainder of this chapter, the following notation is used.  $x_i(t)$ ,  $t \in \mathbb{R}^+$  refers to the signal  $x$  corresponding to vehicle  $i$  defined on all time  $t \in \mathbb{R}^+$ , whereas  $x_{i,k}(t)$ ,  $t \in [t_{i,k}, t_{i,k} + T_i]$  denotes the planned trajectory for signal  $x_i(t)$  in plan step  $k$ , defined on the time horizon  $t \in [t_{i,k}, t_{i,k} + T_i]$ .

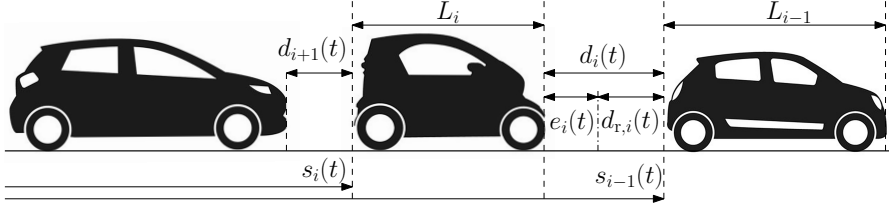


Figure 3.1: Schematic representation of a platoon. Actual inter-vehicle distance,  $d_i(t)$  is regulated towards the desired inter-vehicle distance  $d_{r,i}(t)$ .

### 3.1 Cooperative trajectory objective

Cooperative driving can have many applications in practice, of which some examples were already listed in Section 1.1.2. In cooperative driving, multiple vehicles *cooperate* to realize a certain *collective behavior*. In the context of this work, the collective behavior of interest is cooperative vehicle following at close inter-vehicle distances. In cooperative vehicle following, a certain spacing policy is used to describe the desired inter-vehicle distance between two consecutive vehicles. By defining the desired spacing between vehicles in the Frenet frame instead of in the vehicle or Cartesian frame, spacing is independent from lateral movement within, or between road lanes. Only the curvilinear trajectories,  $S_{i,k}(t)$  are considered, since the lateral trajectories  $\Lambda_{i,k}(t)$  are not critical to the performance of the string in the context of the spacing policy. Also note that, as explained in Section 2.1.4, and visualized in Figure 2.4, each  $S_{i,k}(t)$  has a multitude of lateral trajectories associated with it. By placing multiple of such vehicles behind each other a *platoon* can be formed. A section of such a platoon can be seen in Figure 3.1, which illustrates vehicles with arc length coordinates in the Frenet frame  $s_i(t)$  (see Section 2.1.3), together with the desired and actual inter-vehicle distances, denoted by respectively  $d_{r,i}(t)$  and  $d_i(t)$ .

Several aspects are important regarding a trajectory planning method for vehicles in such a platoon. First of all, this control system faces hard constraints on computation times [21], since it is operating in safety-critical situations. This is especially true with vehicle following at short inter vehicle distances. Hence, it is important that it can be guaranteed that a new trajectory is available for execution when the vehicle needs one.

Additionally, since the vehicles cooperate, the planned trajectories need to be communicated along the vehicle string. The required bandwidth for this communication is also important. Note that the trajectory  $S_i(t)$  can be transmitted in various ways. Intuitively, due to the discrete-time implementation originating from the real-time implementation on the host vehicle, it would make sense to simply communicate the planned trajectory sampled over the planning horizon. However, as the planning horizon and update rate of the

real-time computer may be large, the amount of data that needs to be transmitted is also large. Naturally, a larger data block that needs to be transmitted, may result in larger latency, which is undesirable. Hence, parametrized trajectories seem a better option.

### 3.1.1 Temporal Consistency

Aside from requirements on the trajectory planning method, certain design requirements can also be placed on the generated trajectories itself. An important example is temporal consistency [18]. This refers to the recursive implementation of the trajectory planner, where the overlapping part of the time horizon of trajectories planned in two planning cycles matches in case the situation has not changed. This is important as a following vehicle uses a transmitted trajectory to construct its own trajectory. If consecutively planned trajectories do not match, a following vehicle might anticipate and react on a future motion of the host vehicle that will not take place. Additionally, if consecutive trajectories do not significantly differ, previous trajectories can be utilized in case of packet drop-out in the wireless communication for robustness.

A performance indicator can be used to indicate to what extent the trajectory  $S_{i,k}(t)$ ,  $t \in [t_{i,k}, t_{i,k} + T_i]$  of plan step  $k$  overlaps with the trajectory of another plan step. Planning times are updated with planning interval  $t_p$ , such that  $t_{i,k+1} = t_{i,k} + t_p$ . As performance indicator, the maximum difference between plan step  $k$  and plan step  $j$  (with  $k < j$ ) on the overlapping time horizon is used, which is defined as

$$\Gamma_{k,j} := \begin{bmatrix} \max_t |s_{i,k}(t) - s_{i,j}(t)| \\ \max_t |\dot{s}_{i,k}(t) - \dot{s}_{i,j}(t)| \\ \max_t |\ddot{s}_{i,k}(t) - \ddot{s}_{i,j}(t)| \end{bmatrix}, \quad t \in [t_{j,i}, t_{i,k} + T_i], \quad (3.1)$$

From this definition, two criteria can be derived. The first is the maximum difference between two consecutive trajectories  $\Gamma_{k,k+1}$ . The second is the maximum difference over all trajectories that share a part of their horizon,  $\max_j \Gamma_{k,j}$ ,  $j \in \{k+1, \dots, k + \lfloor T_i/t_p \rfloor\}$ . These performance indicators are used to study temporal consistency of the planner.

### 3.1.2 String Stability and Collision Avoidance

A clear requirement on the generated trajectories is safety. Hence, it is required that the cooperative trajectories are collision free. This is achieved by requiring that  $s_i(t) \leq s_{i-1}(t)$ , where vehicle lengths are assumed to be equal to zero without loss of generality. It should be noted that the trajectory might still have collisions with other vehicles, as other vehicles might decide to cut in. Hence, the collision check described in Section 2.1.4 should still be executed. However,

collision with the vehicle directly in front of the host vehicle is avoided if the curvilinear distance is smaller than that of the preceding vehicle.

The concept of string stability is essential for the behavior of the platoon [29] [28] [30]. String stability implies the attenuation of disturbances in the upstream direction of vehicles. Aside from preventing so called ‘ghost-traffic jams’, this property prevents unbounded growth of the acceleration inputs beyond the accelerations constraints of the vehicle. Therefore, string stability is a necessary condition to ensure feasibility of trajectories of vehicles in the upstream direction. The notion of string stability is described in [30] and the references contained therein, where  $\mathcal{L}_p$  string stability requires

$$\frac{\|\mathbf{y}_i(t)\|_{\mathcal{L}_p}}{\|\mathbf{y}_{i-1}(t)\|_{\mathcal{L}_p}} \leq 1, \quad \forall i, \quad (3.2)$$

$$\|\mathbf{y}_i(t)\|_{\mathcal{L}_p} = \sqrt[p]{\int_{-\infty}^{\infty} \|\mathbf{y}_i\|_p^p(t) dt}, \quad (3.3)$$

in which  $\mathbf{y}_i(t)$  is an output of the cascaded system under consideration. In this case, the acceleration of these vehicles is used as the output, because of the physical significance.

### 3.1.3 Spacing Policy

An important consideration for string stability is the spacing policy that is used. Many different spacing policies can be conceived. For example, [123] proposes a constant-distance spacing policy and derives that lead vehicle information is required to achieve string-stability. A spacing policy that is frequently used in literature [124] [27], is the constant time gap spacing policy. For this spacing policy, information of the lead vehicle is not required to achieve string stability. The desired constant time gap, the actual inter-vehicle distance, and the corresponding spacing error are given by

$$d_{r,i}(t) = c_i + h_i \dot{s}_i(t), \quad (3.4)$$

$$d_i(t) = s_{i-1}(t) - (s_i(t) + L_i), \quad (3.5)$$

$$e_i(t) = d_i(t) - d_{r,i}(t) = s_{i-1}(t) - (s_i(t) + L_i) - c_i - h_i \dot{s}_i(t), \quad (3.6)$$

respectively, where  $L_i$ ,  $h_i$  and  $c_i$  are the vehicle length, time gap and standstill distance of vehicle  $i$ , respectively. For this spacing policy, string stability can be obtained by using information of the preceding vehicle only.

For the remainder of this thesis, a predecessor-following configuration is used, where the host vehicle only uses information of the preceding vehicle. This makes the ad-hoc forming of platoons easier. Moreover, in [125], a multi-vehicle look-ahead was shown not to always be beneficial. A more general communication

topology could potentially be integrated but is outside the scope of the work presented here.

The aim is to integrate string-stable vehicle-following in trajectory planning. To this extent, a string of vehicles indexed by  $i \in \{0, 1, 2, \dots, N\}$  is considered, with the leading vehicle having index  $i = 0$ . The trajectory of vehicle  $i$  is denoted as  $\mathbf{S}_i(t)$ . The objective is now to plan a trajectory  $\mathbf{S}_i(t)$ ,  $t \in [t_{i,k}, t_{i,k} + T]$ , using communicated information,  $\mathbf{S}_{i-1}(t)$ ,  $t \in [t_{i,k}, t_{i,k} + T]$ , of the preceding vehicle.

## 3.2 Cooperative trajectory generation using polynomials

Given the trajectory of a preceding vehicle, a string-stable cooperative trajectory (that satisfies (3.2)) should be constructed, which is a trajectory that achieves a desired spacing policy, while satisfying (3.2). In [21], a simple PD controller in combination with a predicted trajectory of the preceding vehicle was proposed, essentially resulting in adaptive cruise control (ACC). While ACC can be made string-stable for sufficiently large inter vehicle distances [126], the trajectories created by such an approach cannot be communicated efficiently between vehicles, as they need to be sampled. Alternatively, the adopted framework of [18], which makes use of polynomial trajectories, also proposes a method to select a suitable terminal condition for vehicle following. It makes use of a slightly different spacing policy than the one described in (3.4). Instead of making use of the host vehicle velocity, it uses the velocity of the preceding vehicle to create additional spacing as

$$s_{\text{ref},i}(t) = s_{i-1}(t) - (c_i + h_i \dot{s}_{i-1}(t)). \quad (3.7)$$

This is done as it allows for a simple closed form equation to find the polynomial that connects the current state  $\mathbf{S}_i(0) = [s_i(t_0), \dot{s}_i(t_0), \ddot{s}_i(t_0)]$  to terminal state  $\mathbf{S}_i(\tau_i) = [s_{\text{ref},i}(\tau_i), \dot{s}_{i-1}(\tau_i), \ddot{s}_{i-1}(\tau_i)]$ , for a given arrival time  $\tau_i$ . Note that arrival time  $\tau_i$  and time horizon  $T_i$  are not necessarily identical. The arrival time  $\tau_i$  describes the end of the transient of the polynomial trajectory, while  $T_i$  describes the time horizon of the trajectory. The differentiation between these two values allows for a trajectory to be numerically evaluated over  $t \in [t_{i,k}, t_{i,k} + T_i]$  with sufficient temporal resolution, while the polynomial that describes the trajectory connects a state at time  $t_{i,k}$  with a state at time  $\tau_i$ . Trajectories where  $\tau_i > T_i$  are truncated, while trajectories where  $\tau_i < T_i$  are extrapolated with constant acceleration assumption.

When simulating the approach that indeed uses polynomials to connect current state  $\mathbf{S}_i(0) = [s_i(t_0), \dot{s}_i(t_0), \ddot{s}_i(t_0)]$  to terminal state  $\mathbf{S}_i(\tau_i) = [s_{\text{ref},i}(\tau_i), \dot{s}_{i-1}(\tau_i), \ddot{s}_{i-1}(\tau_i)]$  for a given  $\tau_i = \tau_{i-1} = T_i, \forall i$ , the response can be shown to be temporally consistent, and hence suitable for implementation on a single vehicle. However, it is also string-unstable. This can



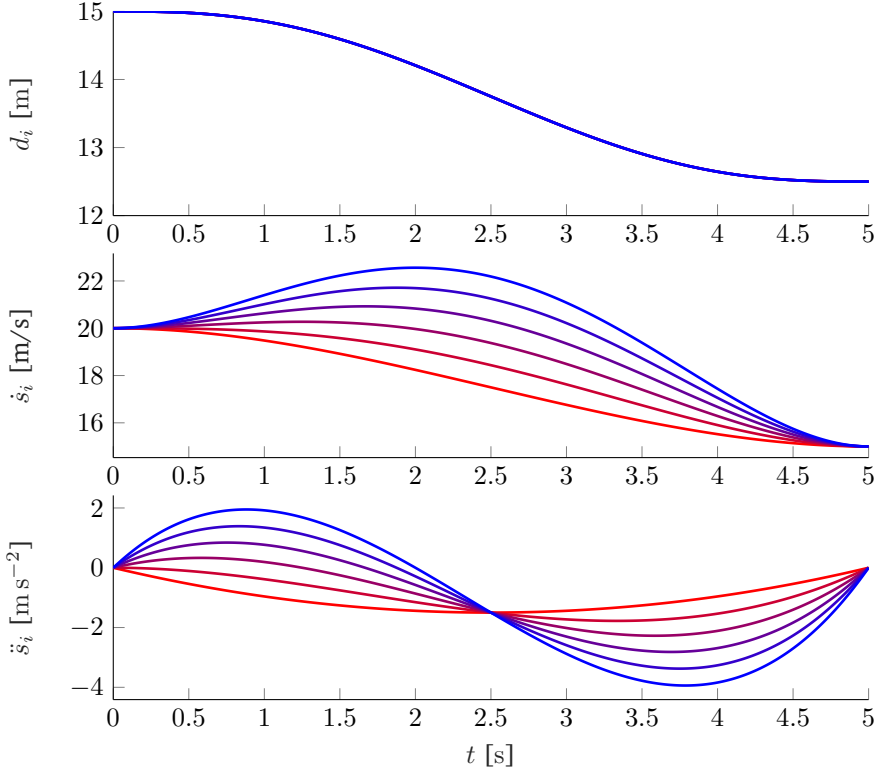


Figure 3.2: Cooperative vehicle following with approach presented in [18], which connects current vehicle state  $[s_i(t_0), \dot{s}_i(t_0), \ddot{s}_i(t_0)]$ , to reference state  $[s_{\text{ref},i}(\tau_i), \dot{s}_{i-1}(\tau_i), \ddot{s}_{i-1}(\tau_i)]$  according to (3.7), by means of polynomials.  $h_i = 0.5$  s,  $c_i = 5$  m. Lead vehicle (—), Last vehicle (—).

be seen in Figure 3.2, which illustrates a platoon of six vehicles that constructs cooperative trajectories by means of polynomials. All vehicles 'arrive' at the same velocity, at the same time  $\tau_i = 5$  s,  $\forall i$ , and the spacing policy is characterised by parameters  $h_i = 0.5$  s and  $c_i = 5$  m. Additionally, all vehicles start with an initial velocity of  $\dot{s}_i(0) = 20 \text{ m s}^{-1}$ ,  $\forall i$ , and transition to a velocity of  $\dot{s}_i(5) = 15 \text{ m s}^{-1}$ ,  $\forall i$ . However, the following vehicles need to cover an increasing larger distance, due to the spacing policy (3.7). To still arrive at the same terminal velocity, which is also specified in (3.7), it requires the vehicle to first accelerate before decelerating. This effect amplifies in the upstream direction. Hence, this method of vehicle following is string unstable.

### 3.2.1 Shift in arrival time <sup>1</sup>

A potential solution to the problem above can be conceived when recognising that the vehicles do not need to complete the transient at the same time. By shifting the arrival time  $\tau_i$  of vehicle  $i$  with respect to that of the preceding vehicle  $\tau_{i-1}$  string stability can potentially be obtained. To find the amount of time shift that is required, the monotonicity of the transient is analyzed, as preventing overshoot implies string stability. Assuming the platoon starts in steady state driving with a velocity of  $\dot{s}_i = v_0, \forall i$ . Then for the  $i$ -th vehicle, the initial condition can be written as

$$\mathbf{S}_i(0) = \begin{bmatrix} s_i(0) \\ \dot{s}_i(0) \\ \ddot{s}_i(0) \end{bmatrix} = \begin{bmatrix} -\sum_{j=1}^i h_j v_0 \\ v_0 \\ 0 \end{bmatrix}, \quad (3.8)$$

where without loss of generality the lead vehicle starts from  $s_0(0) = 0$ . Let  $\delta_{\tau,i}$  denote the shift of arrival time of vehicle  $i$  in the string, then  $\tau_i = \tau_0 + \sum_{j=1}^i \delta_{\tau,j}$ , where arrival time is equal to the time horizon of the lead vehicle  $\tau_0 = T_0$ . The trajectory traversed by the lead vehicle making a minimum jerk velocity transition from  $\dot{s}_0(0) = v_0$  to  $\dot{s}_0(\tau_0) = v_f$ , is described using a quartic polynomial as discussed in Section 2.1.4. For the lead vehicle, using the quartic polynomial in  $s(t)$  with zero initial and terminal acceleration, the position at the end of the transient is given by the polynomial, after which it continues with constant velocity, such that

$$s_0(\tau_0) = \tau_0 \left( v_0 - \frac{1}{2} [v_0 - v_f] \right), \quad (3.9)$$

$$s_0(\tau_i) = s_0(\tau_0) + \sum_{j=1}^i (\delta_{\tau,j} v_f). \quad (3.10)$$

Hence, the terminal position of vehicle  $i$  at time  $\tau_i$ , in a platoon that satisfies (3.7) is given by

$$s_i(\tau_i) = s_0(\tau_i) - \sum_{j=1}^i h_j v_f = s_0(\tau_0) + v_f \sum_{j=1}^i (\delta_{\tau,j} - h_j), \quad (3.11)$$

where  $s_0(\tau_0)$  denotes the position of the lead vehicle at the end of its arrival time  $\tau_0$ . Then terminal state is given by

$$\mathbf{S}_0(\tau_i) = \begin{bmatrix} \tau_0 \frac{1}{2} (v_0 + v_f) + v_f \sum_{j=1}^i (\delta_{\tau,j} - h_j) \\ v_f \\ 0 \end{bmatrix}. \quad (3.12)$$

---

<sup>1</sup>This section is based on [127].

Then (2.18) is used to construct quintic polynomial trajectories  $s_i(t)$  between initial state (3.8) and terminal state (3.12) via

$$\mathbf{S}_i(t) = \mathbf{M}_1(t)\mathbf{c}_{012_i} + \mathbf{M}_2(t)\mathbf{c}_{345_i}, \quad t \in [0, \tau_i], \quad (3.13)$$

$$\mathbf{M}_1(t) = \begin{bmatrix} 1 & t & t^2 \\ 0 & 1 & 2t \\ 0 & 0 & 2 \end{bmatrix}, \quad \mathbf{M}_2(t) = \begin{bmatrix} t^3 & t^4 & t^5 \\ 3t^2 & 4t^3 & 5t^4 \\ 6t & 12t^2 & 20t^3 \end{bmatrix},$$

where  $\mathbf{c}_{012_i}$  and  $\mathbf{c}_{345_i}$  are the coefficients of the polynomials of vehicle  $i$ , and are found by substituting (3.8) and (3.12)

$$\mathbf{c}_{012_i} = \mathbf{M}_1(0)^{-1}\mathbf{S}_i(0), \quad (3.14)$$

$$\mathbf{c}_{345_i} = \mathbf{M}_2(\tau_i)^{-1} [\mathbf{S}_i(\tau_i) - \mathbf{M}_1(\tau_i)\mathbf{c}_{012_i}], \quad (3.15)$$

which results in

$$\mathbf{c}_{012_i} = \begin{bmatrix} -\sum_{j=1}^i h_j v_0 \\ v_0 \\ 0 \end{bmatrix}, \quad \mathbf{c}_{345_i} = (v_0 - v_f) \begin{bmatrix} \frac{10 \sum_{j=1}^i h_j - 6 \sum_{j=1}^i \delta_{\tau,j} - \tau_0}{\tau_i^3} \\ \frac{\tau_0 - 30 \sum_{j=1}^i h_j + 16 \sum_{j=1}^i \delta_{\tau,j}}{2\tau_i^4} \\ \frac{6 \sum_{j=1}^i h_j - 3 \sum_{j=1}^i \delta_{\tau,j}}{\tau_i^5} \end{bmatrix}.$$

with  $\tau_i = \tau_0 + \sum_{j=1}^i \delta_{\tau,j}$ . Note that if the polynomials are monotonic, no overshoot can occur, provided that the terminal velocity is equal to that of the preceding vehicle. To check monotonicity of  $\dot{s}_i(t)$ ,  $t \in [0, \tau_i]$ , the position of the three roots of  $\ddot{s}_i(t)$  are analyzed. Two of these roots are known a-priori, when the platoon is assumed to start and end in steady state with  $\ddot{s}_i(0) = \ddot{s}_i(\tau_i) = 0$ , such that roots are found at  $t = 0$  and  $t = \tau_i$ . To ensure monotonicity on the interval  $t \in [0, \tau_i]$ , the third root is required to lie in  $t \notin (0, \tau_i)$  and can be written as

$$t_{\ddot{s}=0} = -\frac{\left(\tau_0 + \sum_{j=1}^i \delta_{\tau,j}\right) \left(\tau_0 - 10 \sum_{j=1}^j h_j + 6 \sum_{j=1}^i \delta_{\tau,j}\right)}{10 \sum_{j=1}^i (2h_j - \delta_{\tau,j})}. \quad (3.16)$$

This third root can be made to coincide with respectively the root at  $t = 0$ , and coincide with  $t = \tau_i$  for timeshifts  $\delta_{\tau,i}$  as

$$\delta_{\tau_0,i} = \frac{5}{3} \sum_{j=1}^i h_j - \frac{1}{6} \tau_0 - \sum_{j=1}^{i-1} \delta_{\tau,j}, \quad \delta_{\tau_{\tau_i},i} = \frac{\tau_0}{4} + \frac{5}{2} \sum_{j=1}^i h_j - \sum_{j=1}^{i-1} \delta_{\tau,j},$$

where,  $\delta_{\tau_0,i}$  and  $\delta_{\tau_{\tau_i},i}$  denote the values for  $\delta_{\tau,i}$  for which the root coincides with  $t = 0$  and  $t = \tau_i$  respectively. Clearly,  $\delta_{\tau_0,i} < \delta_{\tau_{\tau_i},i}$ , hence  $\delta_{\tau_0,i}$  thus results in the

smallest timeshift for the arrival time. As such, it results in quicker responses for the vehicles. Note that in order for the timeshift to be positive, it is required that provided that  $\tau_0 + 6 \sum_{j=1}^{i-1} \delta_{\tau,j} < 10 \sum_{j=1}^i h_j$ . Ideally, this shift in time does not depend on the position in the string. For a string of vehicles in which the time gaps (i.e.,  $h_i = h_{i-1}$ ,  $\forall i$ ) and timeshifts are homogeneous (i.e.,  $h_i = h_{i-1}$ ,  $\forall i$  and  $\delta_{\tau,i} = \delta_{\tau,i-1}$ ,  $\forall i$ ) the following can be written

$$\delta_{\tau} = \frac{5}{3}h - \frac{1}{6i}\tau_0 \quad (3.17)$$

which still depends on vehicle index  $i$ . To remove this dependency, the term  $-\frac{\tau_0}{6i}$  is dropped to obtain

$$\delta_{\tau} = \frac{5}{3}h, \quad (3.18)$$

for which it can be verified that the third root is equal to  $t = -\tau_0 \frac{3\tau_0 + 5ih}{10ih}$ , which is smaller than 0 for all parameters, and thus monotonicity is ensured with the selected timeshift of (3.18). Hence, by communicating both the time shift  $\tau_i$  as well as the desired state  $\mathbf{S}_i(\tau_{c,i})$  in the upstream direction, a monotonic transient can be realised while planning with quintic polynomial trajectories. Then to find the polynomial coefficients, the following terminal state is used

$$\mathbf{S}_i(\tau_i) = \begin{bmatrix} s_{i-1}(\tau_i) - c - L_i - h\dot{s}_{i-1}(\tau_i) \\ \dot{s}_{i-1}(\tau_i) \\ \ddot{s}_{i-1}(\tau_i) \end{bmatrix}, \quad \tau_i = \tau_{i-1} + \frac{5}{3}h, \quad (3.19)$$

to construct trajectories  $\mathbf{S}_i(0)$ ,  $t \in [0, \tau_i]$ . The results of this approach can be seen in Figure 3.3, which illustrates a string stable transition for the platoon from  $v_0 = 20 \text{ m s}^{-1}$  to  $v_f = 15 \text{ m s}^{-1}$ , with spacing policy parameters  $h_i = 0.5 \text{ s}$  and  $c_i = 5 \text{ m}$ .

In practice, the planner is executed in a receding horizon fashion, such that the planning is updated after  $t_p$ . At each planning time  $t_{i,k} = t_{k-1,i} + t_p$  a new trajectory is planned over  $t \in [t_{i,k}, t_{i,k} + T_i]$ , with  $T_i$  the planning horizon. The result of this planner is demonstrated in Figure 3.4a for  $t_p = 0.2 \text{ s}$ , which illustrates the planned trajectories of each vehicle in lighter colors, as well as the executed trajectories for each vehicle in color. In this scenario  $T_i = 5 \text{ s}$ ,  $\forall i$  and  $\tau_i = T_0 + i\frac{5}{3}h, \forall i$  and  $t_p = 0.2 \text{ s}$ . It can be seen that again string stable behavior is achieved for this scenario.

A problem that occurs however with this approach becomes apparent with long platoons and initial perturbations. Each trajectory is constrained to be at the new steady state scenario at time  $\tau_i$ , where the initial perturbations or disturbances are 'dissipated'. However, due to the timeshift  $\delta_{\tau}$  the dissipation of this initial perturbation is spread out over the time interval  $t \in [t_{i,k}, t_{i,k} + \tau_i]$ , with  $\tau_i = \tau_{i-1} + \delta_{\tau,i} = \tau_{i-1} + \delta_{\tau}$ . As a result vehicles further upstream are slower

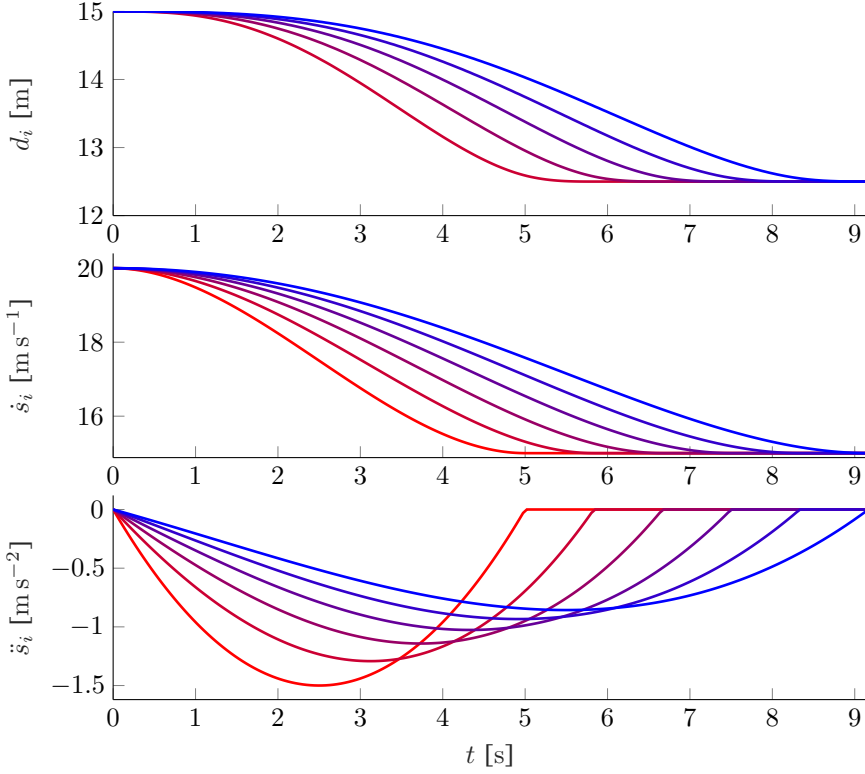
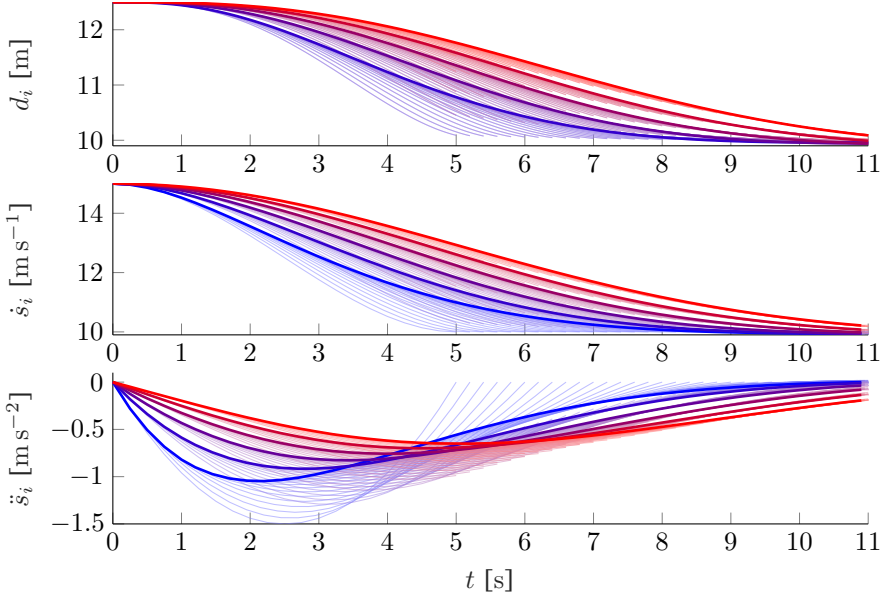


Figure 3.3: Single plan step, connecting connecting  $\mathbf{S}_i(0)$  of (3.8), to  $\mathbf{S}_i(\tau_i)$  from (3.19), by quintic polynomials  $t \in [t_{i,k}, t_{i,k} + T_i]$ .  $h_i = 0.5$  s,  $c_i = 5$  m. Lead vehicle (—), Last vehicle (—).

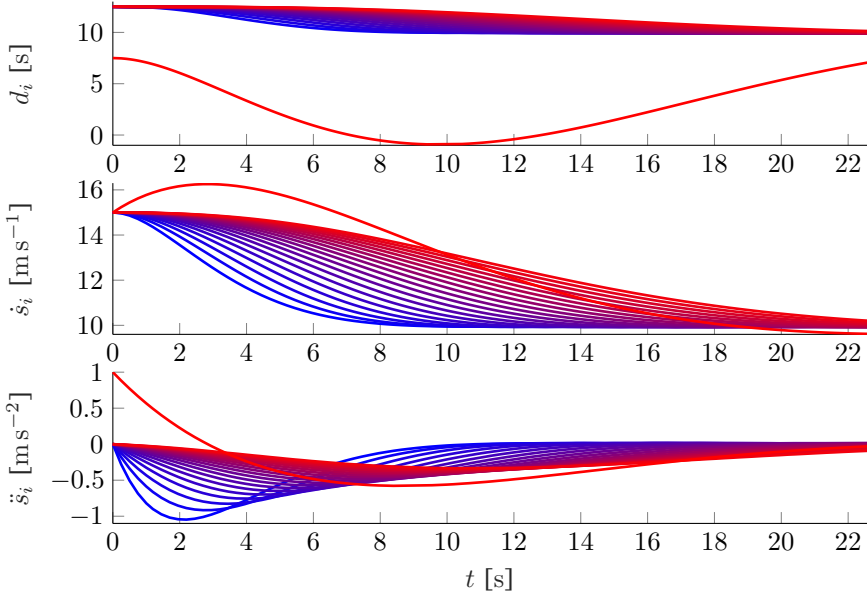
to react to these disturbances. This problem is illustrated in Figure 3.4b, which illustrates a collision for the final vehicle which is perturbed with respect to the initial condition  $\mathbf{S}_i(0)$  of (3.8) as

$$\mathbf{S}_i(0) = \begin{bmatrix} s_i(0) \\ \dot{s}_i(0) \\ \ddot{s}_i(0) \end{bmatrix} = \begin{bmatrix} -\sum_{j=1}^i h_j v_0 \\ v_0 \\ 0 \end{bmatrix} + \begin{cases} [0 \ 0 \ 0]^\top, & i \neq N \\ [c \ 0 \ 1]^\top, & i = N \end{cases}. \quad (3.20)$$

As the inter-vehicle distance is smaller than zero (i.e.,  $d_i(t) < 0$ ) a collision occurs, which is unacceptable for the planning algorithm.



(a) connecting  $\mathbf{S}_i(0)$  of (3.8), to  $\mathbf{S}_i(\tau_i)$  from (3.19). Planned trajectories of lead vehicle (—) and last vehicle (—).



(b) connecting perturbed  $\mathbf{S}_i(0)$  of (3.20), to  $\mathbf{S}_i(\tau_i)$  from (3.19).

Figure 3.4: Recursive planning ( $T_i = 5$  s,  $\forall i$ ,  $\tau_0 = T_0$ ,  $t_p = 0.2$  s), by quintic polynomials  $t \in [t_{i,k}, t_{i,k} + T_i]$ , for cooperative vehicle following  $h_i = 0.5$  s,  $c_i = 5$  m. Executed trajectories by Lead vehicle (—) and Last vehicle (—)

### 3.3 Cooperative B-spline Trajectory Generation <sup>2</sup>

A different approach to cooperative planning can be made by not making the time shift described in Section 3.2, but instead fitting the planned trajectory to the desired spacing policy of (3.4). To gain more freedom in fitting the trajectory, B-splines, as described in Section 2.2.2, are used. The objective is to describe state trajectories  $\mathbf{S}_{i,k}(t)$  for plan step  $k$ , using these B-splines. Therefore, the time  $t$  is taken as the parametrisation variable instead of  $u$ . Then a one dimensional B-spline is used to describe a trajectory  $s_{i,k}(t)$  at plan step  $k$  for vehicle  $i$ :

$$s_{i,k}(t) = \sum_{j=0}^n N_{j,p}(t - t_{i,k}) P_{j,i,k} = \mathcal{B}_{n,p}(t - t_{i,k}) \mathcal{P}_{i,k}, \quad t \in [t_{i,k}, t_{i,k} + T_i], \quad (3.21)$$

where  $\mathcal{P}_{i,k} = [P_{0,i,k}, P_{1,i,k}, \dots, P_{n,i,k}]^\top$  are  $n + 1$  control points. The knots in the knot vector  $\mathbf{U}_i$  are also expressed in the time domain. The trajectory should be defined over the desired time horizon  $T_i$ , therefore the first knot is chosen as 0, while the final knot is chosen as  $T_i$ . Basis functions should then be evaluated relative to the current time  $t_{i,k}$  (hence the argument  $t - t_{i,k}$  of the basis functions  $N_{j,p}(t - t_{i,k})$ ). A uniform clamped knot vector is used,

$$\begin{aligned} \mathbf{U}_i &= [ \underbrace{0, \dots, 0}_{p+1}, \underbrace{u_{p+1}, \dots, u_n}_{n-p}, \underbrace{T_i, \dots, T_i}_{p+1} ], \\ u_j &= \frac{j - p}{n - p + 1} T_i, \quad j = \{p + 1, \dots, n\}. \end{aligned} \quad (3.22)$$

Note that once the trajectory for  $s_{i,k}(t)$  is defined, the hodograph can simply be computed to find  $\dot{s}_{i,k}(t)$  and  $\ddot{s}_{i,k}(t)$ . This means  $\mathbf{S}_{i,k}(t)$  can be evaluated using only control points  $\mathcal{P}_{i,k}$ , current planning time  $t_{i,k}$  and time horizon  $T_i$ . This is illustrated in Figure 3.5, which shows the B-spline for  $s_{i,k}(t)$  and corresponding spline trajectories for  $\dot{s}_{i,k}(t)$  and  $\ddot{s}_{i,k}(t)$ .

#### 3.3.1 B-spline Trajectory Construction

To achieve the spacing policy (3.4), information of the preceding vehicle is used. It is assumed that information of the planned trajectory of the preceding vehicle,  $s_{i-1,k}(t)$ , is available in the form of a B-spline. The  $n + 1$  control points of this trajectory are denoted by  $\mathcal{P}_{i-1,k}$ . In addition to the control points, the time instance,  $t_{k,i-1}$ , at which the preceding vehicle has planned its trajectory, as well as the time horizon,  $T_{i-1}$ , are received such that the clamped uniform knot vector as in (3.22) can be reconstructed, assuming that all vehicles use a uniformly distributed knot vector, with the same degree  $p$  of the B-spline. Hence,

<sup>2</sup>This section is based on [122].

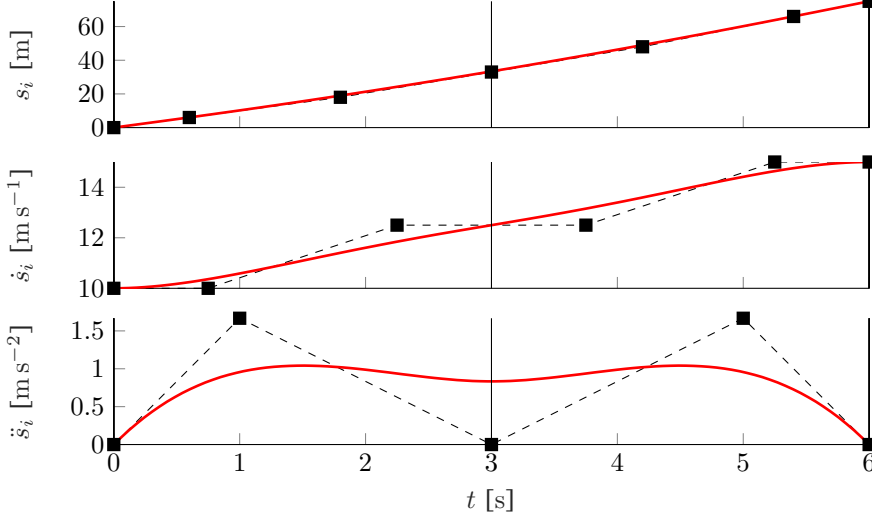


Figure 3.5: B-spline trajectory ( $n = 7, p = 5$ )  $s_i(t)$  (—), control points ( $\mu_j, P_{j,i}$ ) (■), control polygon (---) and knots (|).

the complete trajectory of the preceding vehicle,  $S_{i-1,k}(t)$  can be evaluated, by communicating only  $n + 3$  values. In contrast, methods using Model Predictive Control or sampled trajectories, assuming a control frequency  $f_c$ , would require instead to communicate  $f_c \cdot T_i$  values to achieve the same. Typical values are  $f_c = 100$  Hz,  $T_i = 5$  s and  $n = 6$ , resulting in 500 values that need to be communicated in the case of MPC and only 9 values for the proposed planning method.

Next, to construct the desired trajectory for the host vehicle  $i$ , spacing policy (3.4) is used. In traditional CACC applications, a feedback controller is used to regulate the error,  $e_i$ , towards zero. Along the same line of thought, the B-spline is manipulated to ensure the planned trajectories minimize this error as well. For a given time, the value of the basis functions for a given order can be computed and combined into a row vector

$$\mathcal{B}_{n,p}(t) = [N_{0,p}(t) \quad N_{1,p}(t) \quad \dots \quad N_{n,p}(t)], \quad (3.23)$$

such that (3.21) can be rewritten to

$$s_{i,k}(t) = \mathcal{B}_{n,p}(t - t_{i,k}) \mathcal{P}_{i,k}, \quad (3.24)$$

$$\dot{s}_{i,k}(t) = \mathcal{B}_{n-1,p-1}(t - t_{i,k}) \Delta_{10} \mathcal{P}_{i,k}, \quad (3.25)$$

$$\ddot{s}_{i,k}(t) = \mathcal{B}_{n-2,p-2}(t - t_{i,k}) \Delta_{20} \mathcal{P}_{i,k}. \quad (3.26)$$

Substituting (3.24) and (3.25) in (3.6) yields

$$e_{i,k}(t) = s_{i-1,k}(t) - c_i + L_i - [\mathcal{B}_{n,p}(t) + h_i \mathcal{B}_{n-1,p-1}(t) \Delta_{10}] \mathcal{P}_{i,k}. \quad (3.27)$$



Time instances

$$t_j^* \in [t_{i,k}, t_{i,k} + T_i], \quad j \in \{1, 2, \dots\} \quad (3.28)$$

are used to construct a vector  $\mathcal{T}$  for which it holds that  $t_j^* < t_{j+1}^*$ . For each element in this vector the error in (3.27) is computed and the objective is to minimize the cumulative error

$$J = \sum_{t_j^* \in \mathcal{T}} e_{i,k}(t_j^*)^2. \quad (3.29)$$

To solve this minimization problem, (3.27) can be set equal to zero for each time instance in  $\mathcal{T}$ , to construct a linear system of equations which can be solved for  $\mathcal{P}_{i,k}$ . Note that in order for the solution to be unique,  $t^*$  should satisfy the Schoenberg-Whitney's Condition [128] [114],

$$u_j \leq t_j^* \leq u_{j+p}. \quad (3.30)$$

Thus, the selection of the time instances in  $\mathcal{T}$  is important. Since the error of (3.27) should be small over the entire time horizon, samples over the entire horizon should be included. Note that the coordinates of the control points of (2.55), satisfy both the Schoenberg-Whitney's condition, and span the entire spline. Therefore,  $t_j^* = \mu_j + t_{i,k}$  is used in the vector of time instances  $\mathcal{T}$ .

Planned trajectories start in the current vehicle state as described earlier. By construction,  $N_{0,p}(0) = 1, \forall p$  and  $N_{k,p}(0) = 0, k > 0$ . The control points  $P_{j,i,k}, j \in \{0, 1, 2\}$  are fixed to correspond to the initial conditions (3.24), (3.25), and (3.26) and can be computed via

$$\begin{bmatrix} P_{0,i,k} \\ P_{1,i,k} \\ P_{2,i,k} \end{bmatrix} = \mathbf{C}^{-1} \mathbf{S}_i(t_{i,k}), \quad (3.31)$$

$$[\mathbf{C} \quad \mathbf{0}_{3 \times (n-2)}] = \left[ \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \end{bmatrix} \Delta_{10} \right],$$

with  $\mathbf{C} \in \mathbb{R}^{3 \times 3}$ . These first three control points largely control the initial response. Therefore, only the time instances  $\mathcal{T} = [\mu_3, \mu_4, \dots, \mu_n]$  are used, as the coordinates of the free control points, such that the cost function  $J = \sum_{j=3}^n e_{i,k}(\mu_j + t_{i,k})^2$  is equated to zero, which has a unique solution. The

control points can be found by solving a system of equations of (3.27)

$$\begin{bmatrix} P_{3,i,k} \\ \vdots \\ P_{n,i,k} \end{bmatrix} = \mathbf{\Omega}_{12}^{-1} \left( \begin{bmatrix} s_{i-1}(\mu_3 + t_{i,k}) - c_i - L_i \\ \vdots \\ s_{i-1}(\mu_n + t_{i,k}) - c_i - L_i \end{bmatrix} - \mathbf{\Omega}_{11} \begin{bmatrix} P_{0,i,k} \\ P_{1,i,k} \\ P_{2,i,k} \end{bmatrix} \right), \quad (3.32)$$

$$\begin{bmatrix} \mathbf{\Omega}_{11} & \mathbf{\Omega}_{12} \end{bmatrix} = \begin{bmatrix} \mathcal{B}_{n,p}(\mu_3) + h_i \mathcal{B}_{n-1,p-1}(\mu_3) \mathbf{\Delta}_{10} \\ \vdots \\ \mathcal{B}_{n,p}(\mu_n) + h_i \mathcal{B}_{n-1,p-1}(\mu_n) \mathbf{\Delta}_{10} \end{bmatrix},$$

where  $\mathbf{\Omega}_{11}$ ,  $\mathbf{\Omega}_{12}$  are matrix partitions of corresponding sizes. More timestamps could be included in the optimization, in which case an optimization problem arises which can be easily solved by using the Moore-Penrose pseudo inverse of  $\mathbf{\Omega}_{12}$ . The selection for  $t^*$  includes points on the B-spline over the entire horizon, therefore overshoot is likely to be prevented for the application. This makes it suitable to obtain string stability.

### 3.3.2 Design Considerations

The planner is implemented in a receding horizon fashion that updates at fixed time intervals  $t_p$ , such that  $t_{i,k+1} = t_{i,k} + t_p$ . The design objective of temporal consistency can be achieved by means of a minimization criterion that includes samples over the entire planning horizon, due to the principle of optimality. Ideally, this minimization result remains identical when the time horizon is extended. Note that (3.6) indeed satisfies this criteria.

Additionally, exact temporal consistency can only be realised if the basis functions are identical. This is generally not the case, unless the new knot vector is shifted backwards over  $t_p$  to coincide with the knot vector of the previous plan step. However, this would require additional overhead in keeping track of how the basis functions should be defined. Hence, exact temporal consistency is sacrificed in favour of simple implementation.

An important consideration is the degree  $p$  of the B-spline, and the number of control points  $n + 1$  that is used. The smoothness properties of the trajectories used [127] are retained for comfort reasons, by choosing  $p = 5$ . For  $n$ , note that for a given horizon  $T_i$ , using more control points allows the method to regulate the error faster at the expense of larger control inputs and a higher computational burden on the CPU.

### 3.3.3 Communication / Planning Delay

Traditionally, one of the major bottlenecks in CACC controller design has been the communication delay [129]. This delay impedes the performance of the feedback controller (unless compensated for by means of prediction [130]),

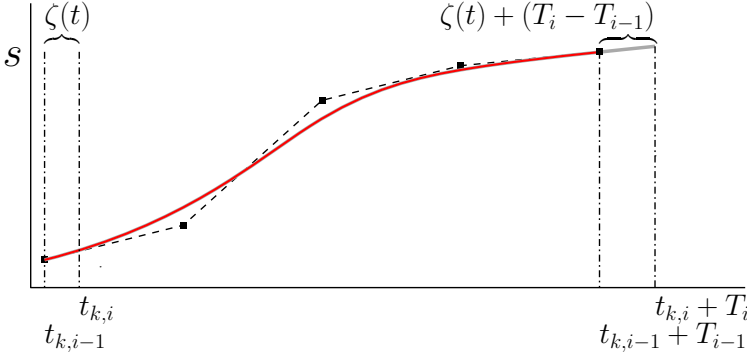


Figure 3.6: B-spline trajectory of predecessor is extended in case of communication delay  $\zeta(t)$ , as information is not included over the entire time horizon of the host vehicle,  $t \in [t_{i,k}, t_{i,k} + T_i]$

since information about the desired acceleration of the preceding vehicle is available to the host vehicle later. Using a different control strategy does not remove this delay as it is the result of the hardware that is used in the system. However, as is demonstrated later, the trajectory planning approach is more robust against a delay  $\zeta(t)$ . This delay does not only include the communication delay, but can also occur due to asynchronous planning, where a following vehicle does not plan at the same time instance as the preceding vehicle. In what follows,  $\zeta(t)$  denotes the planning delay, which is defined as the time between the time instance on which the preceding vehicle has planned its trajectory, and the time at which this information is used by the host vehicle to plan its trajectory. The robustness against the planning delay  $\zeta(t)$  is due to the communicated information not only containing the current desired acceleration, but also the desired acceleration over an entire time horizon. Nevertheless, it is still required to take communication delay into account, as the communicated trajectory now no longer contains the complete time horizon that the host vehicle needs to plan over. This is illustrated in Figure 3.6, where the communicated trajectory of the preceding vehicle can be seen. Note that the horizon of the host vehicle cannot simply be truncated to match that of the preceding vehicle, as this would result in vehicles upstream ending up with ever smaller time horizons. The same problem occurs if the planning horizon of the preceding vehicle is smaller than that of the host vehicle (i.e.,  $T_i > T_{i-1}$ ).

To ensure that the host vehicle is still capable of constructing a trajectory over its time horizon,  $t \in [t_{i,k}, t_{i,k} + T_i]$ , some options are available. One option would be to change the time vector  $t^*$ , which contains the time instances on which the error (3.27) is equated to zero (or minimized if more time instances are added). However, for large mismatches in time horizon  $T_i - T_{i-1}$ , this could result in a violation of the Schoenberg-Withney relation (3.30). The other option is to simply

extend the B-spline trajectory of the preceding vehicle by means of prediction. A constant acceleration is assumed for the extended part of the trajectory. The following notation is used to evaluate the vehicle states planned at plan step  $k$ , on the planning horizon, and beyond its horizon (i.e.,  $t > T_i$ ),

$$\mathbf{S}_{i,k}(t) = \begin{bmatrix} s_{i,k}(t) \\ \dot{s}_{i,k}(t) \\ \ddot{s}_{i,k}(t) \end{bmatrix} = \Upsilon_i(t - t_{i,k}) \mathcal{P}_{i,k}, \quad t_{i,k} \leq t, \quad (3.33)$$

where

$$\Upsilon_i(t) = \begin{cases} \begin{bmatrix} \mathcal{B}_{n,p}(t) \mathbf{I} \\ \mathcal{B}_{n-1,p-1}(t) \Delta_{10} \\ \mathcal{B}_{n-2,p-2}(t) \Delta_{20} \end{bmatrix}, & \text{if } t \leq T_i, \\ \begin{bmatrix} 1 & (t - T_i) & \frac{1}{2}(t - T_i)^2 \\ 0 & 1 & (t - T_i) \\ 0 & 0 & 1 \end{bmatrix} \Upsilon_i(T_i), & \text{if } t > T_i, \end{cases} \quad (3.34)$$

where if the time  $t$  satisfies  $t_{i,k} < t \leq t_{i,k} + T_i$ , the B-spline trajectory is simply evaluated by means of basis functions  $\mathcal{B}_{n,p}(t)$ , and when  $t > t_{i,k} + T_i$ , a constant acceleration is assumed for the part for which  $t > t_{i,k} + T_i$ .

### 3.3.4 Simulation Results

This section demonstrates the effectiveness of this algorithm for path planning. Therefore, various important scenarios for a platoon of vehicles are implemented. The lead vehicle uses B-spline trajectories, that transition to a new velocity with minimum jerk. Note that these correspond with the quartic polynomial trajectories from [18], [127]. This trajectory for the lead vehicle is represented by a B-spline with  $p = 5$ ,  $T_0 = 5$  s, knots according to (3.22) and  $n + 1 = 7$  control points  $\mathcal{P}_{0,0} = [0 \ 10 \ 30 \ 48.75 \ 65 \ 80 \ 87.5]$ . This lead vehicle trajectory corresponds to a velocity change from  $\dot{s}_{0,0}(0) = 20 \text{ m s}^{-1}$  to a velocity of  $\dot{s}_{0,0}(T_0) = 15 \text{ m s}^{-1}$ . Note that this trajectory updates every planning cycle, such that  $\mathcal{P}_{0,k}$  updates every plan step  $k$ , to correspond with a new initial condition. This is done in a similar manner as for all the following vehicles, except that for the lead vehicle, the desired trajectory is modified to match the trajectory planned in the previous planning cycle as closely as possible. Additionally, a terminal constraint of  $\dot{s}_{0,k}(t_{k,0} + T_0) = 0$  is applied to prevent oscillations in spline fitting. As such, the control points  $\mathcal{P}_{0,k}$  for the lead vehicle

are solved by means of a minimization with equality constraint

$$\begin{aligned} \min \quad & \left\| \begin{bmatrix} \mathcal{B}_{n,p}(\mu_3) \\ \vdots \\ \mathcal{B}_{n,p}(\mu_n) \end{bmatrix} \mathcal{P}_{0,k} - \begin{bmatrix} s_{0,k-1}(\mu_3 + t_{k,0}) \\ \vdots \\ s_{0,k-1}(\mu_n + t_{k,0}) \end{bmatrix} \right\|_2, \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{C} \\ [0 \ \dots \ 0 \ 1] \Delta_{20} \\ [0 \ \dots \ 0 \ 1] \Delta_{10} \end{bmatrix} \mathcal{P}_{0,k} = \begin{bmatrix} \mathbf{S}_0(t_{k,0}) \\ 0 \\ \dot{s}_{0,k-1}(t_{0,k} + T_0) \end{bmatrix}, \end{aligned} \quad (3.35)$$

where  $s_{0,k-1}(t)$  represents the values of plan step  $k - 1$  and where  $\mathbf{C}$  is identical to that in (3.31). Also note that similarly to the case of communication delay, the trajectory planned in the previous plan step needs to be extrapolated. This is done by assuming a constant velocity at the value to which the velocity transition is made.

Note that in Section 3.2.1, the performance of the polynomial planner of [18] was already demonstrated and shown to be string-unstable. A potential solution was also demonstrated in Section 3.2.1, however, this required to shift the time horizon for each vehicle in the string, which would not be practical for large platoons, due to the issues shown in Figure 3.4b.

### Temporal Consistency

One of the control objectives noted in Section 3.1 was that of temporal consistency. This refers to how similar consecutively planned trajectories are. As already stated, in general, exact temporal consistency cannot be achieved due to the basis functions being shifted over time  $t_p$ . However, in practice it suffices to require that consecutive trajectories are close to each other (i.e., requiring that  $\Gamma_{k,k+1}$  in (3.1) is ‘small’).

A simulation study was performed to analyze the performance indicators from Section 3.1.1 for a scenario in which the lead vehicle brakes from  $\dot{s}_{0,0}(0) = 20 \text{ m s}^{-1}$  to a velocity of  $\dot{s}_{0,0}(T_0) = 15 \text{ m s}^{-1}$ . As spacing policy parameters, a time gap of  $h_i = 0.5 \text{ s}$ , and standstill distance  $c_i = 5 \text{ m}$  is used for all vehicles; all vehicle lengths are set to zero, without loss of generality. For the B-spline trajectories, the same degree  $p$ , number of control points  $n + 1$  and time Horizon  $T_i$  are used as before. This simulation is depicted in Figure 3.7 for a planning update interval of  $t_p = 0.2 \text{ s}$ . The first vehicle is illustrated by a black line, and upstream vehicle responses are indicated by an increasingly blue color. For each vehicle, all planned trajectories are plotted over the entire horizon to show discrepancies between the consecutive planning cycles. The actual response of each vehicle, which is determined by the consecutive planned B-spline trajectories on time spans  $t \in [t_{i,k}, t_{i,k} + t_p]$ , updated at intervals  $t_p = 0.2 \text{ s}$ , is illustrated by the color corresponding to that vehicle.

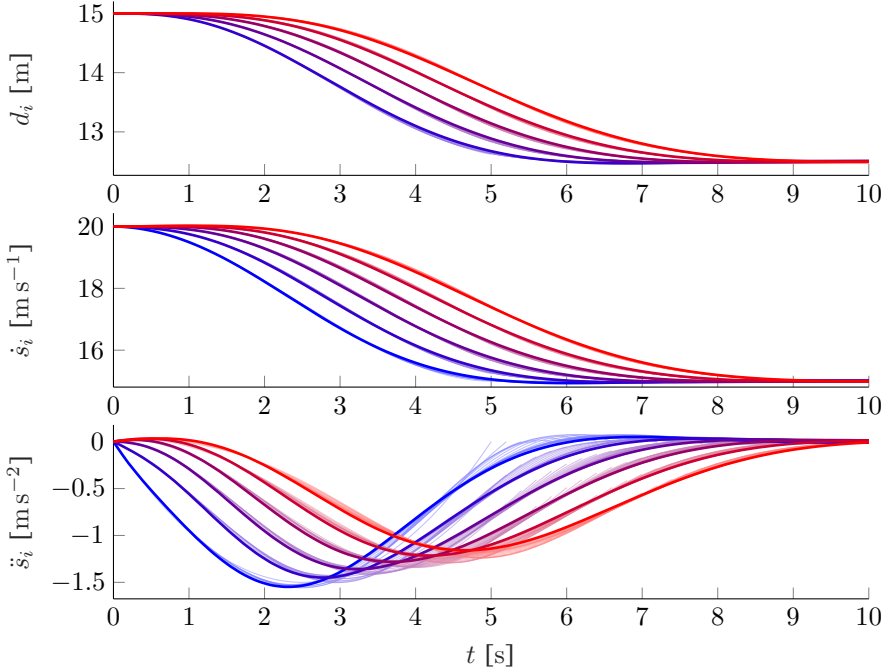


Figure 3.7: Results of B-spline based trajectories for a velocity change in a platoon with six vehicles,  $t_p = 0.2$  s, lead vehicle (—blue—), sixth vehicle (—red—); Planned trajectories of Lead vehicle (—light blue—) and Last vehicle (—light red—).

As can be seen from Figure 3.7, the response from all vehicles is a smooth transition to the new velocity. The string of vehicles can be seen to not demonstrate string-unstable behavior, due to the magnitude of acceleration decreasing over the vehicle index. Moreover, the consecutive planning cycles show (close too) temporal consistency. The new knots do not coincide with the previous knots since  $u_{j+1} - u_j \neq t_p$ . Therefore, the transitions of the lowest order basis functions also occur at slightly different times. Hence, exact temporal consistency is not possible. However, the figure shows that both position and velocity remain close to the original trajectory.

This is confirmed in Figure 3.8, which illustrates the aforementioned performance indicators for various values of planning update time  $t_p$ , for a platoon of 6 vehicles (figure illustrates the maximum over all vehicles). Clearly, for smaller update times, the difference between two consecutive planning cycles are smaller as can be seen from the black crosses. Additionally, since more trajectories span the same time instance, the largest difference between any of the planned trajectories is larger for smaller planning update times  $t_p$ . Note that the maximum difference in position between overlapping trajectories ( $\approx 0.37$  m)

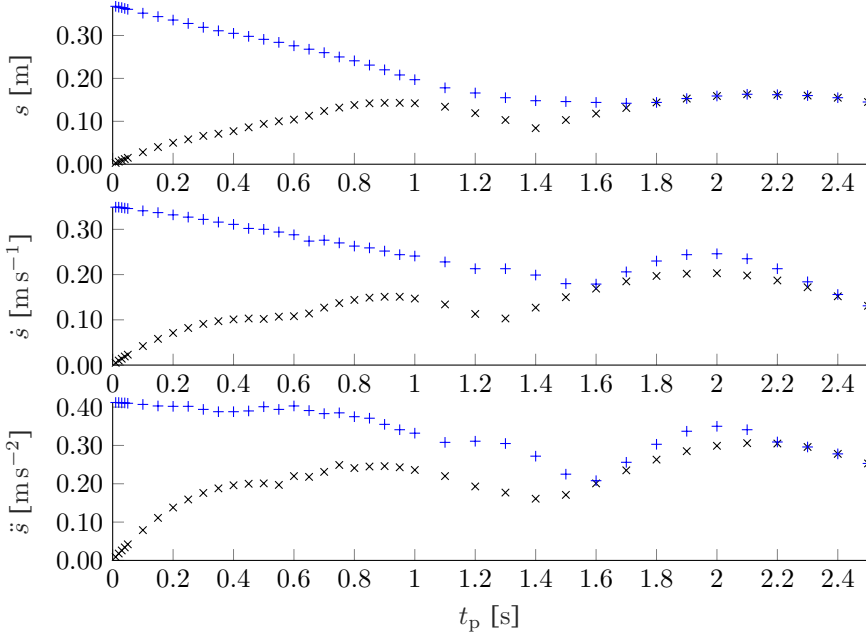


Figure 3.8: Sensitivity of  $t_p$  on temporal consistency for  $T_i = 5$  s. Maximum difference per element of  $\Gamma_{k,j}$  between consecutive trajectories  $\max_k \Gamma_{k,k+1}$  ( $\times$ ), and all overlapping trajectories  $\max_k (\max_j \Gamma_{k,j})$ ,  $j \in \{k+1, \dots, \lfloor k + T_i/t_p \rfloor\}$  ( $+$ )

for this manoeuvre is less than 2.5% ( $\approx 0.37/15$ ) of the inter-vehicle distance ( $= 15$  m) of the presented manoeuvre, which is sufficiently small to be acceptable.

It should be noted that the update time  $t_p$  does not only affect temporal consistency, but also affects the planning delay. The expected value of planning delay due to vehicles planning in an asynchronous fashion increases linearly with the value of  $t_p$ . Hence,  $t_p$  is best chosen as small as the on-board CPU allows in terms of computational capabilities. In the remainder of this section  $t_p = 0.2$  s is chosen.

### Initial Perturbation

A second important test scenario is where the lead vehicle is driving at a constant velocity and parameters are identical to that of the temporal consistency analysis, but the platoon is not in steady state, since the position of the second vehicle has an offset with respect to its desired position. The proposed method of using polynomials of Section 3.2.1 is not capable of handling these initial perturbations well, even resulting in collisions. However, as can be seen from Figure 3.9, the newly proposed method of using B-splines handles the initial disturbance well.

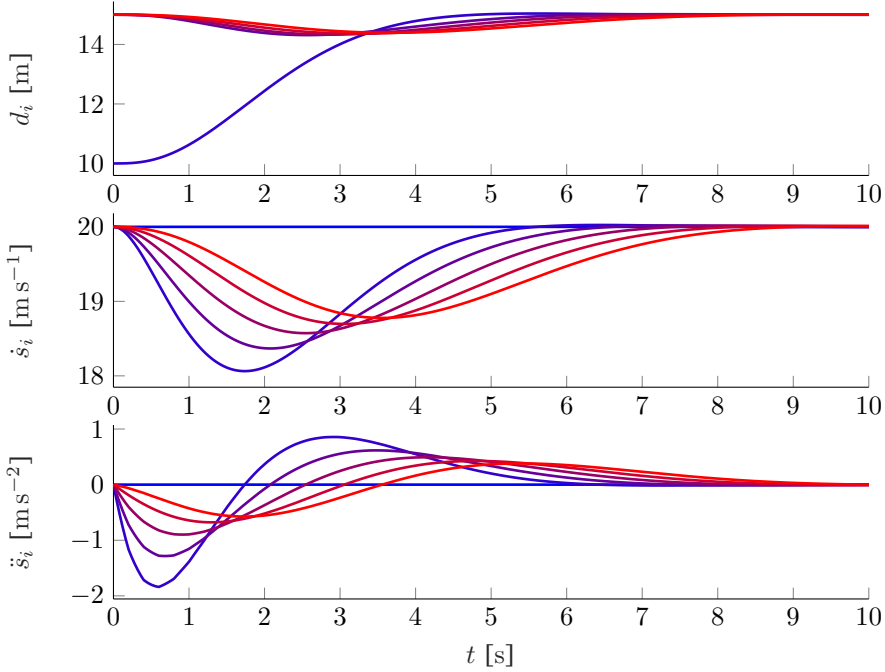


Figure 3.9: Results of B-spline based trajectories for disturbance rejection, disturbance of  $e_1(0) = 5$  m, in a platoon with six vehicles

The initial errors are all regulated towards zero, without collisions. Moreover, it can clearly be seen that the disturbance attenuates in upstream direction, with again the magnitude in acceleration decreasing for vehicles upstream. Hence, the response does not demonstrate string-unstable behavior.

Finally, the problem scenario of Figure 3.4b is repeated for the cooperative B-spline method, with identical string length and disturbance. The result is shown in Figure 3.10. Clearly, the perturbed last vehicle now remains collision free, as seen by the inter vehicle distance  $d_i(t)$ . This illustrates that this approach of constructing the trajectories is suitable, even for long platoons subject to perturbations.

### Communication and Planning Delay

In reality, a time delay is always present between the time at which the preceding vehicle construct its planning and the time at which the follower vehicle uses that information to construct its trajectory. This is due to communication delay and the asynchronous nature of planning updates between vehicles. Therefore, the same simulation as before is performed, but now information to the following vehicles is only available after a communication delay  $\zeta(t)$ . In the simulation, a



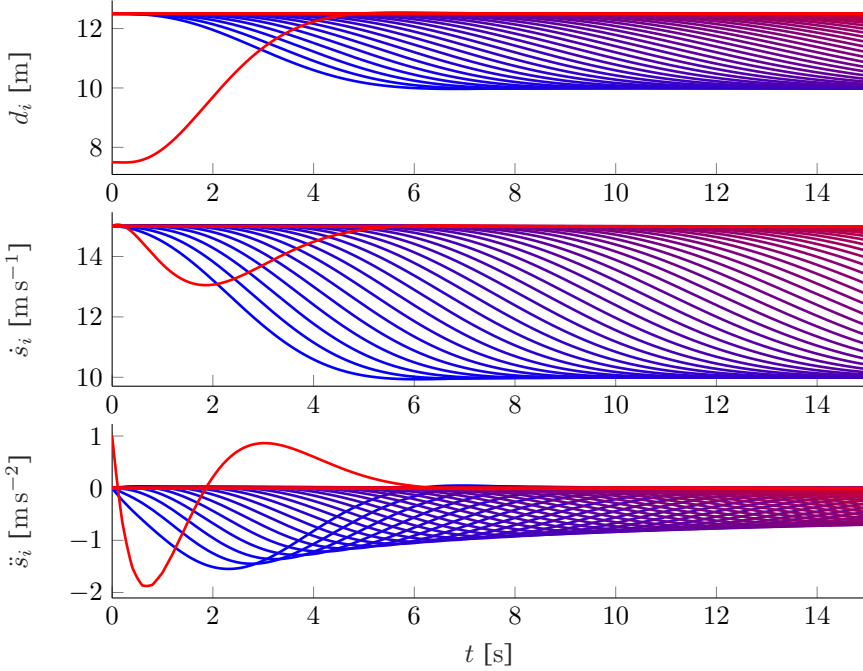


Figure 3.10: Recursive planning ( $t_p = 0.2$  s) for B-splines with  $p = 5$ ,  $n = 6$ , initial perturbed state  $\mathbf{S}_i(0)$  of (3.20).  $h_i = 0.5$  s,  $c_i = 5$  m. Executed trajectories by: Lead vehicle (—) and Last vehicle (—).

constant delay is assumed, although this is not required for the planner, since the time instance at which the planning is computed is always communicated. In this case,  $\zeta(t)$  does not only include the communication delay, but also the delay due to asynchronicity in the planning of consecutive vehicles, thus representing the total time between the planning update time,  $t_{i,k}$ , of the host vehicle, and the time instance,  $t_{k,i-1}$ , of the planning of the preceding vehicle that is available. This is illustrated in Figure 3.6. The simulation is set up in such a way that the following vehicle is able to directly update its planning, once the communicated information becomes available. In case the delay due to asynchronous planning is larger than the communication delay, the latter does not impact the performance of the string any more.

A delay of  $\zeta = 0.4$  s is used and the remaining parameters again the same as those used earlier in this section. The results are given in Figure 3.11, which shows that the response still does not demonstrate string unstable behavior, and gives nearly the same results as the non delayed case of Figure 3.7. This result emphasizes the strength of this approach, as string-unstable behavior is not observed despite significant time delays. It can be seen that the following

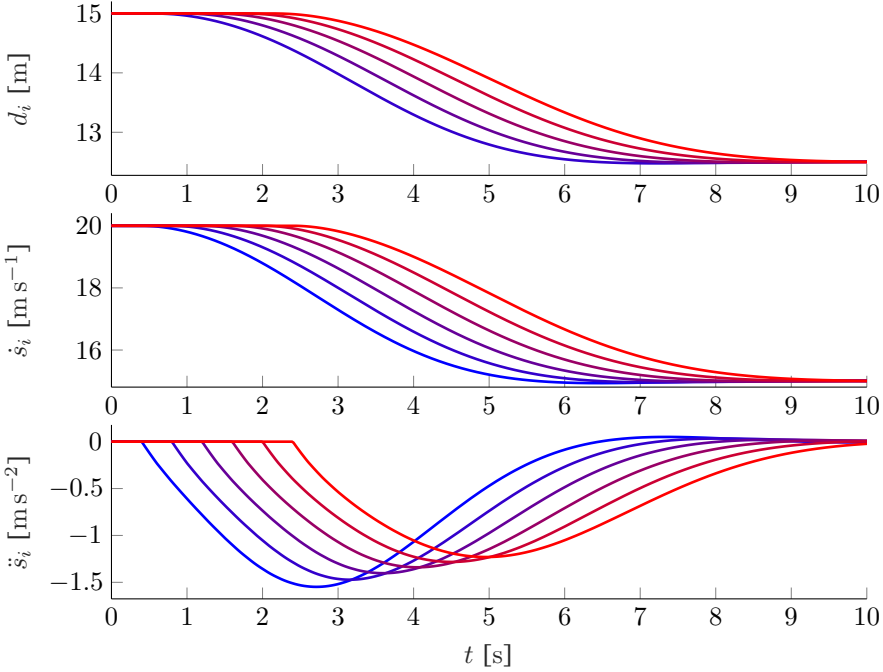


Figure 3.11: Results of B-spline based trajectories with  $p = 5$ ,  $n = 6$ , for a velocity change,  $\zeta(t) = 0.4$  s,  $\forall t$ ,  $h_i = 0.5$  s

vehicles initially keep planning to remain driving at the initial velocity, when the information of the velocity change has not been received yet.

It was noticed that the following three parameters are important in the resulting behavior of the string: the planning horizon  $T_i$ , the time gap  $h_i$ , and the delay time  $\zeta$ . To evaluate the performance, a numerical parameter study is performed to see for which combinations the platoon does not show string-unstable behavior for the selected manoeuvres. As the time horizon is often the result of requirements for the autonomous planner, it is fixed to  $T_i = 5$  s,  $\forall i$  as before, with an additional simulation study performed for a longer horizon  $T_i = 10$  s,  $\forall i$  separately. For this study, several manoeuvres are simulated with a string of 50 vehicles. The manoeuvres that are tested are as follows. Velocity transitions:

(i)  $\dot{s}_i(0) = 20 \text{ m s}^{-1} \rightarrow \lim_{t \rightarrow \infty} \dot{s}_i(t) = 15 \text{ m s}^{-1}, \forall i$ , as illustrated in Figure 3.11;

(ii)  $\dot{s}_i(0) = 20 \text{ m s}^{-1} \rightarrow \lim_{t \rightarrow \infty} \dot{s}_i(t) = 1 \text{ m s}^{-1}, \forall i$ ;

(iii)  $\dot{s}_i(0) = 5 \text{ m s}^{-1} \rightarrow \lim_{t \rightarrow \infty} \dot{s}_i(t) = 15 \text{ m s}^{-1}, \forall i$ ;

Driving at velocity  $\dot{s}_i(0) = 20 \text{ m s}^{-1}$ , with an initial error for the second vehicle of:

(iv)  $e_2(0) = 5$  m (vehicle 2 is further away then desired), as illustrated in Figure 3.9;

(v)  $e_2(0) = -5$  m (vehicle 2 is closer then desired).

Random acceleration input lead vehicle:

(vi)  $P_{j,1}^{(2)} \sim \mathcal{N}(0, 1)$ ,  $j \in \{1, \dots, n-2\}$ , with  $P_{n-1,1}^{(2)} = 0$

Scenarios (i)-(iii) are used to cover multiple velocities. Scenarios (iv)-(v) demonstrate more real world scenarios, where the velocity profile is determined by a vehicle correcting an error. Finally, twenty different random scenarios, described by (vi), are used as worst case scenarios, covering a larger frequency range, where the acceleration control points of the first plan step are taken from the normal distribution  $\mathcal{N}(0, 1)$ , with zero mean, and variance of 1. After this first plan step, (3.35) is used for the remainder of the simulation. An example of such a scenario is illustrated in Figure 3.12, which shows the response of 50 vehicles. It can be seen that the behavior in the scenario is string stable for the particular combination of delay  $\zeta$  and time gap  $h$ . It should be noted that since the lead vehicle is assumed to plan trajectories by means of B-splines, the random choice of control points still results in smooth trajectories, as can also clearly be observed in Figure 3.12.

A grid over  $h$  and  $\zeta$  is used in combination with a bisection method to find parameter values for which the platoon does not demonstrate  $\mathcal{L}_2$  string-unstable behavior for all simulated manoeuvres (i) - (v), and twenty scenarios of (vi), according to (3.2) for the acceleration  $\ddot{s}_i(t)$ . For a given time gap  $h$ , the maximum delay  $\zeta$  for which the simulated platoon demonstrates string-unstable behavior, are then shown in Figure 3.13 and Figure 3.14. For very large delays, the error that has accumulated over the delay becomes too large to be compensated with a sufficiently small acceleration, resulting in string unstable behavior. However, the delay at which this happens is far larger then in the traditional CACC controller from for example [27]. Clearly, by utilizing future information, string-unstable behavior can still be prevented with large delays as indicated by the green area. For a larger time horizon, a small advantage can be seen, as string unstable behavior can be prevented with a slightly larger time delay as was the case for the smaller time horizon.

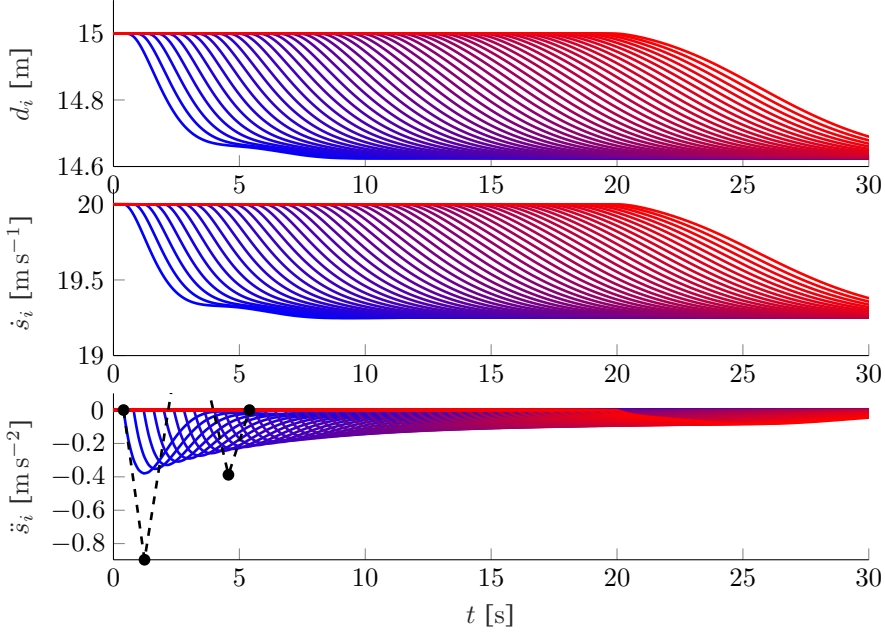


Figure 3.12: Single scenario of (vi),  $\zeta(t) = 0.4$  s,  $\forall t$ ,  $h_i = 0.5$  s, control polygon of the random input  $P_{j,1}^{(2)} \sim \mathcal{N}(0, 1)$  (- • -)

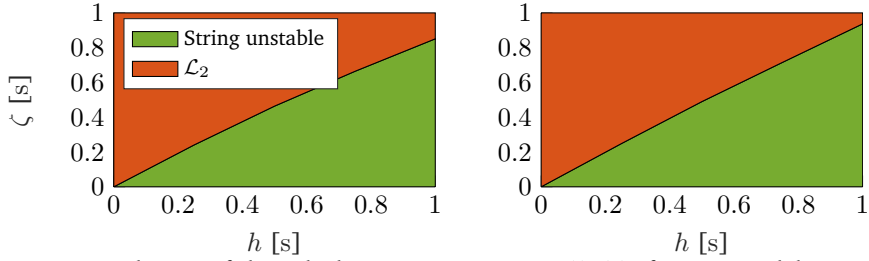


Figure 3.13: Behavior of the vehicle string in scenarios (i)-(v), for various delays  $\zeta$ , and time gaps  $h_i = h$ ,  $\forall i$ . Left:  $T_i = 5$  s; Right:  $T_i = 10$  s.

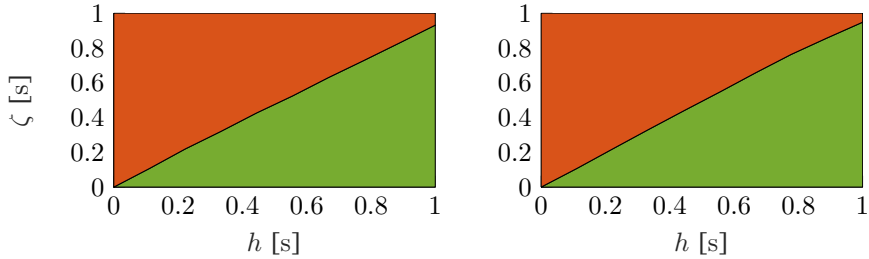


Figure 3.14: Behavior of the vehicle string in thirty random scenarios of (vi), for various delays  $\zeta$ , and time gaps  $h_i = h$ ,  $\forall i$ . Left:  $T_i = 5$  s; Right:  $T_i = 10$  s.

### 3.4 Gap Closing B-spline Trajectories<sup>3</sup>

While the method of Section 3.3 works well for vehicle following, large accelerations are generated in gap closing scenarios, due to large initial spacing error. These result in discomfort for passengers. This problem is not unique to the trajectory planning method. In fact, various solutions have been proposed for the traditional feedback CACC system. In [131], CACC is evaluated in real life traffic situations. Three controllers are developed for the tasks of velocity keeping, gap closing and distance keeping. In [132], a single controller is used for both distance keeping as well as gap closing and gap making, in which the control objective is modified accordingly. In [133] [134], Artificial Potential Fields are used for the synthesis of a non-linear controller, while maintaining a single control objective. However, these methods do not make use of the benefits of cooperative trajectory planning.

This section presents a unified method for cooperative trajectory planning, that integrates both comfortable gap closing as well as string-stable distance regulation. The main contribution is a time-varying spacing policy, that is used for the construction of B-spline trajectories. Both varying time gap and standstill distance are considered and compared with the constant spacing policy.

#### 3.4.1 Variable Spacing Policy

The objective is to construct B-spline trajectories such that gap closing is comfortable and independent of distance to the preceding vehicle, while distance keeping is unaffected. To achieve this, the spacing policy is adjusted as a function of time, such that the planner does not attempt to close the gap all at once. Therefore, either the speed-independent reference standstill distance  $c_{r,i}(t)$  or the reference time gap  $h_{r,i}(t)$  is made a function of time. Both time-varying parameters are lower bounded by user specified values  $\underline{h}_i$  and  $\underline{c}_i$ . This ensures that only gap closing is affected, while distance keeping remains unchanged. The spacing policy (3.4) is rewritten to include time dependent  $c_{r,i}(t)$  and  $h_{r,i}(t)$  as

$$d_{r,i}(t) = c_{r,i}(t) + h_{r,i}(t)\dot{s}_i(t), \quad (3.36)$$

$$e_i(t) = d_i(t) - d_{r,i}(t) = s_{i-1}(t) - s_i(t) - c_{r,i}(t) - h_{r,i}(t)\dot{s}_i(t). \quad (3.37)$$

The control points of the B-spline trajectories are found by modifying (3.32), to also include a time-varying time gap  $h_{r,i,k}(t)$  and standstill distance  $c_{r,i,k}(t)$  of

---

<sup>3</sup>This section is based on [56].

plan step  $k$

$$\begin{bmatrix} P_{3,i,k} \\ P_{4,i,k} \\ \vdots \\ P_{n,i,k} \end{bmatrix} = \mathbf{\Omega}_{12}^{-1} \left( \begin{bmatrix} s_{i-1}(\mu_3 + t_{i,k}) - c_{r,i,k}(\mu_3 + t_{i,k}) - L_i \\ s_{i-1}(\mu_4 + t_{i,k}) - c_{r,i,k}(\mu_4 + t_{i,k}) - L_i \\ \vdots \\ s_{i-1}(\mu_n + t_{i,k}) - c_{r,i,k}(\mu_n + t_{i,k}) - L_i \end{bmatrix} - \mathbf{\Omega}_{11} \begin{bmatrix} P_{0,i,k} \\ P_{1,i,k} \\ P_{2,i,k} \end{bmatrix} \right),$$

$$\begin{bmatrix} \mathbf{\Omega}_{11} & \mathbf{\Omega}_{12} \end{bmatrix} = \begin{bmatrix} \mathcal{B}_{n,p}(\mu_3) + h_{r,i,k}(\mu_3 + t_{i,k}) \mathcal{B}_{n-1,p-1}(\mu_3) \Delta_{10} \\ \mathcal{B}_{n,p}(\mu_4) + h_{r,i,k}(\mu_4 + t_{i,k}) \mathcal{B}_{n-1,p-1}(\mu_4) \Delta_{10} \\ \vdots \\ \mathcal{B}_{n,p}(\mu_n) + h_{r,i,k}(\mu_n + t_{i,k}) \mathcal{B}_{n-1,p-1}(\mu_n) \Delta_{10} \end{bmatrix}, \quad (3.38)$$

where  $h_{r,i,k}(t)$  and  $c_{r,i,k}(t)$  are respectively the reference time-varying time gap and standstill distance defined in plan step  $k$ .

### Variable Time Gap

First, a varying time gap  $h_{r,i}(t)$ , is considered. The initial time gap  $h_i(t_{i,k})$  is determined by rewriting (3.37) with  $e_i(t_{i,k}) = 0$  and  $c_{r,i}(t) = \underline{c}_i$  as,

$$h_i(t_{i,k}) = \frac{s_{i-1}(t_{i,k}) - s_i(t_{i,k}) - L_i - \underline{c}_i}{\dot{s}_i(t_{i,k})}. \quad (3.39)$$

The desired time gap is defined over time interval  $t \in [t_{i,k}, t_{i,k} + T_i]$  as a linearly decreasing function starting in  $h_i(t_{i,k})$  and lower bounded by the final value  $\underline{h}_i$  as typically used in vehicle-following situations:

$$h_{r,i,k}(t) = \max [h_i(t_{i,k}) + \phi_{i,k} \cdot (t - t_{i,k}), \underline{h}_i], \quad (3.40)$$

in which,  $\phi_{i,k} < 0$  is a parameter used to control the rate of change of  $h_{r,i,k}(t)$ .

**Fixed decrease rate of reference time gap** Initially, a user defined constant value is used for  $\phi_{i,k} = \underline{\phi}_i, \forall k$ , which can be used for comfort tuning. Due to (3.40), only the gap closing aspect is modified and distance keeping is maintained and  $h_{r,i}(t)$  never drops below  $\underline{h}_i$ . This is illustrated in Figure 3.15a.

The value for  $\phi_{i,k}$  can be used to determine the velocity with which the gap is closed. This can be seen by writing down the dynamics, in which a constant rate of change is assumed for the time gap  $\dot{h}_{r,i,k} = \phi_{i,k} = \phi_i \forall k$ ,

$$\dot{e}_i = v_{i-1} - v_i - \phi_i v_i - h_{r,i,k} a_i, \quad (3.41)$$

$$\dot{v}_i = a_i, \quad (3.42)$$

where  $v_i := \dot{s}_i(t)$ . This system has the following equilibrium for a given velocity  $v_{i-1}$  of vehicle  $i - 1$  and  $a_i = 0$ :

$$v_i = \frac{1}{1 + \phi_i} v_{i-1}. \quad (3.43)$$

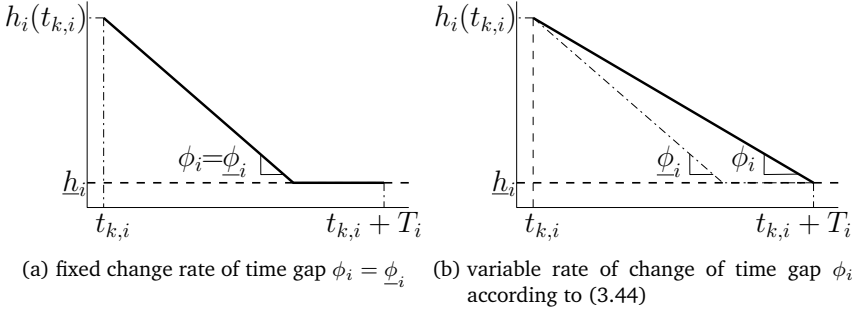


Figure 3.15: Desired time gap  $h_{r,i,k}(t)$  (3.40) of plan step  $k$ .

Thus  $\phi_i \in (-1, 0)$  can be used to control the velocity difference with which the gap to the preceding vehicle is closed. This is also somewhat intuitive, as it would be desirable to close the gap on a highway with a larger velocity difference than in urban areas. However, using a constant  $\phi_i = \underline{\phi}_i$  requires  $\dot{s}_i(t)$  to be discontinuous, where the saturation in (3.40) occurs. This discontinuity originates from the vehicle being requested to maintain a higher velocity (to reduce the time gap) and then instantaneously switch to maintaining a constant time gap at a lower velocity. Due to the continuity requirements of the B-spline, this introduces oscillations when fitting the spline, which is undesirable.

**Decrease rate of reference time gap depending on plan step** To realise a continuous gap closing velocity, a second method is presented that uses (3.40), but in which the slope  $\phi_{i,k}$  is adjusted, such that  $h_i(t_{i,k}) + \phi_{i,k}T_i \geq \underline{h}_i$  (i.e., the linear decrease in plan step  $k$  does not drop below  $\underline{h}_i$ ), via

$$\phi_{i,k} = \begin{cases} \underline{\phi}_i, & \text{if } h_i(t_{i,k}) + \underline{\phi}_i T_i \geq \underline{h}_i, \\ \frac{h_i(t_{i,k}) - \underline{h}_i}{T_i}, & \text{otherwise,} \end{cases} \quad (3.44)$$

and no saturation is required. Note that in doing so, no discontinuity in velocity occurs within the planning horizon, which can also be seen when comparing Figure 3.15a and Figure 3.15b. A discontinuity instead occurs at the end of each time horizon, such that no oscillations occur in the fitting process of the B-spline. Note that this naturally introduces a smoothing effect as the vehicle starts to slow down slightly as soon as the minimum time gap comes within the horizon at the current velocity, according to the second equation in (3.44). Then, every next planning update, the initial time gap has decreased further, such that the slope  $\phi_{i,k}$  decreases even more, converging to zero as  $t \rightarrow \infty$ . Clearly, this spacing strategy changes every plan step due to (3.44). This is illustrated in Figure 3.16a, which shows the desired time gap  $h_{r,i}(t)$  constructed in consecutive planning cycles. This shows that the newly created desired time gap

does not coincide with the one that was planned in the previous plan step. As a result, temporal consistency is lost, and the planned trajectories do not coincide with those of the previous planning cycle. This is undesirable, as the communicated trajectories are used for planning in following vehicles as well.

**Dynamics for reference time gap** To achieve temporal consistency, observe that an identical smoothing effect can be achieved by creating dynamics for  $h_{r,i}$  as following

$$\dot{h}_{r,i,k}(t) = \begin{cases} \underline{\phi}_i, & \text{if } h_{r,i,k}(t) + \underline{\phi}_i T_i > \underline{h}_i, \\ \frac{h_{r,i,k}(t) - \underline{h}_i}{T_i}, & \text{otherwise.} \end{cases} \quad (3.45)$$

The resulting reference trajectory for  $h_{r,i,k}(t)$ , starting in the initial condition  $h_{r,i,k}(t_{i,k}) = h_i(t_{i,k})$  can then be found. This resulting trajectory consists of two parts, the constant rate part and the exponential part. The time gap  $h_{tr}$  where the dynamics (3.45) transitions to the exponential part, is found by equating the condition in (3.45), and noting that  $h_{tr}$  cannot be bigger than the initial time gap  $h_i(t_{i,k})$  (as the trajectory then simply starts in the exponential part). The corresponding time  $t_{tr}$  is then found via

$$h_{tr} = \min [\underline{h}_i - \underline{\phi}_i T_i, h_i(t_{i,k})], \quad t_{tr} = \frac{h_i(t_{i,k}) - h_{tr}}{-\underline{\phi}_i} + t_{i,k}.$$

This transition is also illustrated in Figure 3.16b. Using this transition, the trajectories,  $h_{r,i}(t)$  for  $t \in [t_{i,k}, t_{i,k} + T_i]$ , with initial condition  $h_{r,i}(t_{i,k}) = h_i(t_{i,k})$ , can be computed as

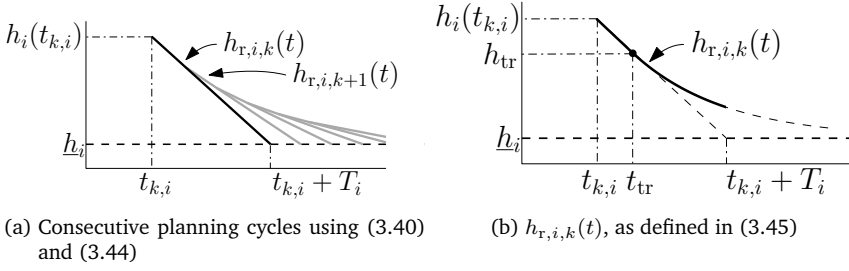
$$h_{r,i}(t) = \begin{cases} h_i(t_{i,k}) + \underline{\phi}_i (t - t_{i,k}), & \text{if } t < t_{tr}, \\ (h_{tr} - \underline{h}_i) \exp\left(\frac{t_{tr} - t}{T_i}\right) + \underline{h}_i, & \text{otherwise.} \end{cases} \quad (3.46)$$

This is illustrated in Figure 3.16b, which plots the resulting  $h_{r,i}(t)$ . As a result, the strategy of (3.46) does result in temporal consistency, as the desired time gap  $h_{r,i}$  follows the same curve in the next planning update. An important drawback of using strategies for desired time gap  $h_{r,i,k}(t)$  is that at low velocities or standstill, the current time gap from (3.39) is very large or undefined, respectively.

### Variable Standstill Distance

To overcome the drawback that originates from using  $h_{r,i,k}(t)$  at low velocities, a second approach can be conceived. A reference standstill distance  $c_{r,i,k}(t)$  does not suffer from these issues. In this case, the time gap  $h_{r,i}(t) = \underline{h}_i$  is assumed



Figure 3.16: Desired time gap,  $h_{r,i,k}(t)$ .

constant. Instead, standstill distance is now constructed to be dependent on time, with the current standstill distance being determined as

$$c_i(t) = s_{i-1}(t) - s_i(t) - \underline{h}_i \dot{s}_i(t) - L_i. \quad (3.47)$$

Similarly to (3.45), dynamics for the desired gap  $c_{r,i}(t)$  can be derived as

$$\dot{c}_{r,i,k}(t) = \begin{cases} \underline{\psi}_i, & \text{if } c_{r,i,k}(t) + \underline{\psi}_i T_i > c_i, \\ \frac{c_{r,i,k}(t) - c_i}{T_i}, & \text{otherwise,} \end{cases} \quad (3.48)$$

in which  $\underline{\psi}_i$  is the rate at which the standstill distance decreases. This rate is determined with three considerations. The resulting velocity should not be smaller than that of (3.43) for comparison with strategy (3.46). In case the vehicle already drives faster than (3.43), no unnecessary hard braking should occur for user comfort. Finally, in case both vehicles are driving too slow or are at standstill, a minimum standstill distance rate of  $\underline{v}_{cl}$  is utilized instead, to ensure the gap can always be decreased. These lower bounds on  $\underline{\psi}$  yield:

$$\underline{\psi}_i = -\max \left[ \left( \frac{1}{1+\phi_i} - 1 \right) v_{m,i-1,k} \quad \dot{s}_i(t_{i,k}) - v_{m,i-1,k} \quad \underline{v}_{cl} \right], \quad (3.49)$$

$$v_{m,i-1,k} = \frac{1}{T_i} (s_{i-1,k}(t_{i,k}) + s_{i-1,k}(t_{i,k} + T_i)), \quad (3.50)$$

where  $v_{m,i-1,k}$  represents the mean velocity of vehicle  $i - 1$  during plan step  $k$ , which is used to scale the rate at which to close the gap. The transition time  $t_{tr}$  and standstill distance  $c_{tr}$  in the dynamics (3.48) can be determined similar as before via

$$c_{tr} = \min [c_i - \underline{\psi}_i T_i, \quad c_i(t_{i,k})], \quad t_{tr} = \frac{c_i(t_{i,k}) - c_{tr}}{-\underline{\psi}_i} + t_{i,k},$$

which can be used to compute the trajectories for  $c_{r,i}(t)$

$$c_{r,i}(t) = \begin{cases} c_i(t_{i,k}) + \underline{\psi}_i(t - t_{i,k}), & \text{if } t < t_{tr}, \\ (c_{tr} - \underline{c}_i) \exp\left(\frac{t_{tr} - t}{T_i}\right) + \underline{c}_i, & \text{otherwise.} \end{cases} \quad (3.51)$$

The gap closing strategy and resulting trajectory planner should be able to operate in all conditions in which the vehicle can reasonably be expected to operate. The handling of low velocities or standstill is thus an important criterion, as well as the lead vehicle not driving at a constant velocity. These gap closing strategies are all defined relative to the trajectory of the preceding vehicle via (3.32), such that a disturbance from the lead vehicle does not pose a problem.

### 3.4.2 Results

In this section, simulations are used to compare the four gap closing strategies in terms of user comfort, measured by acceleration and jerk, and temporal consistency. The four gap closing strategies can be summarized as follows.

- (i) linear decreasing time gap (3.40)
- (ii) receding time gap (3.44)
- (iii) dynamic time gap (3.46)
- (iv) dynamic standstill distance (3.51)

The simulations are set up with  $p = 5$ ,  $n = 6$ ,  $T_{1,2} = 5$  s and planning is updated with planning interval  $t_p = 0.2$  s. In all cases, the user defined values are chosen as  $\underline{\phi}_i = -0.1$ ,  $\forall i$ ,  $\underline{h}_i = 0.5$  s,  $\forall i$ ,  $\underline{c}_i = 5$  m,  $\forall i$ ,  $L_i = 0$  m,  $\forall i$  and  $\underline{v}_{cl} = 1$  m s<sup>-1</sup>.

Two vehicles are considered, which are driving at a forward velocity of  $\dot{s}_0(0) = \dot{s}_1(0) = 15$  m s<sup>-1</sup>. Note that if more vehicles are added, these vehicles would simply perform vehicle following behind the vehicle that is closing the gap. The initial spacing error of the following vehicle is set to  $e_1(0) = 25$  m. An overview of the vehicle response in terms of distance  $d_1(t)$  speed  $\dot{s}_1(t)$  and  $\ddot{s}_1(t)$ , is given in Figure 3.17. In this figure, the distance  $d_1(t) = s_0(t) - s_1(t)$ , the velocity  $\dot{s}_1(t)$  and acceleration  $\ddot{s}_1(t)$  are given for all the introduced spacing policies. The original B-spline trajectory planning strategy with constant spacing policy is also shown. This system successfully regulates the initial error towards zero. However, the corresponding control inputs are excessively large. A typical vehicle is not capable of achieving such large forward accelerations (here  $\|\ddot{s}_1(t)\| = 9.19$  m s<sup>-2</sup>) due to limited drive train power. Moreover, this is highly uncomfortable for the passengers in terms of both acceleration and high approaching velocity (here  $\|\dot{s}_1(t)\| = 24.67$  m s<sup>-1</sup>).

The gap-closing strategy (i) constructs the adjusted time gap,  $h_{r,i,k}(t)$  via (3.40) and uses a constant rate of change for the time gap  $\phi_{i,k} = \underline{\phi}_i$ ,  $\forall k$ . Clearly, the gap is closed more gradually, with far smaller accelerations, than is the case for the system without gap-closing strategy. The closing velocity can indeed be

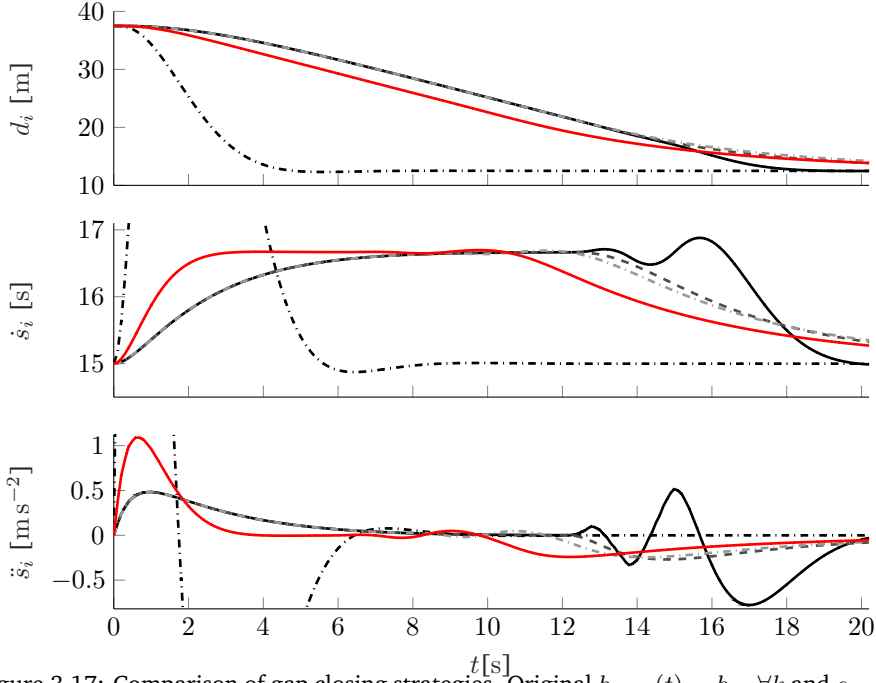


Figure 3.17: Comparison of gap closing strategies. Original  $h_{r,1,k}(t) = \underline{h}_1, \forall k$  and  $c_{r,1,k} = \underline{c}_1, \forall k$  (---),  $h_{r,1,k}(t)$  via (3.40)  $\phi_{1,k} = \underline{\phi}_1, \forall k$  (—),  $h_{r,1,k}(t)$  via (3.40)  $\phi_{1,k}$  via (3.44) (- -),  $h_{r,1,k}(t)$  via (3.46) (-·-),  $c_{r,1,k}(t)$  via (3.51) (—).

seen to be related to the parameter  $\underline{\phi}_i$  according to (3.43). When the vehicle comes near the preceding vehicle, the time gap is saturated at the minimum value  $h_{r,i,k}(t) = \underline{h}_i, \forall k$  and the vehicle starts decelerating. However, the discontinuity in the time gap change rate results in oscillatory behavior in terms of acceleration. The resulting response is fairly jerky and uncomfortable for passengers. These oscillations are absent in the response given by the gap closing strategy (ii), which uses a variable change rate  $\phi_{i,k}$  (3.44) based on the current time gap. As a result, a more gradual deceleration is obtained, which is far more comfortable for the passengers. The response of strategy (iii) can be seen to nearly coincide with the strategy (iii). The small difference is due to the dynamics of  $h_{r,i,k}(t)$  being specified in continuous time via (3.45), whereas the strategy using a variable  $\phi_{i,k}$  via (3.44), can be viewed as discrete time with the planning update time,  $t_p$  being the sample time of this discrete time system. Finally, strategy (iv), in which a variable standstill distance  $c_{r,i,k}(t)$  (3.51) is used, is shown. The response can be seen to be fairly similar to the response of strategy (iii). It can be seen that  $\underline{\psi}_i$  via (3.49) is indeed a proper analogue to  $\underline{\phi}_i$  as the same approaching velocity is obtained. However, this approaching velocity

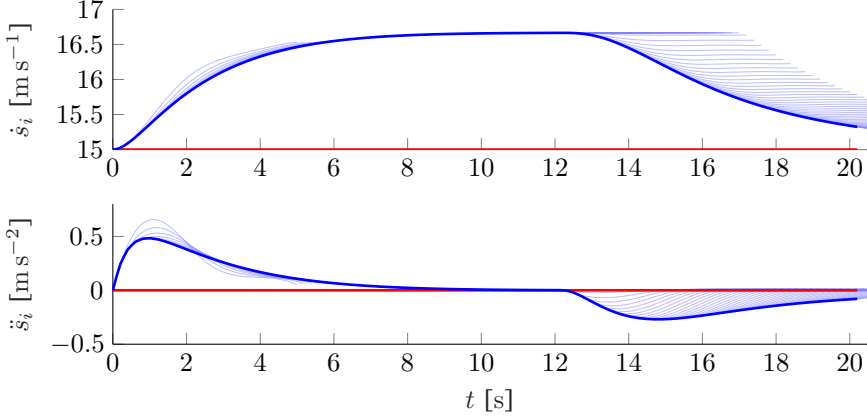
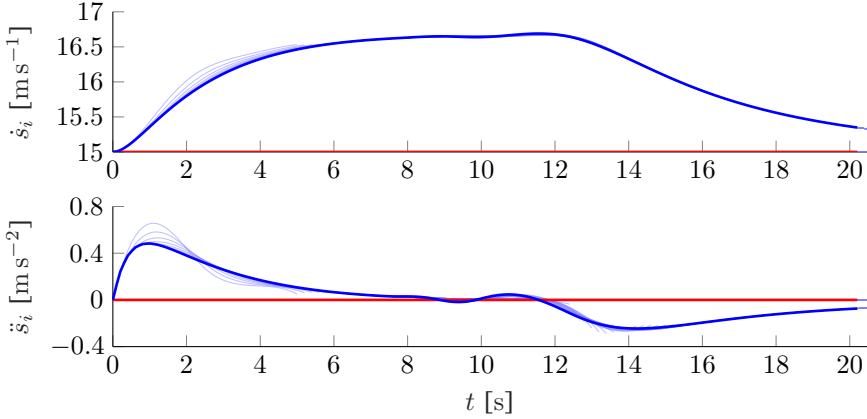
(a) Gap closing strategy (ii), using (3.40) with variable  $\phi_{i,k}$  as in (3.44)(b) Gap closing strategy (iii), with dynamics for  $h_{r,i,k}(t)$  via (3.45)

Figure 3.18: Comparison of strategy (ii) and strategy (iii). Executed trajectories: Lead vehicle (—), Following vehicle (—); Planned trajectories of follower vehicle (—)

is realised much faster due to a higher initial acceleration, which is less comfortable for the passengers.

To better understand the advantage of strategy (ii) over strategy (iii), a comparison of the consecutive trajectories is shown in Figure 3.18, where Figure 3.18a and Figure 3.18b illustrate strategies (ii) and (iii) respectively. Although the executed response of both strategies is fairly similar (see also Figure fig: Comparison), the consecutively planned trajectories differ significantly. For strategy (ii), the planned trajectories in light blue all converge to slightly lower constant velocities given in (3.43), corresponding to the lower

time gap rates,  $\phi_{i,k}$ . This implies temporal consistency is indeed lost. This temporal consistency is regained when using the gap closing strategy (iii). This is illustrated in Figure 3.18b, where it can be seen that all the consecutive planned trajectories in the deceleration phase all remain close to the previously planned trajectory, such that temporal consistency is achieved. As a result, the executed trajectory also follows these planning cycles closely. This makes the vehicle behave more predictable, which is beneficial when other vehicles base their planning around the previously communicated trajectories.

Despite having a larger initial acceleration strategy (iv), has the advantage that it is also capable of working well at low velocities. This is shown in Figure 3.19a, where the initial velocities of both vehicles are set to  $\dot{s}_i(0) = 0 \text{ m s}^{-1}$ ,  $\forall i$  and the second vehicle has an initial error  $e_1(0) = 25 \text{ m}$  back with respect to the spacing policy. The lead vehicle remains stationary for 4 seconds, after which it accelerates to  $\dot{s}_1(t) = 10 \text{ m s}^{-1}$ . It can be seen that the strategy using  $c_{r,i,k}(t)$  is capable of handling standstill, closes the distance initially with predefined velocity  $\underline{v}_{cl}$ . When the lead vehicle accelerates, the planning of the host vehicle is updated accordingly, even though the newly communicated trajectory of the lead vehicle is not consistent with the previously communicated trajectory. This can be seen by the sudden change in planned trajectories, where initially the host vehicle plans to maintain its velocity until  $t = 8.5 \text{ s}$ , but reacts accordingly once the preceding vehicle starts moving. This demonstrates the versatility of the approach. Another important situation is the approach of a stationary vehicle, which can occur for instance while approaching a traffic light. In this scenario, the host vehicle approaches with a velocity of  $\dot{s}_2(0) = 10 \text{ m s}^{-1}$  and error  $e_1(0) = 50 \text{ m}$  backwards of the desired position. As can be seen from Figure 3.19b, the vehicle gradually brakes towards the stationary vehicle. This is due to the added term  $\dot{s}_{i,k}(t_i) - \left( \frac{1}{T_i} (s_{i-1,k}(t_{i,k}) + s_{i-1,k}(t_{i,k} + T_i)) \right)$  in (3.49), that prevents the vehicle from hard initial braking and closing the remaining gap with  $\underline{v}_{cl}$ . These examples demonstrate the the gap closing strategy using (3.51) and (3.49), is not only capable of performing normal gap closing, but also is able to operate in various perturbed situations, hence making it suitable for practical implementation. Note that with these examples all three modes of (3.49) are demonstrated and simulating at other velocities does not lead to new insights.

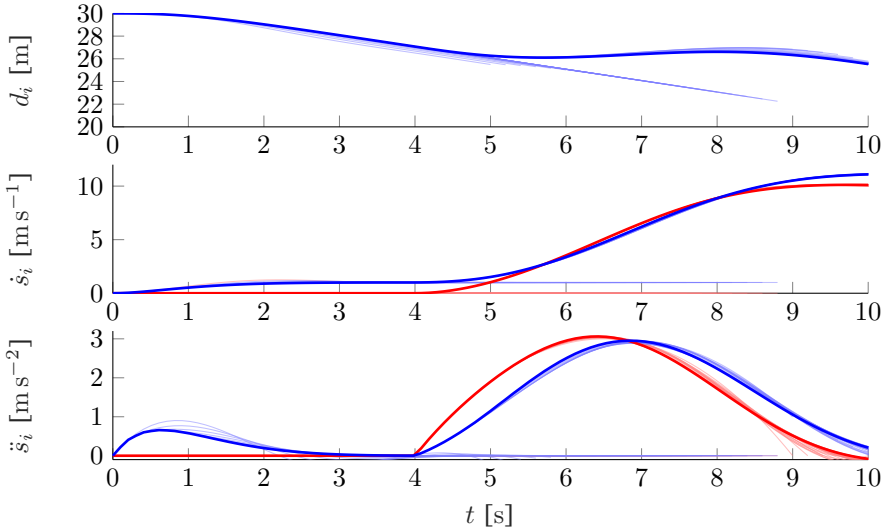
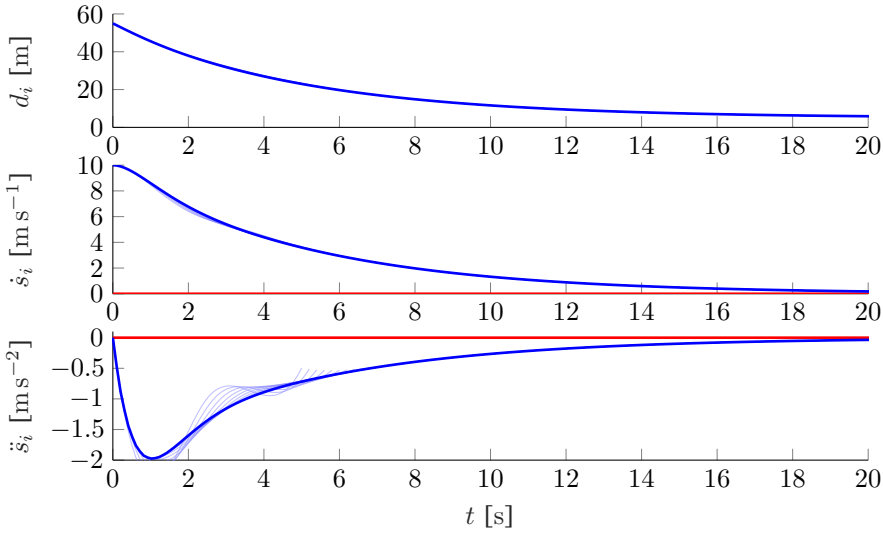
(a) Depart from standstill with spacing error  $e_1(0) = 25$  m.(b) Approach of a standstill vehicle, spacing error  $e_1(0) = 50$  m.

Figure 3.19: Special cases for Strategy (iv), using  $c_{r,i,k}(t)$  via (3.48) and (3.49). Executed trajectories: Lead vehicle (—) Following vehicle (—); Planned trajectories (—) and (—)

### 3.5 Summary

In this chapter, the cooperative trajectory planning approach is discussed. First it was demonstrated that the polynomial planner of [18] was not suitable for string stable vehicle following. A potential solution based on [122] was shown to be able to result in string stable vehicle following, which relied on shifting the arrival time forward in time for increasing vehicle index,  $i$  in the platoon. However, this results in potential collisions when the platoon is not initially in equilibrium.

Next, a different approach is presented that makes use of B-spline trajectories. In this approach, the spacing error is minimized over several time instances spanned over the planning horizon, instead of a single time instance at the end of the time horizon, as was the case with the polynomial trajectories. This solved the issue with potential collisions and was shown to demonstrate string-stable behavior in several scenarios by means of numerical simulation. Moreover, by utilizing future information, string-unstable behavior can still be prevented, even with large planning delays, which was shown by means of simulations.

To improve passenger comfort, several gap-closing approaches are presented. These approaches modify the spacing policy as a function of time, by having a time-varying time-gap and time-varying standstill distance. The approaches are compared in terms of passenger comfort and temporal consistency. The time-varying standstill distance is concluded to be most suitable for implementation, as it was also well defined at standstill.

The following vehicle does requires that the preceding vehicle also utilizes a B-spline trajectory planning approach. It could occur that the preceding vehicle has also selected a cooperative trajectory, which is generated by an identical method, or it has selected an autonomous (i.e., non-cooperative) trajectory, which is generated by a autonomous trajectory planner.

# Chapter 4

## Autonomous Planner and Combined Framework

Chapter 3, discusses methods to construct cooperative vehicle following trajectories. These cooperative trajectories are only constructed in the curvilinear or longitudinal sense. Cooperative vehicle following in the lateral direction can lead to either corner cutting or amplification of the lateral motion in the upstream direction, both of which are detrimental for longer platoons. To prevent these issues, the cooperative trajectories include a lateral part that regulates the lateral position of the host vehicle with respect to road.

The construction of cooperative trajectories requires communicated trajectories of preceding vehicles, which are also parametrised as B-splines. This communicated trajectory is not necessarily always available (e.g., for the lead vehicle of the platoon), or not always beneficial to follow (e.g., when the preceding vehicle is driving too slow). Therefore, the vehicles should not only be equipped with a cooperative vehicle following trajectory planner, but also with an autonomous planner. These autonomous trajectories enable the automated vehicle to handle a high variety of scenarios and consist of both a longitudinal and lateral part. Additionally, the construction of these lateral parts is also used to assign lateral trajectories to the longitudinal cooperative trajectories.

The construction of both cooperative and autonomous trajectories allows each vehicle to determine if either cooperative vehicle following, or autonomous driving is beneficial for the current scenario. Finally, a cost function is used to select the trajectory that is to be executed, which can be either a cooperative vehicle following trajectory, or an autonomous trajectory. This allows each vehicle in the platoon to determine for the given scenario whether or not it is beneficial to remain in the platoon, or to break from the platoon and potentially overtake slower road users.

This chapter first discusses how an automated vehicle can generate



autonomous B-spline trajectories. The method for generating lateral trajectories can also be used to provide the cooperative longitudinal trajectories with lateral trajectories. The second part of this chapter then presents cost functions that can be used to obtain the desired behavior, to select the most suitable trajectory.

## 4.1 Autonomous Planner

The lead vehicle of the platoon does not have a vehicle in front of it to follow. Instead, it generates trajectories in an autonomous manner. In Section 3.3 and Section 3.4, the B-spline trajectories of the lead vehicle are generated by simply fitting B-splines to minimum-jerk polynomials, because the trajectory of the lead vehicle was less important. However, using B-splines as trajectories introduces an advantage for constraints that so far have not been utilized. To construct the non-cooperative (or autonomous) trajectories for the vehicles the following objectives are considered. This lead vehicle might either be controlling its cruise speed, as is the case while driving on a highway, or could be regulating its terminal position, as is the case when coming to a stop before a traffic light. Either scenario can be constructed by means of B-spline trajectories. Inspired by [18], a method for speed control is conceived to generate a diverse set of trajectories, similar to the framework described in Section 2.1.4. From this set of trajectories, one is selected for execution.

As mentioned in Section 3.1.1, temporal consistency is desired for the planned trajectories, meaning that in unchanged circumstances, the same trajectory should be selected. It should be noted that a B-spline trajectory can only be exactly temporally consistent if the basis functions of plan step  $k$  can represent the same functions as in plan step  $k - 1$ . This can only happen if the corresponding knots in the knot vector,  $U_i$ , off-setted by the current time planning  $t_{i,k}$ , reoccur the next plan step at the same time instances, such that identical basis functions can be constructed. However, this would require knots at intervals of the planning time interval  $t_p$ , which would result in many knots ( $T_i/t_p$  to be precise), and also many control points, requiring more communication bandwidth and more computation time. The requirement on temporal consistency is therefore relaxed, and only practical temporal consistency is considered, which in this context means that the closest trajectory in the search space should be selected.

The polynomial trajectory planner of [18] was derived in such a manner that minimum jerk trajectories were generated, given a time horizon for the polynomial and a terminal state. Temporal consistency was then achieved by penalizing trajectories with a larger time horizon for the polynomial and including the selected trajectory of the previous planning cycle. However, as a consequence, higher energetic trajectories in terms of frequency content (and thus higher accelerations and jerk) are selected due to the prioritization of the

smaller time horizon. In this section, a different approach is taken. The trajectories are generated by means of a cost function that can include a jerk, acceleration and velocity error, hence higher energetic trajectories are automatically penalized. Moreover, the user has a significant influence on how comfortable or energetic the vehicle should behave. Additionally, no book keeping is required, which needs to keep track of previously selected trajectories, in order to achieve temporal consistency.

### 4.1.1 Longitudinal Trajectory Generation

Individual trajectories are generated by means of a minimization problem. This has as a major benefit that due to the principle of optimality, the consecutive planned trajectories demonstrate the aforementioned practical temporal consistency. For a given desired velocity  $\nu_i(t)$  and given initial condition  $\mathbf{S}_i(t_{i,k})$ , the trajectories can be computed.

A benefit of B-splines over polynomials as used in [18], is that the Control points  $\mathcal{P}_{i,k}$  and the control points of the  $d$ -th derivative  $\mathcal{P}_{i,k}^{(d)}$  have a physical representation. This can be seen by the strong convex hull property of the B-spline, which states that for  $t \in [u_j, u_{j+1})$ ,  $s_i(t)$  is in the convex hull of control points  $\{P_{l,i}, l \in \{j, j-1, \dots, j-p\}\}$ . Similarly to [103], it can be stated that

$$\underline{s_i^{(d)}} \leq \min(\mathcal{P}_{i,k}^{(d)}) \leq s_i^{(d)}(t) \leq \max(\mathcal{P}_{i,k}^{(d)}) \leq \overline{s_i^{(d)}}, \quad (4.1)$$

where  $\underline{s_i^{(d)}}$  and  $\overline{s_i^{(d)}}$  are respectively the lower- and upper bound for the  $d$ -th derivative of the spline  $s(t)$ , and  $\min(\mathcal{P}_{i,k}^{(d)})$  and  $\max(\mathcal{P}_{i,k}^{(d)})$ , denote the minimum and maximum element in  $\mathcal{P}_{i,k}^{(d)}$ . Hence, the maximum and minimum control points of the  $d$ -th derivative are conservative upper and lower bounds on the  $d$ -th derivative of the spline. This can for example be used to constrain the acceleration of the planned trajectories, based on for example the limitations of the powertrain of the vehicle. With this in mind, a least squares problem

$$\min_{\mathcal{P}_{i,k}} \|\mathbf{V}\mathcal{P}_{i,k} - \mathbf{w}\|^2, \quad (4.2a)$$

$$\text{s.t. } \mathbf{Z}\mathcal{P}_{i,k} = \mathbf{q}, \quad (4.2b)$$

can be set up to minimize the weighted squared sum of acceleration and jerk, while minimizing the squared error of the velocity control points with respect to a given reference velocity. To take into account a time-varying reference, the squared error for each velocity control point is taken with respect to a reference velocity at the location of its corresponding Greville abscissa (see also

Section 2.2.2)  $\mu_j^{(1)}, j \in \{0, \dots, n-1\}$

$$J_{s|i,k} = \left\| \underbrace{\begin{bmatrix} Q_{\dot{s}} \Delta_{10} \\ Q_{\ddot{s}} \Delta_{20} \\ Q_{\dddot{s}} \Delta_{30} \end{bmatrix}}_{\mathbf{V}_s} \mathcal{P}_{i,k} - Q_{\dot{s}} \underbrace{\begin{bmatrix} \nu_i(t_{i,k} + \mu_0^{(1)}) \\ \vdots \\ \nu_i(t_{i,k} + \mu_{n-1}^{(1)}) \\ \mathbf{0}_{n-1 \times 1} \\ \mathbf{0}_{n-2 \times 1} \end{bmatrix}}_{\mathbf{w}_s} \right\|_2, \quad (4.3a)$$

$$\text{s.t.} \quad \underbrace{\begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \Delta_{10} \\ \Delta_{20} \end{bmatrix}}_{\mathbf{Z}_s} \mathcal{P}_{i,k} = \underbrace{\mathbf{S}_i(t_{i,k})}_{\mathbf{q}_s}, \quad (4.3b)$$

where  $Q_{\dot{s}}$ ,  $Q_{\ddot{s}}$  and  $Q_{\dddot{s}}$ , represents the weight on velocity error, acceleration and jerk respectively. In Addition, the acceleration is constrained to  $\underline{s_i^{(2)}} \leq \ddot{s}_i(t) \leq \overline{s_i^{(2)}}$ . The continuity constraint is represented by (4.3b). The linear least squares problem with equality constraint (4.3) is solved to obtain  $\mathcal{P}_{i,k}$ , for the given weights  $Q_{\dot{s}}$ ,  $Q_{\ddot{s}}$  and  $Q_{\dddot{s}}$ .

The inequality of the acceleration constraint is handled differently, by means of the following sequential approach. First the linear least squares problem (4.3a) and (4.3b) is solved. The resulting control points are then checked sequentially if they satisfy the bound on acceleration. If a control point does not satisfy the bound, the control point is set equal to the bound, and the optimization is ran again, with the control points up until the one that was adjusted now fixed. The same check is done to prevent trajectories from reversing, by simply enforcing that  $P_{j,i} \geq P_{j-1,i}$ ,  $\forall j$ . This is summarized in Algorithm 4.2.

Running Algorithm 4.2 with constant reference velocity  $\nu_i(t) = 10 \text{ m s}^{-1}$ ,  $\forall t$ , weights  $\mathbf{Q}_s = [Q_{\dot{s}}, Q_{\ddot{s}}, Q_{\dddot{s}}] = [1, 0, 1]$ , and acceleration constraints  $\underline{s_i^{(2)}} = -1 \text{ m s}^{-2}$ ,  $\overline{s_i^{(2)}} = 3 \text{ m s}^{-2}$  results in the planned trajectories illustrated in Figure 4.1. Note that the consecutive trajectories do not exactly overlap, due to the fact that the B-splines in the consecutive planning cycles do not have the same basis functions. Despite not being exactly temporally consistent, it can be seen that trajectories are close to each other and thus are approximately temporal consistent. Also note that the lower bound on  $\ddot{s}_i(t)$  is satisfied by the bound on the corresponding control points.

#### 4.1.2 Discretized Search Space

Similarly in spirit to [18] and the framework discussed in Section 2.1.4, a discretized search space is used in order to generate a set of trajectories. With this approach, trajectories are generated and checked for collisions with

**Algorithm 4.2** Compute Control Points

---

```

1: function VELOCITYCONTROL( $\mathbf{V}_s, \mathbf{w}_s, \mathbf{Z}_s, \mathbf{q}_s, n, \underline{s_i^{(2)}}, \overline{s_i^{(2)}}$ )
2:    $\mathcal{P}_i \leftarrow \min_{\mathbf{Z}_s, \mathcal{P}_i = \mathbf{q}_s} (\|\mathbf{V}_s \mathcal{P}_i - \mathbf{w}_s\|_2)$ 
3:    $N \leftarrow \# \text{ rows of } \mathbf{Z}_s$  ▷ number of fixed control points
4:   while  $N \leq n + 1$  do ▷ while not all control points are fixed
5:     for  $j \leftarrow N + 1$  to  $n + 1$  do
6:       if  $P_{j-2,i}^{(2)}$  does not satisfy bounds  $\underline{s_i^{(2)}}, \overline{s_i^{(2)}}$  then
7:          $P_{j,i} \leftarrow$  set such that  $P_{j-2,i}^{(2)}$  satisfies the bound
8:          $N \leftarrow j$ 
9:       end if
10:      if  $P_{j,i} < P_{j-1,i}$  then ▷ do not allow reversing
11:         $P_{j,i} = P_{j-1,i}$ 
12:      end if
13:    end for
14:    if  $N < n + 1$  then ▷ redo optimization
15:       $\mathbf{Z}_s \leftarrow I_{N \times N}, \mathbf{q}_s \leftarrow [P_{1,i}, \dots, P_{N,i}]$ 
16:       $\mathcal{P}_i \leftarrow \min_{\mathbf{Z}_s, \mathcal{P}_i = \mathbf{q}_s} (\|\mathbf{V}_s \mathcal{P}_i - \mathbf{w}_s\|_2)$ 
17:    else ▷ all control points satisfy the bound
18:      return  $\mathcal{P}_i$ 
19:    end if
20:  end while
21: end function

```

---

(dynamic) obstacles and feasibility afterwards, rather than combining everything into a single optimization. This makes generating the trajectories simpler. To generate a diverse set of trajectories for the planner to choose from, both the reference velocity,  $\nu_i(t)$  as well as the weights  $\mathbf{Q}_s$  are varied. This results in trajectories with various terminal velocities, as well as various rise times for a given terminal velocity. To find a valid range for the terminal reference velocity  $\nu_i(t_{i,k} + T_i) \in [\nu_{\min}, \nu_{\max}]$ , that result in feasible trajectories that satisfy the acceleration constraint, a search space is computed.

To find a feasible range of terminal velocities, splines corresponding to the minimum and maximum acceleration are constructed. Note that by including an initial condition for the  $d - 1$ -th derivative of the spline,  $s^{(d-1)}(t_{i,k})$ , a unique mapping of  $\mathcal{P}_{i,k}^{(d)}$  back to  $\mathcal{P}_{i,k}^{(d-1)}$  can be created as

$$\mathcal{P}_{i,k}^{(d-1)} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ & \Delta_d^{(d-1)} & & \end{bmatrix}^{-1} \begin{bmatrix} s_i^{(d-1)}(t_{i,k}) \\ \mathcal{P}_{i,k}^{(d)} \end{bmatrix}. \quad (4.4)$$

This corresponds to finding the integral of  $s_i^{(d)}(t)$ ,  $t \in [t_{i,k}, t_{i,k} + T_i]$ . The control points corresponding to the maximum deceleration spline and maximum

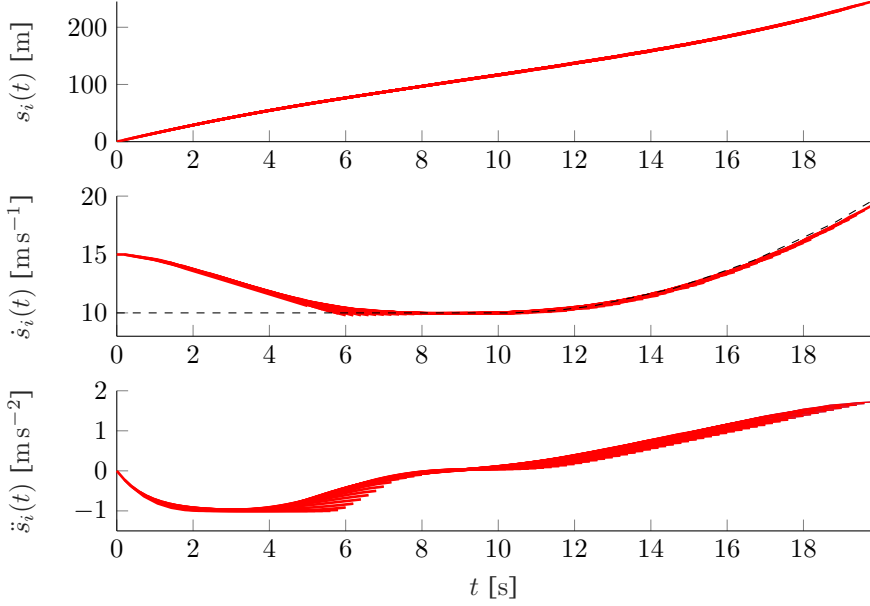


Figure 4.1: B-spline trajectory (—), reference velocity  $\nu_i(t)$  (---),  $n = 6$ ,  $p = 5$ ,  $\mathcal{Q}_s = [1, 0, 1]$ ,  $\underline{s_i^{(2)}} = -1 \text{ m s}^{-2}$ ,  $\overline{s_i^{(2)}} = 3 \text{ m s}^{-2}$

acceleration spline of plan step  $k$  are defined as

$$\mathcal{P}_{\min i, k}^{(2)} = \begin{bmatrix} \ddot{s}_i(t_{i, k}) & \underline{s_i^{(2)}} & \dots & \underline{s_i^{(2)}} \end{bmatrix}^\top, \quad (4.5)$$

$$\mathcal{P}_{\max i, k}^{(2)} = \begin{bmatrix} \ddot{s}_i(t_{i, k}) & \overline{s_i^{(2)}} & \dots & \overline{s_i^{(2)}} \end{bmatrix}^\top. \quad (4.6)$$

By combining these with the initial condition for the velocity  $\dot{s}(t_{i, k})$ , the control points of B-spline trajectories with the lowest and highest velocity respectively, that satisfy acceleration bound  $\underline{s_i^{(2)}} \leq \ddot{s}(t) \leq \overline{s_i^{(2)}}$  can be found via

$$\begin{bmatrix} \mathcal{P}_{\min i, k}^{(1)} & \mathcal{P}_{\max i, k}^{(1)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ & \Delta_{21} & & \end{bmatrix}^{-1} \begin{bmatrix} \dot{s}(t_{i, k}) & \dot{s}(t_{i, k}) \\ \mathcal{P}_{\min i, k}^{(2)} & \mathcal{P}_{\max i, k}^{(2)} \end{bmatrix}. \quad (4.7)$$

Now reference velocities with which trajectories are generated by means of Algorithm 4.2 vary between the splines described by  $\mathcal{P}_{\min i, k}^{(1)}$  and  $\mathcal{P}_{\max i, k}^{(1)}$ . This search space is divided in  $N_s$  parts. Hence, the vector  $\mathbf{w}$  in (4.3a) can be constructed by simply blending, maximum with the minimum velocity, according

to

$$\mathbf{w}_{s|l} = \left( \frac{l-1}{N_{\dot{s}}-1} \mathbf{w}_{s|\nu_{\min}} + \left( 1 - \frac{l-1}{N_{\dot{s}}-1} \right) \mathbf{w}_{s|\nu_{\max}} \right), \quad l \in \{1, \dots, N_{\dot{s}}\}, \quad (4.8)$$

$$\mathbf{w}_{s|\nu_{\min}} = Q_{\dot{s}} \begin{bmatrix} \max \left( [0 \quad \dots \quad 0 \quad 1] \mathcal{P}_{\min i, k}^{(1)}, 0 \right) \mathbf{1}_{n \times 1} \\ \mathbf{0}_{n-1 \times 1} \\ \mathbf{0}_{n-2 \times 1} \end{bmatrix}, \quad (4.9)$$

$$\mathbf{w}_{s|\nu_{\max}} = Q_{\dot{s}} \begin{bmatrix} \max \left( [0 \quad \dots \quad 0 \quad 1] \mathcal{P}_{\max i, k}^{(1)}, 0 \right) \mathbf{1}_{n \times 1} \\ \mathbf{0}_{n-1 \times 1} \\ \mathbf{0}_{n-2 \times 1} \end{bmatrix}. \quad (4.10)$$

However, the objective of the automated vehicle is to track the overall reference velocity  $v_c(t)$  (see Section 2.3.2), which most likely varies in time in a different way than the splines described by  $\mathcal{P}_{\min i, k}^{(1)}$  and  $\mathcal{P}_{\max i, k}^{(1)}$  do. Since the objective is to track the reference velocity  $v_c(t)$ , and these additional reference velocities are merely chosen to give the autonomous planner the option to deviate from reference velocity  $v_c(t)$ , the reference velocity  $v_c(t)$  should be included in one of the vectors  $\mathbf{w}_l$ ,  $l \in \{1, \dots, N_{\dot{s}}\}$ , such that a trajectory is generated corresponding with the overall reference velocity  $v_c(t)$ . Additionally, it should be noted that  $\mathcal{P}_{\min i, k}^{(1)}$  corresponds to a spline that eventually comes to a stop and starts reversing, as it represents a constant deceleration. This is undesirable and should therefore be excluded from the set of generated trajectories. To prevent the construction of reversing trajectories,  $\nu_{\min, k}$  and  $\nu_{\max, k}$  are defined as

$$\nu_{\min, k} = \max \left( [0 \quad \dots \quad 0 \quad 1] \mathcal{P}_{\min i, k}^{(1)}, 0 \right), \quad (4.11)$$

$$\nu_{\max, k} = [0 \quad \dots \quad 0 \quad 1] \mathcal{P}_{\max i, k}^{(1)}, \quad (4.12)$$

which are in turn used to define an array of terminal velocities  $\nu_l$  and the set of terminal velocities  $\mathcal{V}_{N_{\dot{s}}}$ , which include a trajectory towards  $v_c(t)$ , as

$$\nu_{l, k} = \begin{cases} v_{c, k}, & \text{if } l = \arg \min_{j \in \{1, \dots, N_{\dot{s}}\}} |v_{c, k} - v_j|, \\ v_l, & \text{otherwise,} \end{cases} \quad l \in \{1, \dots, N_{\dot{s}}\}, \quad (4.13)$$

$$v_{c, k} = v_c(t_{i, k} + T_i), \quad (4.14)$$

$$v_l = \nu_{\min, k} + (\nu_{\max, k} - \nu_{\min, k}) \frac{l-1}{N_{\dot{s}}-1}, \quad l \in \{1, \dots, N_{\dot{s}}\}, \quad (4.15)$$

$$\mathcal{V}_{N_{\dot{s}}} = \{\nu_1, \nu_2, \dots, \nu_{N_{\dot{s}}}\}. \quad (4.16)$$

The vector  $\mathbf{w}_{l, k}$  corresponding to  $\nu_{l, k}$  that is used in (4.3a), is constructed by

means of convex combinations of  $\mathbf{w}_{s|c,k}$ ,  $\mathbf{w}_{s|\nu_{\min,k}}$  and  $\mathbf{w}_{s|\nu_{\max,k}}$  as

$$\mathbf{w}_{s|l,k} = \begin{cases} \mathbf{w}_{s|c,k}, & \text{if } \nu_{l,k} = v_{c,k}, \\ \mathbf{w}_{s|c,k} \frac{\nu_{\max,k} - \nu_{l,k}}{\nu_{\max,k} - v_{c,k}} + \left(1 - \frac{\nu_{\max,k} - \nu_{l,k}}{\nu_{\max,k} - v_{c,k}}\right) \mathbf{w}_{s|\nu_{\max,k}}, & \text{if } \nu_{l,k} > v_{c,k}, \\ \mathbf{w}_{s|c,k} \frac{\nu_{l,k} - \nu_{\min,k}}{v_{c,k} - \nu_{\min,k}} + \left(1 - \frac{\nu_{l,k} - \nu_{\min,k}}{v_{c,k} - \nu_{\min,k}}\right) \mathbf{w}_{s|\nu_{\min,k}}, & \text{otherwise,} \end{cases} \quad (4.17)$$

where the vectors related to the reference velocity,  $\mathbf{w}_{s|c,k}$  and the minimum and maximum velocity,  $\mathbf{w}_{s|\nu_{\min,k}}$ ,  $\mathbf{w}_{s|\nu_{\max,k}}$  are given by

$$\mathbf{w}_{s|c,k} = Q_{\dot{s}} \begin{bmatrix} v_c(t_{i,k} + \mu_0^{(1)}) \\ \vdots \\ v_c(t_{i,k} + \mu_{n-1}^{(1)}) \\ \mathbf{0}_{n-1 \times 1} \\ \mathbf{0}_{n-2 \times 1} \end{bmatrix}, \quad (4.18)$$

$$\mathbf{w}_{s|\nu_{\min,k}} = Q_{\dot{s}} \nu_{\min,k} \begin{bmatrix} \mathbf{1}_{n \times 1} \\ \mathbf{0}_{n-1 \times 1} \\ \mathbf{0}_{n-2 \times 1} \end{bmatrix}, \quad \mathbf{w}_{s|\nu_{\max,k}} = Q_{\dot{s}} \nu_{\max,k} \begin{bmatrix} \mathbf{1}_{n \times 1} \\ \mathbf{0}_{n-1 \times 1} \\ \mathbf{0}_{n-2 \times 1} \end{bmatrix}. \quad (4.19)$$

Additionally, to generate trajectories with faster and slower transients, trajectories are also generated with modified cost function weights

$$J_{s|j,l,i,k} = \underbrace{\left\| \begin{bmatrix} Q_{\dot{s}} \Delta_{10} \\ j Q_{\ddot{s}} \Delta_{20} \\ j Q_{\dddot{s}} \Delta_{30} \end{bmatrix} \mathcal{P}_{i,k} - \mathbf{w}_{s|l,k} \right\|_2}_{\mathbf{V}_{s|j}}, \quad \begin{aligned} j &\in \{0, \dots, N_t - 1\}, \\ l &\in \{1, \dots, N_s\}. \end{aligned} \quad (4.20)$$

Note that the same could be achieved by introducing more weight, but by modifying the existing weights  $Q_{\dot{s}}$  and  $Q_{\ddot{s}}$ , the number of user changeable parameters is reduced for simplicity. An array of  $N_t \times N_s$  trajectories is then generated, by running Algorithm 4.2, with all combinations of  $\nu_{l,k}, l \in \{1, \dots, N_s\}$  and  $\mathbf{V}_{s|j}, j \in \{0, \dots, N_t - 1\}$ . The result can be seen in Figure 4.2, which also demonstrates the effectiveness of multiple cost function weights in generating a diverse set of trajectories. Note that by (4.20), for  $j = 0$  there is no cost on either jerk and acceleration. Additionally, for  $\nu_{l,k} = \nu_{\min,k}$  and  $\nu_{l,k} = \nu_{\max,k}$ , which are both contained in  $\mathcal{V}_{N_s}$  (4.16), respectively  $\mathbf{w}_{s|l,k} = \mathbf{w}_{s|\nu_{\min,k}}$ ,  $\mathbf{w}_{s|l,k} = \mathbf{w}_{s|\nu_{\max,k}}$  hold via (4.17). Hence, these trajectories attain the maximum acceleration as fast as possible for B-spline trajectories with continuity constraint on the acceleration. Despite being generated with *different* cost weights, each of the trajectories is assigned a cost with the *same* original (i.e.,  $j = 1$ ) cost  $\mathbf{V}_{s|1} = \mathbf{V}_s$  and vector related to reference velocity  $\mathbf{w}_{s|c,k}$ . This

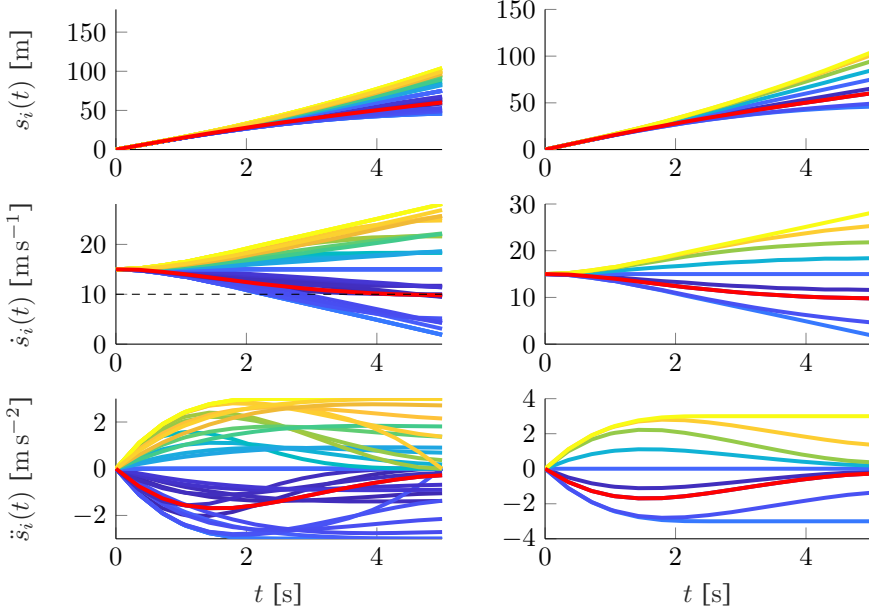


Figure 4.2: Trajectories single plan step: generated (cost: low — — high), selected (—) and reference velocity  $v_c(t)$  (- -). With  $n = 6$ ,  $p = 5$ ,  $N_s = 9$ ,  $\begin{bmatrix} s_i^{(2)} \\ \overline{s_i^{(2)}} \end{bmatrix} = [-3, 3] \text{ m s}^{-2}$ ,  $\mathcal{Q}_s = [1, 0, 1]$ . Left:  $\mathbf{V}_{s|j}, j \in \{1, \dots, 4\}$ , Right:  $\mathbf{V}_{s|j}, j = 1$ .

ensures that the trajectory that indeed follows the global reference  $v_c(t)$ , while minimizing (4.3a), gets selected. The other trajectories are only generated in case this trajectory gets invalidated, either due to a collision with obstacles or by violation of constraints.

Consecutive implementation of this planner can be seen in Figure 4.3, which shows the generated and selected trajectories for  $t_p = 1 \text{ s}$ . Note that typically the update rate would be higher, but for illustrative purposes a slower update rate was chosen. The figure demonstrates that the selected trajectories are again practically temporally consistent, as the red line of the selected trajectory in each of the plan steps nearly overlaps with the other red lines.

### 4.1.3 Lateral Trajectory Generation

Vehicle following is only performed in the longitudinal direction (i.e., coordinate  $s(t)$ ). However, lateral trajectories are still required successful navigation of traffic by an automated vehicle, in both cooperative as well as autonomous operation. Lateral trajectories are constructed with the objective to perform lane



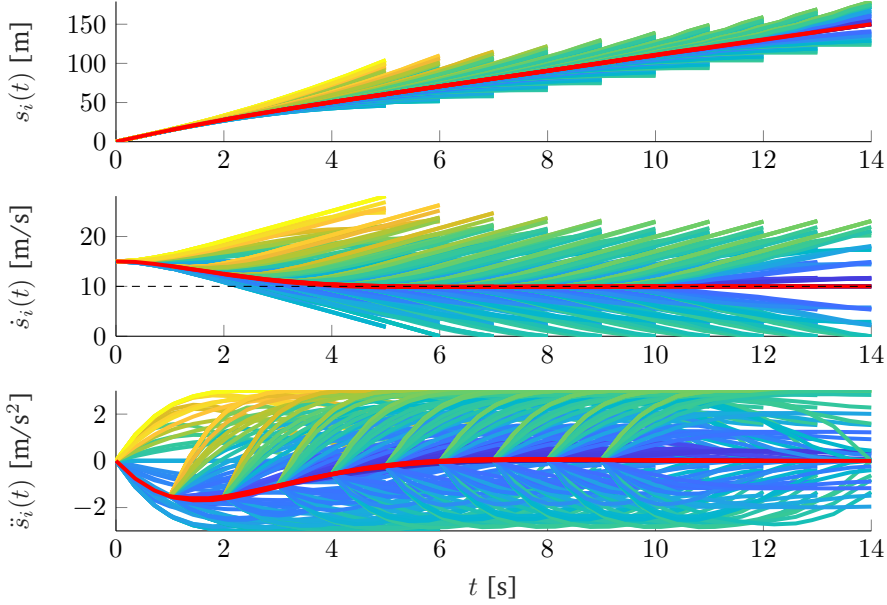


Figure 4.3: Trajectories: generated (cost: low ■ ■ high), selected (—) and reference velocity  $v_c(t)$  (---). With  $n = 6$ ,  $p = 5$ ,  $t_p = 1$  s,  $N_s = 9$ ,  $N_t = 4$ ,  $\begin{bmatrix} s_i^{(2)} \\ \overline{s_i^{(2)}} \end{bmatrix} = [-3, 3]$  m s<sup>-2</sup>,  $\mathbf{Q}_s = [1, 0, 1]$ .

centering regardless of the motion of the preceding vehicle. An option for the construction of lateral trajectories by means of B-splines is presented next. For lateral trajectories a similar but slightly different approach is taken. The reference for lateral trajectories  $\ell_c(t)$  is typically equal to zero, which would result in lane centering (see also Figure 2.2). If the top layer route planner indicates that a different lane should be driven to for example take the offramp of a highway or pre-sort when approaching a multi-lane traffic light. The lateral trajectories can also be represented by means of B-splines, although this is not strictly required for the framework of the autonomous and cooperative vehicle that is used here. Denote the  $n + 1$  control points for the lateral B-spline trajectory of degree  $p$  by

$$\mathcal{R}_{i,k} = [R_{0,i,k} \quad R_{1,i,k} \quad \dots \quad R_{n,i,k}]^T \quad (4.21)$$

Note that it is not required that longitudinal and lateral trajectories are of the same degree, or have the same number of control points. However, it is convenient as this allows to use the same basis functions  $\mathcal{B}_{n,p}(t)$ , which speeds up implementation on the real time platform, as basis functions only need to be evaluated once, and can then be simply post-multiplied by the coefficients of all

lateral and longitudinal trajectories.

For lateral B-spline trajectories the following general cost function is minimized

$$J_{\ell|i,k} = \left\| \underbrace{\begin{bmatrix} Q_\ell \mathbf{I}_{n+1 \times n+1} \\ Q_{\dot{\ell}} \Delta_{20} \\ Q_{\ddot{\ell}} \Delta_{30} \\ Q_{\dddot{\ell}} \Delta_{40} \end{bmatrix}}_{\mathbf{V}_\ell} \mathbf{R}_{i,k} - Q_\ell \underbrace{\begin{bmatrix} \ell_c(\mu_0) \\ \vdots \\ \ell_c(\mu_n) \\ \mathbf{0}_{n \times 1} \\ \mathbf{0}_{n-1 \times 1} \\ \mathbf{0}_{n-2 \times 1} \end{bmatrix}}_{\mathbf{w}_{\ell|c,k}} \right\|_2, \quad (4.22a)$$

$$\text{s.t.} \quad \underbrace{\begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \Delta_{10} \\ \Delta_{20} \end{bmatrix}}_{\mathbf{Z}_\ell} \mathbf{R}_{i,k} = \underbrace{\Lambda_i(t_{i,k})}_{\mathbf{q}_\ell}, \quad (4.22b)$$

where,  $Q_\ell$  represents the weight associated with the difference of lateral position with respect to the desired lateral position, and  $Q_{\dot{\ell}}$ ,  $Q_{\ddot{\ell}}$ ,  $Q_{\dddot{\ell}}$  represent the cost associated to lateral velocity, acceleration and jerk respectively. Similar to the various reference velocities that are used to generate multiple longitudinal trajectories, various lateral reference positions are considered to generate multiple swerves. This allows the vehicle to deviate from the desired lane centerline, as also discussed in Section 2.1.4. The range in which these lateral trajectories can vary is determined by right road bound  $\ell_R$ , left road bound  $\ell_L$  and specified acceleration bound  $\overline{\ell^{(2)}}$ . To find a range of feasible terminal lateral positions, the equivalent of (4.4) for lateral trajectories is used

$$\mathbf{R}_i^{(d-1)} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ & \Delta_{d(d-1)} & & \end{bmatrix}^{-1} \begin{bmatrix} \ell^{(1)}(t_{i,k}) \\ \mathbf{R}_i^{(d)} \end{bmatrix}, \quad (4.23)$$

which can be used to in combination with the control points corresponding to the maximum left swerve spline and maximum right swerve spline that satisfy the acceleration constraint  $|\ddot{\ell}(t)| \leq \overline{\ell^{(2)}}$  of plan step  $k$  are defined as

$$\mathbf{R}_{Ri,k}^{(2)} = [\ddot{\ell}(t_{i,k}), -\overline{\ell^{(2)}}, \dots, -\overline{\ell^{(2)}}]^\top, \quad (4.24)$$

$$\mathbf{R}_{Li,k}^{(2)} = [\ddot{\ell}(t_{i,k}), \overline{\ell^{(2)}}, \dots, \overline{\ell^{(2)}}]^\top. \quad (4.25)$$

By combining these with the initial condition for the lateral velocity  $\dot{\ell}(t_{i,k})$ , the control points of B-spline trajectories with the lowest and highest velocity respectively, that satisfy acceleration bound  $|\ddot{\ell}(t)| \leq \overline{\ell^{(2)}}$  can be found analogous

to (4.7) via

$$\begin{bmatrix} \mathcal{R}_{R,i,k}^{(1)} & \mathcal{R}_{L,i,k}^{(1)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ & \Delta_{21} & & \end{bmatrix}^{-1} \begin{bmatrix} \dot{\ell}(t_{i,k}) & \dot{\ell}(t_{i,k}) \\ \mathcal{R}_{R,i,k}^{(2)} & \mathcal{R}_{L,i,k}^{(2)} \end{bmatrix}, \quad (4.26)$$

$$\begin{bmatrix} \mathcal{R}_{R,i,k} & \mathcal{R}_{L,i,k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ & \Delta_{10} & & \end{bmatrix}^{-1} \begin{bmatrix} \ell(t_{i,k}) & \ell(t_{i,k}) \\ \mathcal{R}_{R,i,k}^{(1)} & \mathcal{R}_{L,i,k}^{(1)} \end{bmatrix}. \quad (4.27)$$

Note that this acceleration is not the lateral acceleration of the vehicle in Cartesian coordinates, but instead the relative acceleration in the direction perpendicular to the road centerline. As a result of curvature of the reference line, the acceleration in the vehicle frame can still be higher than the acceleration bound. Hence, acceleration constraints in the Cartesian coordinates are checked afterwards, during the feasibility check of all generated trajectories.  $N_\ell$  lateral trajectories are generated for each of the  $(N_s + N_s) N_t$  longitudinal trajectories. These  $N_\ell$  trajectories are distributed between right road bound  $\ell_R$  and left road bound  $\ell_L$ . Additionally, the maximum and minimum allowed lateral deviation computed by (4.27) is considered via

$$\ell_{\min,k} = \max(\ell_R, [0 \ \dots \ 0 \ 1] \mathcal{R}_{R,i,k}), \quad (4.28)$$

$$\ell_{\max,k} = \min(\ell_L, [0 \ \dots \ 0 \ 1] \mathcal{R}_{L,i,k}). \quad (4.29)$$

Then lateral positions  $\ell_{l,k}$  are defined as

$$\ell_{l,k} = \begin{cases} \ell_{c,k}, & \text{if } l = \arg \min_{j \in \{1, \dots, N_\ell\}} |\ell_{c,k} - R_j|, \\ R_l, & \text{otherwise,} \end{cases} \quad l \in \{1, \dots, N_\ell\}, \quad (4.30)$$

$$R_l = \ell_{\min,k} + (\ell_{\max,k} - \ell_{\min,k}) \frac{l-1}{N_\ell-1}, \quad l \in \{1, \dots, N_\ell\} \quad (4.31)$$

$$\ell_{c,k} = \ell_c(t_{i,k} + T_i). \quad (4.32)$$

With these lateral positions  $\ell_{l,k}, l \in \{1, \dots, N_\ell\}$ , trajectories are generated by means of minimization of the following cost function

$$J_{\ell|j,l,i,k} = \left\| \underbrace{\begin{bmatrix} Q_\ell \mathbf{I}_{n+1 \times n+1} \\ jQ_\ell \Delta_{20} \\ jQ_\ell \Delta_{30} \\ jQ_\ell \Delta_{40} \end{bmatrix}}_{\mathbf{V}_{\ell,j}} \mathcal{R}_{i,k} - Q_\ell \underbrace{\begin{bmatrix} \ell_{l,k} \mathbf{1}_{n+1 \times 1} \\ \mathbf{0}_{n \times 1} \\ \mathbf{0}_{n-1 \times 1} \\ \mathbf{0}_{n-2 \times 1} \end{bmatrix}}_{\mathbf{w}_{\ell|l,k}} \right\|_2, \quad \begin{matrix} j \in \{0, \dots, N_t-1\}, \\ l \in \{1, \dots, N_\ell\}. \end{matrix} \quad (4.33)$$

Note that this cost function to generate trajectories differs from the cost function (4.22a), which is used to assign a cost. Trajectories minimizing  $J_{\ell|j,l,i,k}, j \in \{1, \dots, 5\}, l \in \{1, \dots, 9\}$  are shown in Figure 4.4. It can be seen that the lateral trajectories, are very diverse, such that it is likely a feasible

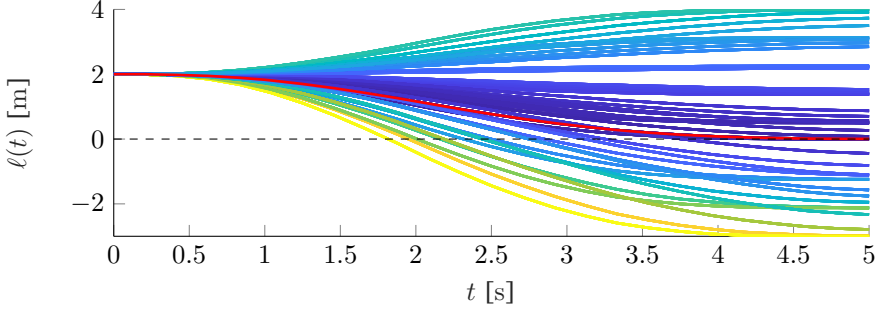


Figure 4.4: Trajectories single plan step: generated (cost: low █ █ █ █ █ high), selected (—) and reference position  $\ell_c(t)$  (- - -). With  $n = 6$ ,  $p = 5$ ,  $\ell_L = 4\text{m}$ ,  $\ell_R = -3\text{m}$ , Lateral weights  $Q_\ell = 2$ ,  $Q_{\dot{\ell}} = 1$ ,  $Q_{\ddot{\ell}} = Q_{\ddot{\ell}} = 0$ .

trajectory is generated. Note that only single swerves (e.g., such as lane changes) are generated by means of this minimization approach, even though the B-spline could be used to describe lateral trajectories with multiple lateral swerves (e.g., overtaking by first moving to the left, and moving back to the right when a vehicle is passed), especially for higher values of  $n - p$ . The construction of lateral trajectories was however, not the focus of this work, and is only included to demonstrate some basic functionality. The use of B-splines for lateral trajectories yield an extendible approach: lateral trajectories can be constructed that do include multiple swerves, which has advantages as explained in [21], such as more feasible trajectories for overtaking with on-coming traffic. Additionally, it should be noted that for the cooperative trajectory that is generated by means of the approaches explained in Section 3.3,  $N_t \times N_\ell$  lateral trajectories are generated. This is done such that a cooperating vehicle is still capable of navigating around static obstacles that are positioned near the road centerline, for example. The lateral components of the cooperative trajectories are thus relative to the road, rather than to the preceding vehicle. This prevents issues with corner cutting as described in [135] [136].

## 4.2 Combined Autonomous and Cooperative Trajectory Planning

With both the lateral as well as the longitudinal trajectories generated, the autonomous vehicle is capable of deviating from the reference path and velocity. However, the complete functionality of the automated vehicle includes both taking into account obstacles and other road users, as well as selecting either cooperative or autonomous driving. This is explained next. First, consider the

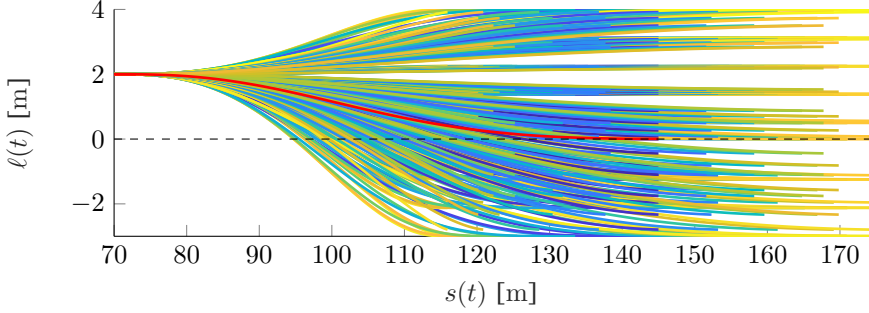


Figure 4.5: Trajectories single plan step: generated (cost: low — high —), selected (—) and reference position  $\ell_c(t)$  (- -). With  $n = 6$ ,  $p = 5$ ,  $\ell_L = 4\text{m}$ ,  $\ell_R = -3\text{m}$ , Longitudinal  $\mathcal{Q}_s = [25, 0, 25]$ , Lateral weights  $\mathcal{Q}_\ell = [1, 0.5, 0, 0]$ .

combined cost function

$$J_{i,k} = \frac{1}{v_c(t_{i,k} + T_i)} \left\| \mathbf{V}_s \mathcal{P}_{i,k} - Q_s \mathbf{w}_{s|c,k} \right\|_2 + \left\| \mathbf{V}_\ell \mathcal{R}_{i,k} - Q_\ell \mathbf{w}_{\ell|c,k} \right\|_2, \quad (4.34)$$

where  $\mathbf{w}_{s|c,k}$  is according to (4.18) and  $\mathbf{w}_{\ell|c,k}$  according to (4.22a). Note that the scaling factor  $v_c(t_{i,k} + T_i)$  is used to make the longitudinal cost relative to the reference velocity. This makes sure that overtaking is possible at all reference velocities. This cost function is used to assign a cost to the combined lateral and longitudinal trajectories. Figure 4.5 shows the generated combined autonomous trajectories in the spatial frame, with an initial forward velocity of  $\dot{s}_i(0) = v_c = 10 \text{ m s}^{-1}$ ,  $[N_{\dot{s}}, N_\ell, N_t] = [9, 9, 5]$ , such that a total of 405 trajectories are generated. Cost function parameters are chosen as  $\mathcal{Q}_s = [Q_{\dot{s}}, Q_{\ddot{s}}, Q_{\ddot{s}}] = [25, 0, 25]$  and  $\mathcal{Q}_\ell = [Q_\ell, Q_{\dot{\ell}}, Q_{\ddot{\ell}}, Q_{\ddot{\ell}}] = [1, 0.5, 0, 0]$ .

Clearly, a very diverse set of trajectories is generated that spans the entire range of desired lateral trajectories from  $\ell_L$  to  $\ell_R$ . However, the used cost function does not include all of the desired behavior. For example, trajectories that only nearly miss obstacles and other road vehicles, are still labelled as ‘collision’ free, whereas in practice a safety margin is desired. Additionally, in continental Europe, the convention is to overtake vehicle on the left, rather than on the right.

### Combined Lateral and Longitudinal Planning with Obstacles

To deal with obstacles and other road users in the cost function, an additional cost similar to the cost used in [94] and [95] is used. A two dimensional Gaussian

function is used defined by

$$G(x, y, \theta, \sigma_x, \sigma_y) = \exp \left[ - \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} \frac{\cos^2 \theta}{2\sigma_x^2} + \frac{\sin^2 \theta}{2\sigma_y^2} & \frac{\sin(2\theta)}{4\sigma_x^2} - \frac{\sin(2\theta)}{4\sigma_y^2} \\ \frac{\sin(2\theta)}{4\sigma_x^2} - \frac{\sin(2\theta)}{4\sigma_y^2} & \frac{\sin^2 \theta}{2\sigma_x^2} + \frac{\cos^2 \theta}{2\sigma_y^2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \right], \quad (4.35)$$

which has a shape determined by parameter  $\sigma_x$  and  $\sigma_y$  and  $\theta$  rotates this shape with respect to the primary coordinates  $x$  and  $y$ . The function is then evaluated with the relative distance to all objects in the Frenet frame, denoted by the set  $\mathcal{O}$ , and all other road users, denoted by the set  $\mathcal{V}$ , and  $\mathcal{X} = \mathcal{O} \cup \mathcal{V}$ ,

$$J_{o,j} = Q_o \int_{t_{i,k}}^{t_{i,k}+T_i} G(\delta_{s_j}(t), \delta_{\ell_j}(t), \theta_j, \sigma_{s,j}, \sigma_{\ell,j}) dt, \quad j \in \mathcal{O}, \quad (4.36)$$

$$J_{u,j} = Q_u \int_{t_{i,k}}^{t_{i,k}+T_i} G(\delta_{s_j}(t), \delta_{\ell_j}(t), \theta_j, \sigma_{s,j}, \sigma_{\ell,j}) dt, \quad j \in \mathcal{V}. \quad (4.37)$$

The shape of the Gaussian function is determined by parameters  $\sigma_{s,j}, \sigma_{\ell,j}$  and weights  $Q_u$  and  $Q_o$  for respectively the other road users and obstacles.  $\delta_{\ell_j}(t)$  and  $\delta_{s_j}(t)$  are the distances in the Frenet frame of the edge of the host vehicle, to the edge of the obstacle computed as,

$$\delta_{s_j}(t) = \max \left( \left| \mathcal{B}_{n,p}(t) \mathcal{P}_{i,k} + \frac{L_i - L_j}{2} - s_j(t) \right| - \frac{L_j + L_i}{2}, 0 \right), \quad j \in \mathcal{X}, \quad (4.38)$$

$$\delta_{\ell_j}(t) = \begin{cases} 0 & \text{if: } j \in \mathcal{V} \quad \wedge \quad \mathcal{B}_{n,p}(t) \mathcal{R}_{i,k} < \ell_j(t) \\ & \quad \wedge \quad \mathcal{B}_{n-1,p-1}(t) \Delta_{10} \mathcal{P}_{i,k} > \dot{s}_j(t) \\ \max \left( |\mathcal{B}_{n,p}(t) \mathcal{R}_{i,k} - \ell_j(t)| - \frac{W_j + W_i}{2}, 0 \right) & \text{otherwise} \end{cases} \quad j \in \mathcal{X}, \quad (4.39)$$

where  $L_i$  and  $W_i$  represent the length and width of the host vehicle, and  $L_j$  and  $W_j$  the length and width of obstacle  $j$ . Recall that the progression  $s_i(t)$  of vehicle  $i$  defines the position of the rear of each vehicle, as can be seen in Figure 3.1. Hence, the longitudinal distance between the edges of the vehicles, described by (4.38), needs to be corrected for the difference in vehicle length, unlike the lateral distance (4.39). Note that for other road users (i.e.,  $j \in \mathcal{V}$ ), the lateral distance is set to 0, if the trajectory of the host vehicle is to the right of the other road user, and the host vehicle is driving faster then the other road user. This ensures that a large cost is assigned if the trajectory passes the obstacle on the right. Note that this should not be applied in cases where the law does not prevent ‘overtaking’ on the right, such as for example when a vehicle is at standstill for a red light at a traffic light, while the lane of the host vehicle has a green light. A visualization of the Gaussian function is given in Figure 4.6. The left figure shows a simple

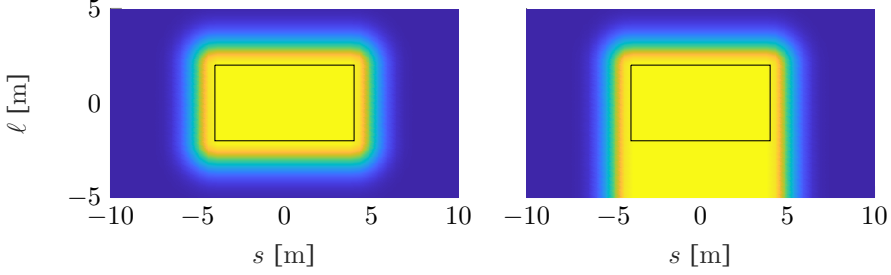



Figure 4.6: Gaussian cost function (cost: low  high), for  $\sigma_s^2 = \sigma_\ell^2 = 1$ , left:  $j \in \mathcal{O}$ , right:  $\mathcal{B}_{n,p}(t)\mathcal{P}_{i,k} > \dot{s}_j(t)$ ,  $j \in \mathcal{V}$

example for an obstacle (i.e.,  $j \in \mathcal{O}$ ), while the right figure shows the modified Gaussian function, in case the host vehicle drives faster than the other road user (i.e.,  $j \in \mathcal{V}$ ), in which the cost associated to the vehicle extends to the right side of the vehicle.

By combining (4.36) and (4.37) with the cost function without obstacles, (4.34), the following cost function is found

$$J_{i,k} = \frac{1}{v_c(t_{i,k} + T_i)} \left\| \mathbf{V}_s \mathcal{P}_{i,k} - Q_s \mathbf{w}_{s|c,k} \right\|_2 + \left\| \mathbf{V}_\ell \mathcal{R}_{i,k} - Q_\ell \mathbf{w}_{\ell|c,k} \right\|_2 + \sum_{o \in \mathcal{O}} J_{o,o} + \sum_{u \in \mathcal{V}} J_{u,u}. \quad (4.40)$$

A scenario is simulated using this cost function, with B-spline parameters  $n = 6$ ,  $p = 5$ ,  $T_i = 5$  s and trajectory parameters  $[N_s, N_\ell, N_t] = [9, 9, 5]$ , and that are updated at  $t_p = 1$  s. The cost function is evaluated with longitudinal weights  $\mathcal{Q}_s = [Q_{\dot{s}}, Q_{\ddot{s}}, Q_{\ddot{s}}] = [25, 0, 25]$  and lateral weights  $\mathcal{Q}_\ell = [Q_\ell, Q_{\dot{\ell}}, Q_{\ddot{\ell}}, Q_{\ddot{\ell}}] = [1, 0.5, 0, 0]$ , with the shape of the obstacle cost function determined by  $\sigma_s^2 = \sigma_\ell^2 = 0.15$ ,  $Q_u = Q_o = 5$ . The host vehicle using this cost function then result in the behavior demonstrated in Figure 4.7, which illustrates how the host vehicle first drives around a stationary vehicle and then drives around an obstacle. Note that the cost function can be improved further; to fine-tune the behavior of the vehicle.

### Switching between Cooperative and Autonomous Trajectory Planning

Finally, each vehicle equipped with the presented planning framework plans both autonomous as well as cooperative trajectories simultaneously. A cooperative trajectory can be generated for the nearest vehicle in front of the host vehicle in each lane. The set of next vehicles is denoted  $\mathcal{W}$ , which is contained in the set of other road users  $\mathcal{W} \subseteq \mathcal{V}$ . As a result, each plan step  $(N_s + N_\ell + |\mathcal{W}|)N_\ell N_t$  trajectories are generated, where  $|\mathcal{W}|$  denotes the cardinality or size of the set  $\mathcal{W}$ . Note that the trajectories in Chapter 3 can be generated for a multitude of

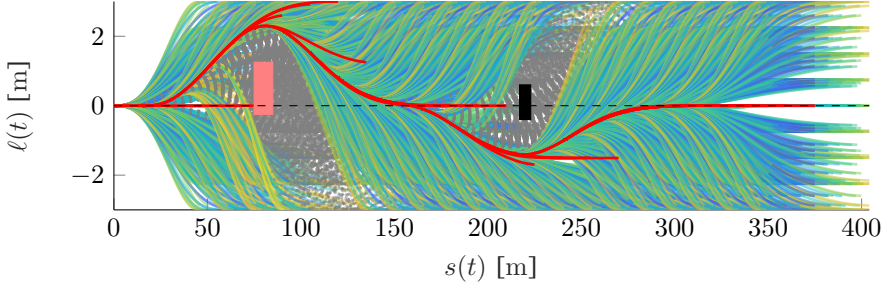


Figure 4.7: Interaction with road user (■) and obstacle (■). Generated trajectories (cost: low — high —), collision (····) selected (—) and reference position  $\ell_c(t)$  (---).

candidate vehicles, with which platooning can be initiated. This can be useful in scenarios with multiple lanes for example. From this set of generated trajectories, a single trajectory is selected for execution. Each of the generated trajectories is assigned a cost using the same cost function. To allow for cooperative trajectories to be selected, a discount is attributed to them. This discount compensates for the additional cost that results from driving at a different velocity than the desired velocity (e.g.,  $\frac{1}{v_c(t_{i,k}+T_i)} \|\mathbf{V}_s \mathcal{P}_{i,k} - Q_s \mathbf{w}_{s|c,k}\|_2$ ). With the discount for cooperative driving, the following combined cost function is used

$$J_{i,k} = \frac{1}{v_c(t_{i,k}+T_i)} \|\mathbf{V}_s \mathcal{P}_{i,k} - Q_s \mathbf{w}_{s|c,k}\|_2 + \|\mathbf{V}_\ell \mathcal{R}_{i,k} - Q_\ell \mathbf{w}_{\ell|c,k}\|_2 + \sum_{o \in \mathcal{O}} J_{o,o} + \sum_{u \in \mathcal{V}} J_{u,u} - \sum_{w \in \mathcal{W}} J_{p,w}, \quad (4.41)$$

$$J_{p,j} = Q_p \int_{t_{i,k}}^{t_{i,k}+T_i} G(e_{i,j}(t), 0, 0, \sigma_{p_s,j}, \sigma_{p_\ell,j}) dt, \quad j \in \mathcal{W}, \quad (4.42)$$

$$e_{i,j}(t) = s_j(t) - (s_i(t) + L_i) - c_{r,i}(t) - \underline{h}_i \dot{s}_i(t), \quad (4.43)$$

where  $\sigma_{p_s,j}$  determines the shape of the ‘platooning discount’ area, for discount  $J_{p,j}$  with discount parameter  $Q_p$  and the spacing error is defined according to the time-varying spacing policy of Section 3.4. The discount area spans the entire width of the road due to the lateral component in the cost being set to zero, as is illustrated in Figure 4.8. Therefore, parameter  $\sigma_{p_\ell,j}$  can be set to one arbitrarily, as it does not impact the shape of the discount area. Note that since the lateral position is not important the Gaussian function is in fact one dimensional. Because the discount area spans the entire width of the road, it allows a platoon to overtake a slower moving vehicle one by one, and prevent the whole platoon from switching lanes simultaneously, which does not scale well to larger platoons. If the platooning discount parameter  $Q_p$  is sufficiently



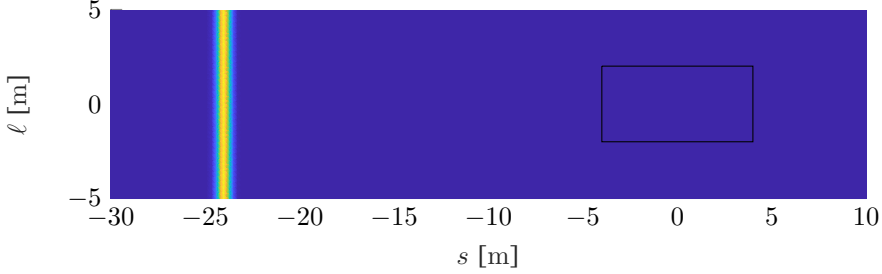
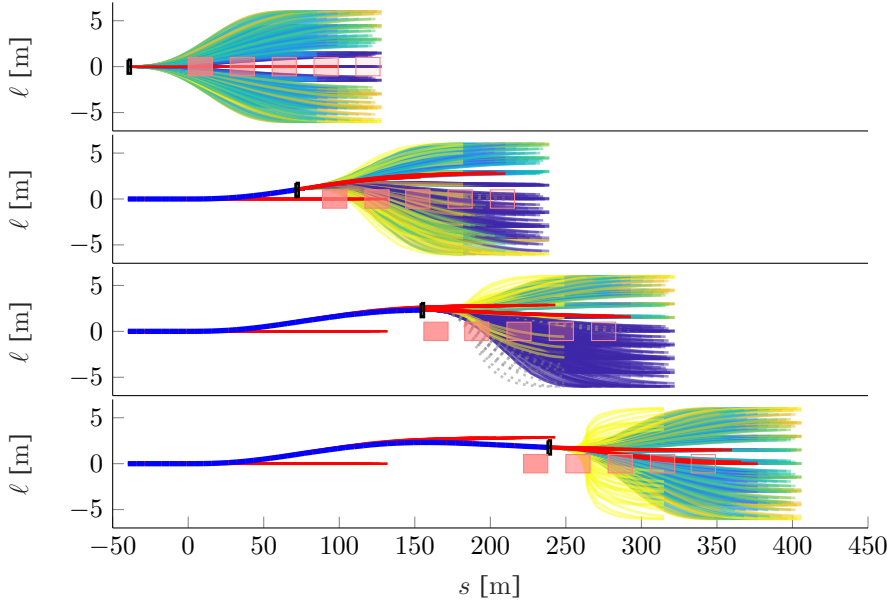


Figure 4.8: Gaussian discount function for cooperation (cost: low ■ high), for  $\sigma_{ps,j} = \frac{1}{4}$ . Center of discount area is located at  $d_{r,i}(t)$  of (3.4) of rear bumper of other road user.

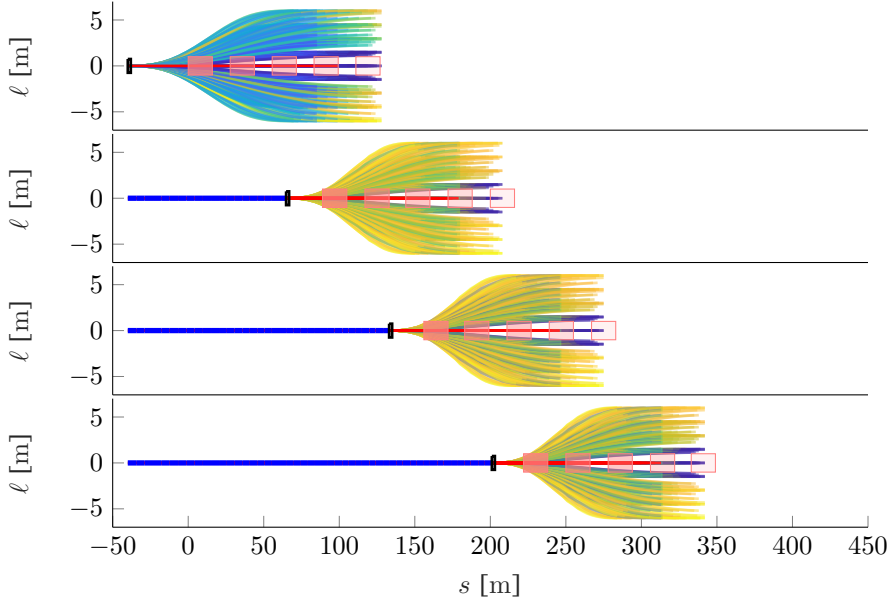
large for a given velocity of the preceding vehicle, the cooperative trajectories are selected in favour of the trajectories that overtake the slower preceding vehicle.

A simulation study is used to demonstrate that parameter  $Q_p$  can be used to control the behavior of the host vehicle in terms of whether or not to cooperate with the vehicle in front. In both cases, the following parameters are used. The trajectories are constructed by means of B-splines with  $n = 6$  and  $p = 5$ ,  $T_i = 5$  s, and are updated each  $t_p = 0.2$  s. The trajectory parameters are chosen as  $[N_s, N_\ell, N_t] = [9, 9, 5]$ . The cost function is evaluated with longitudinal weights  $\mathbf{Q}_s = [Q_{\dot{s}}, Q_{\ddot{s}}, Q_{\ddot{s}}] = [25, 0, 25]$  and lateral weights  $\mathbf{Q}_\ell = [Q_\ell, Q_{\dot{\ell}}, Q_{\ddot{\ell}}, Q_{\ddot{\ell}}] = [1, 0.5, 0, 0]$ , with the shape of the obstacle cost function determined by  $\sigma_s^2 = \sigma_\ell^2 = 0.15$ ,  $Q_u = Q_o = 5$ . The shape of the discount area is determined by means of  $\sigma_{ps,j} = 0.5$  m. A vehicle, which represents a commercial truck, with  $W_0 = 2$  m and  $L_0 = 16$  m, is simulated driving at a constant velocity with a reference velocity of  $v_c = 80$  km h<sup>-1</sup>. A second vehicle representing a regular vehicle with  $W_1 = 1.5$  m and  $L_1 = 2$  m and a higher reference velocity of  $v_c = 100$  km h<sup>-1</sup> is approaches this commercial truck at this higher reference velocity from a distance of 40 m. The results of two such simulations are shown in Figure 4.9, for two values of  $Q_p$ . The second vehicle has platooning parameters with a gap size  $\underline{c}_1 = 5$  m and a time gap of  $\underline{h}_1 = 0.5$  s. The two cases shown in the figure illustrate one case in which the second vehicle starts platooning, and one case in which the vehicle overtakes the slower truck. In the plots, the outline of the truck is plotted for varying stages of the planning horizon, varying from  $t_{i,k}$  to  $t_{i,k} + T_i$ , with decreasing opacity.

A second scenario, where a steady state platoon approaches a truck, which is driving at a slower steady state velocity, is simulated with identical parameters as before and  $Q_p = 0$ . The results are summarized in Figure 4.10, which depicts the executed path of the lead vehicle in the platoon in black, and blue for the trailing vehicle in the platoon. It can be seen that the platoon overtakes the slower moving truck, where each vehicle determines when to move to the adjacent lane independently from the other vehicles in the platoon, while in the curvilinear



(a) Overtaking is selected,  $Q_p = 0$ ,  $\sigma_{p_{s,j}} = 0.5$ , at times  $t = [0, 4, 7, 10]$  s



(b) Platooning is selected,  $Q_p = 20$ ,  $\sigma_{p_{s,j}} = 0.5$ , at times  $t = [0, 4, 7, 10]$  s

Figure 4.9: Platooning switching with Road user (■). Generated trajectories (cost: low (blue), high (yellow)), collision (····), selected (—), executed (—).

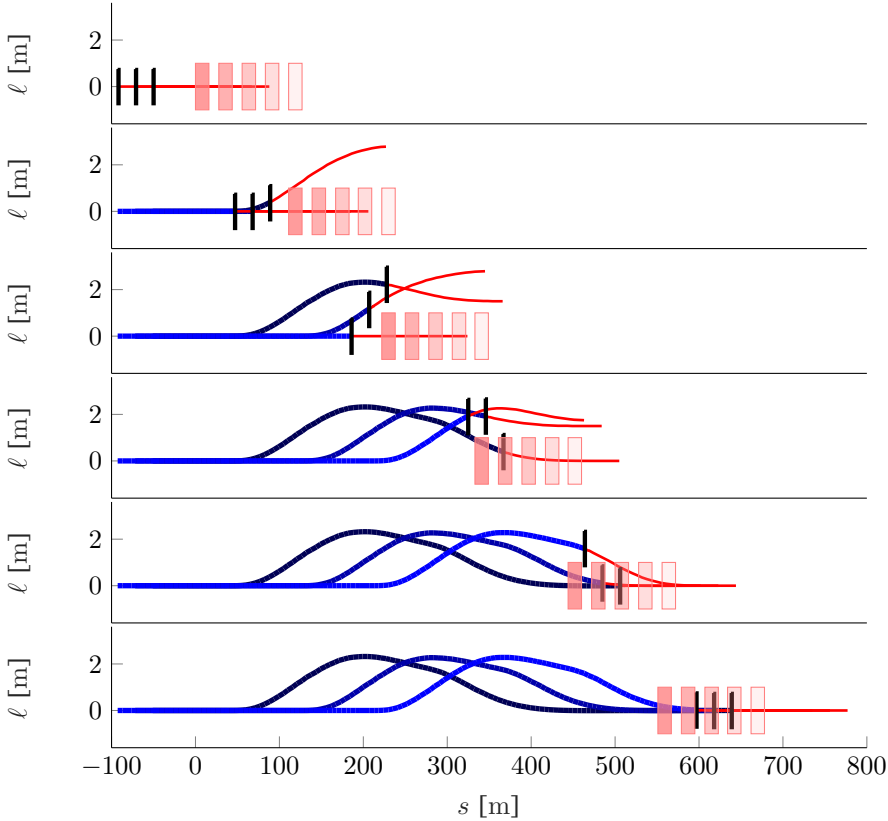


Figure 4.10: Platoon overtaking a Truck (■). Selected trajectories(—), executed first platoon vehicle (—), last vehicle (—), at times  $t = [0, 5, 10, 15, 20, 25]$  s

coordinate, the platoon is driving in steady state according to the spacing policy (3.4). Note that if a higher value for  $Q_p$  was chosen for the lead vehicle of the platoon, the platoon would have started platooning with the truck instead of overtaking. If then also a vehicle further upstream (e.g., for  $i = N$ ), has a sufficiently low value of  $Q_p$ , the platoon would break up, and all vehicles ( $i \geq N$ ) behind would follow the vehicle with (e.g.,  $i = N$ ) in overtaking the vehicles in front. This occurs because a cooperative trajectory then becomes available that drives closer to the desired velocity. This, in combination with the switching behavior of Figure 4.9 illustrates the effectiveness of the planner in the ad-hoc breaking and forming of platoons.

## 4.3 Summary

The framework for cooperative trajectory planning by means of B-splines requires that the lead vehicle of the platoon also utilizes B-spline trajectories. The construction of these autonomous B-spline trajectories presented first. Trajectories are generated by means of minimization of a cost function. By adjusting the cost parameters, a diverse set of trajectories is generated, which includes both highly energetic trajectories as well as smooth trajectories. A single cost function is then used to select the trajectory for execution. Unlike the polynomial planner of [18], the higher energetic (and uncomfortable) trajectories are not required to be selected for temporal consistency. Both longitudinal, as well as lateral trajectories are constructed. Although currently not implemented, the use of B-spline lateral trajectories would allow a single trajectory to contain multiple lateral swerves. This could benefit temporal consistency, as for an overtaking manoeuvre both the departure to the overtake lane, as well as the departure from the overtake lane could be described using a single trajectory.

Additionally, a framework for combined cooperative and autonomous driving is presented. A single parameter can be used to tune the behavior of the host vehicle, and determine for what velocity difference the cooperative platooning trajectory is selected and when the autonomous trajectory that overtakes the vehicle in front is selected. In summary, a framework for B-spline trajectory planning is presented, for both autonomous as well as cooperative trajectories. By equipping all vehicles with the same framework, all vehicles generate both autonomous as well as cooperative trajectories simultaneously, while communicating their selected trajectory for execution. Each vehicle in the string can determine whether or not remaining in the platoon is beneficial, or if it is more suitable to overtake the vehicle in front. In this framework, lateral behavior is never cooperative, but defined in terms of lateral position on the road. This prevents issues with corner cutting traditionally seen in lateral vehicle following.



# Chapter 5

## String Stability Analysis

The cooperative trajectory planning method presented in Chapter 3, is intended to be string stable. A numerical study is performed to examine for which combinations of time gap and planning delay, the string demonstrated string stable behavior in the considered scenarios. However, a more rigorous mathematical analysis is required to formally analyse string-stability. This chapter presents this mathematical analysis.

First, in Section 5.2 the hybrid dynamics of the trajectory planning method are discussed, which combine the discrete planning updates for time steps  $k$ , with the continuous time description of the B-spline trajectories. For the purpose of simplicity, the analysis is limited to the time-invariant spacing policy of (3.4). It is demonstrated that the consecutive B-spline coordinates can actually be written as states of a linear time invariant (LTI) discrete time system. The B-spline control points are hence used as state description of the discrete time system. Furthermore, the control points completely define the behavior of the continuous time signals (state trajectories can be expressed linearly in the control points). Subsequently, the  $\mathcal{L}_2$  signal norm of the relevant continuous time states can be expressed quadratically in the control points of the B-spline trajectories.

This is utilized to formally define  $\mathcal{L}_2$  string stability, for a broader class of trajectory planning algorithms. First, in Section 5.3, string stability is considered in the frequency domain, for arbitrary inputs. It is demonstrated that in fact the planning algorithm is string unstable for arbitrary inputs of the lead vehicle. However, this might very well correspond to scenarios that are not relevant for practical use. Therefore, a second analysis is presented in Section 5.4 for lead vehicles that are approximately temporal consistent. It is shown that for arbitrary changes in velocity for the lead vehicle, the planning algorithm of the following vehicles does result in string stable behavior, for platoons following the lead vehicle of Section 3.3.4 as well as the lead vehicle described in Chapter 4.

Finally, delays are also taken into consideration for the approximately temporal consistent lead vehicle.

## 5.1 $\mathcal{L}_2$ -String Stability

First, the string stability criterion is formalized. A detailed definition for  $\mathcal{L}_p$  string stability is provided in [30] and [3], which define  $\mathcal{L}_p$  string stability as follows. Given an interconnected system

$$\dot{\mathbf{x}}_0 = f_r(\mathbf{x}_0, \mathbf{u}_r), \quad (5.1a)$$

$$\dot{\mathbf{x}}_i = f_i(\mathbf{x}_i, \mathbf{x}_{i-1}), \quad i \in \mathcal{V}_m, \quad (5.1b)$$

$$\mathbf{y}_i = g(\mathbf{x}_i), \quad i \in \mathcal{V}_m, \quad (5.1c)$$

with  $\mathbf{x}_i \in \mathbb{R}^{q_x}$ ,  $\mathbf{y}_i \in \mathbb{R}^{q_y}$ , the states and outputs of the subsystems, with  $i \in \mathcal{V}_m$ ,  $\mathcal{V}_m = \{1, \dots, m\}$  and  $\mathbf{u}_r \in \mathbb{R}^{q_u}$  the external input to the first system. Let  $\mathbf{x} = [\mathbf{x}_0^\top \ \mathbf{x}_1^\top \ \dots \ \mathbf{x}_m^\top]^\top$  with equilibrium solution  $\check{\mathbf{x}} = [\check{\mathbf{x}}_0^\top \ \check{\mathbf{x}}_1^\top \ \dots \ \check{\mathbf{x}}_m^\top]^\top$  for  $\mathbf{u}_r = \mathbf{0}$ . The interconnected system is then  $\mathcal{L}_p$  string-stable if there exist class  $\mathcal{K}$  functions  $\alpha$  and  $\beta$  (i.e., functions that are continuous, strictly increasing and have  $\alpha(0) = 0$ ,  $\beta(0) = 0$ ), such that, for any initial state  $\mathbf{x}(0) \in \mathbb{R}^{(m+1)q_x}$  and any  $\mathbf{u}_r \in \mathcal{L}_p^{q_u}$

$$\|\mathbf{y}_i(t) - g(\check{\mathbf{x}}_i)\|_{\mathcal{L}_p} \leq \alpha(\|\mathbf{u}_r(t)\|_{\mathcal{L}_p}) + \beta(\|\mathbf{x}(0) - \check{\mathbf{x}}\|), \quad t \in [0, \infty). \quad (5.2)$$

If in addition

$$\|\mathbf{y}_i(t) - g(\check{\mathbf{x}}_i)\|_{\mathcal{L}_p} \leq \|\mathbf{y}_{i-1}(t) - g(\check{\mathbf{x}}_{i-1})\|_{\mathcal{L}_p}, \quad (5.3)$$

the system is strictly  $\mathcal{L}_p$  string stable with respect to its input  $\mathbf{u}_r(t)$ . Here  $\|\cdot\|_{\mathcal{L}_p}$  denotes the signal  $p$ -norm, which for signal  $\mathbf{y}_i(t)$  is defined as

$$\|\mathbf{y}_i(t)\|_{\mathcal{L}_p} = \sqrt[p]{\int_0^\infty \|\mathbf{y}_i(t)\|_p^p dt}, \quad (5.4)$$

and  $\mathcal{L}_p^q$  denotes the  $q$ -dimensional space of vector signals that are bounded in the  $\mathcal{L}_p$  sense (i.e., have a bounded  $p$ -norm). In (5.2), the term  $\beta(\|\mathbf{x}(0) - \check{\mathbf{x}}\|)$  accounts for initial condition perturbations and the term  $\alpha(\|\mathbf{u}_r(t)\|_{\mathcal{L}_p})$  takes the external input  $\mathbf{u}_r(t)$  into account.

In the case of the planned cooperative trajectories, vehicle states  $\mathbf{S}_{i,k}(t) = [s_{i,k} \ \dot{s}_{i,k} \ \ddot{s}_{i,k}]^\top$  in plan step  $k$  are prescribed by B-splines over a time horizon spanning from planning time  $t_{i,k}$  to  $t_{i,k} + T_i$ . However, due to the implementation, the planning is updated after update time  $t_p$ , after which a new B-spline is generated to describe the vehicles states via planning  $\mathbf{S}_{i,k+1}(t)$ . As a

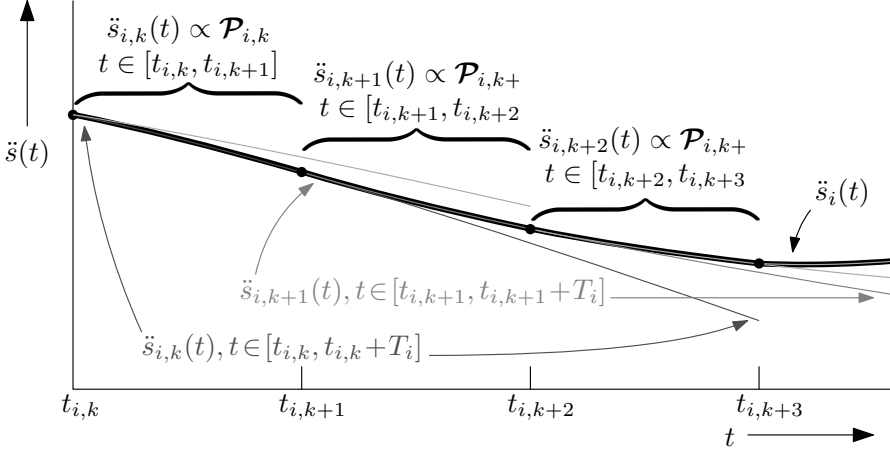


Figure 5.1: Discrete time representation of continuous time state trajectory

result, the vehicle states over time can be represented by piecewise B-splines. Figure 5.1 illustrates how the continuous-time state trajectory  $\ddot{s}_i(t)$  can be represented exactly by the sequentially planned trajectories  $\ddot{s}_{i,k}(t)$ ,  $t \in [t_{i,k}, t_{i,k+1}]$ . Consequently, the output  $\mathbf{y}_i(t)$  is also a piecewise B-spline, such that (5.4) can be written as

$$\|\mathbf{y}_i(t)\|_{\mathcal{L}_p} = \sqrt[p]{\sum_{k=0}^{\infty} \int_0^{t_p} \|\mathbf{y}_{i,k}(t)\|_p^p dt}, \quad (5.5)$$

where  $\mathbf{y}_{i,k}(t)$  denotes the output of the trajectory at plan step  $k$ . Considering  $\ddot{s}_i(t)$  as output  $\mathbf{y}_i(t)$  and the definition of B-spline trajectory of Chapter 3 in (3.26), the planned B-spline trajectories directly describe the output as

$$\mathbf{y}_{i,k}(t) := \ddot{s}_{i,k}(t) = \mathcal{B}_{n-2,p-2}(t) \mathcal{P}_{i,k}^{(2)} = \mathcal{B}_{n-2,p-2}(t) \Delta_{20} \mathcal{P}_{i,k}. \quad (5.6)$$

Next, considering specifically  $\mathcal{L}_2$  string stability, then

$$\|\mathbf{y}_i(t)\|_{\mathcal{L}_2} = \sqrt{2 \sum_{k=0}^{\infty} \int_0^{t_p} \|\ddot{s}_{i,k}(t)\|_2^2 dt} = \sqrt{2 \sum_{k=0}^{\infty} \mathcal{P}_{i,k}^T \Delta_{20}^T \Psi_s^T \Psi_s \Delta_{20} \mathcal{P}_{i,k}}, \quad (5.7)$$

where  $(\Psi_s^T \Psi_s) \in \mathbb{R}^{n-1 \times n-1}$  can be determined by means of Appendix A. Note that  $(\Psi_s^T \Psi_s)$  is indeed positive semi-definite, due to  $\|\mathbf{y}_i(t)\|_{\mathcal{L}_2}$  being non-negative. It is not positive definite, as not all control points in  $\mathcal{P}_{i,k}$  necessarily affect the output on the executed part of the trajectory  $\mathbf{y}_i(t)$ ,  $0 \leq t \leq t_p$ . Hence, the requirement on the  $\mathcal{L}_2$ -norm of the output can be written as a requirement on the control



points

$$\begin{aligned} \|y_i(t)\|_{\mathcal{L}_2} &\leq \|y_{i-1}(t)\|_{\mathcal{L}_2}, \\ \sum_{k=0}^{\infty} \mathcal{P}_{i,k}^\top \Delta_{20}^\top \Psi_s^\top \Psi_s \Delta_{20} \mathcal{P}_{i,k} &\leq \sum_{k=0}^{\infty} \mathcal{P}_{i-1,k}^\top \Delta_{20}^\top \Psi_s^\top \Psi_s \Delta_{20} \mathcal{P}_{i-1,k}. \end{aligned} \quad (5.8)$$

Additionally, by defining

$$\tilde{\mathbf{y}}_{i,k} := \Psi_s \Delta_{20} \mathcal{P}_{i,k}, \quad (5.9)$$

$$\|\tilde{\mathbf{y}}_{i,k}\|_{\ell_2}^2 := \sum_{k=0}^{\infty} \tilde{\mathbf{y}}_{i,k}^\top \tilde{\mathbf{y}}_{i,k} = \|\mathbf{y}_i(t)\|_{\mathcal{L}_2}^2, \quad (5.10)$$

where  $\|\cdot\|_{\ell_2}$  denotes the  $\ell_2$ -norm of a time-series. Hence, (5.8) can be rewritten as

$$\|\tilde{\mathbf{y}}_{i,k}\|_{\ell_2} \leq \|\tilde{\mathbf{y}}_{i-1,k}\|_{\ell_2}. \quad (5.11)$$

Thus to check the string stability in continuous time, it is sufficient to consider the discrete time series of the control points  $\mathcal{P}_{i,k}$ . Next, the discrete time dynamics of both the lead vehicle, as well as following vehicles is discussed.

## 5.2 Discrete-Time Dynamics of Planning Algorithm

Although the planned trajectories are planned in continuous time, they are planned at discrete time intervals. The vehicle state  $\mathbf{S}_i(t_{i,k})$  at which plan step  $k$  is initiated, is a function of the planning in the previous plan step  $k - 1$ . The planned vehicle state as function of time, is computed by means of the basis functions  $\mathcal{B}_{n,p}(t)$  of (3.23) (defined on  $t \in [t_{i,k}, t_{i,k} + T_i]$ ), and control points  $\mathcal{P}_{i,k}$ . Constant acceleration extrapolation is included to also define the planning past the time horizon  $t > t_{i,k} + T_i$  by means of (3.34), such that the vehicle states  $\mathbf{S}_{i,k}(t)$  of vehicle  $i$  planned at step  $k$  can be written as

$$\mathbf{S}_{i,k}(t) = \begin{bmatrix} s_{i,k}(t) \\ \dot{s}_{i,k}(t) \\ \ddot{s}_{i,k}(t) \end{bmatrix} = \Upsilon_i(t - t_{i,k}) \mathcal{P}_{i,k}, \quad t_{i,k} \leq t, \quad (5.12)$$

$$\Upsilon_i(t) = \begin{cases} \begin{bmatrix} \mathcal{B}_{n,p}(t) \mathbf{I} \\ \mathcal{B}_{n-1,p-1}(t) \Delta_{10} \\ \mathcal{B}_{n-2,p-2}(t) \Delta_{20} \end{bmatrix}, & \text{if } t \leq T_i, \\ \begin{bmatrix} 1 & (t - T_i) & \frac{1}{2}(t - T_i)^2 \\ 0 & 1 & (t - T_i) \\ 0 & 0 & 1 \end{bmatrix} \Upsilon(T_i), & \text{if } t > T_i, \end{cases} \quad (5.13)$$

where  $\Upsilon(t)$  is again defined as in (5.13). This notation is used to have a single notation for both evaluating within the time horizon of the planning, as well as for  $t > T_i$ . Note that planning times  $t_{i,k}$  are related as  $t_{i,k} = t_{i,k-1} + t_p$ , with  $t_p$  the planning update time. Denote the actual trajectory of the vehicle by  $\mathbf{S}_i(t) = [s_i(t) \ \dot{s}_i(t) \ \ddot{s}_i(t)]^\top$ , whereas the planned trajectories are denoted by  $\mathbf{S}_{i,k}(t)$ . By assuming perfect tracking of the planned trajectories, the vehicle state  $\mathbf{S}_i(t_{i,k})$  is equal to the planned vehicle state of plan step  $k - 1$  via

$$\mathbf{S}_i(t_{i,k}) = \mathbf{S}_{i,k-1}(t_{i,k}) = \Upsilon_i(t_p)\mathcal{P}_{i,k-1}. \quad (5.14)$$

Additionally, as the new planning at time step  $k$  is initiated from the current vehicle state, the following equality with control points for time step  $k$  should also be satisfied

$$\mathbf{S}_i(t_{i,k}) = \mathbf{S}_{i,k}(t_{i,k}) = \Upsilon_i(0)\mathcal{P}_{i,k}. \quad (5.15)$$

Hence, the control points for all vehicles (leader or following) are subjected to the constraint

$$\Upsilon_i(0)\mathcal{P}_{i,k} = \Upsilon_i(t_p)\mathcal{P}_{i,k-1}. \quad (5.16)$$

The remaining dynamics are different for the leader and following vehicles, and are discussed next. All of the presented trajectory generating algorithms can be summarized as quadratic minimizations over the control points, subjected to equality constraints. Hence, the planning problem can be written in the general form

$$\min_{\mathcal{P}_{i,k}} \|\mathbf{V}\mathcal{P}_{i,k} - \mathbf{w}\|^2, \quad (5.17a)$$

$$\text{s.t.} \quad \mathbf{Z}\mathcal{P}_{i,k} = \mathbf{q}, \quad (5.17b)$$

where  $\mathbf{Z}\mathcal{P}_{i,k} = \mathbf{q}$  represents the equality constraint and for which the solutions are found directly via

$$\mathcal{P}_{i,k} = [\mathbf{I}_{n+1} \quad \mathbf{0}]^{-1} \begin{bmatrix} 2\mathbf{V}^\top \mathbf{V} & \mathbf{Z}^\top \\ \mathbf{Z} & \mathbf{0} \end{bmatrix} \begin{bmatrix} 2\mathbf{V}^\top \mathbf{w} \\ \mathbf{q} \end{bmatrix}. \quad (5.18)$$

In what follows, for each of the algorithms it is discussed which values it takes for  $\mathbf{Z}$ ,  $\mathbf{V}$ ,  $\mathbf{w}$  and  $\mathbf{q}$ . Note that if  $\mathbf{w}$  and  $\mathbf{q}$  are linear in  $\mathcal{P}_{i-1,k}$  and  $\mathcal{P}_{i,k-1}$  respectively (e.g., when the equality constraint involves the previously planned trajectory via  $\mathcal{P}_{i-1,k}$ ), the solution (5.18) can also be written as

$$\mathcal{P}_{i,k} = \mathbf{A}_i\mathcal{P}_{i,k-1} + \mathbf{B}_i\mathcal{P}_{i-1,k} + \mathbf{E}_i. \quad (5.19)$$

In order to satisfy certain properties (such as continuity and achieving steady state vehicle following), Appendix B presents requirements on discrete-time dynamics matrices  $\mathbf{A}_i$ ,  $\mathbf{B}_i$  and  $\mathbf{E}_i$ . These can be used to come up with different planning strategies than presented in this thesis.

This result can be summarized in the following theorems

**Theorem 5.1.** *The continuous-time trajectory of a vehicle (assuming perfect tracking) that utilizes a receding horizon trajectory planner, in which the planned trajectories  $\mathbf{S}_{i,k}(t), t \in [t_{i,k}, t_{i,k+1}]$  (where  $t_{i,k+1} = t_{i,k} + t_p$ ) are constructed by means of B-splines, can be represented exactly by a discrete-time series of control points  $\mathcal{P}_{i,k}$  for given Basis functions  $\mathcal{B}_{n,p}(t)$ .*

*Proof.* Assuming perfect tracking of the planned trajectories, the executed trajectory perfectly follows the planned trajectories. These planned trajectories are constructed by means of B-splines. Consequently, the executed trajectory of such a vehicle consists of piecewise B-splines. Specifically, during planning step  $k$ , the vehicle will execute the motion spanning from  $t \in [t_{i,k}, t_{i,k+1}]$ , where  $t_{i,k+1} = t_{i,k} + t_p$ . During planning step  $k$  the executed motion is described by

$$\mathbf{S}_{i,k}(t) = \mathbf{\Upsilon}_i(t - t_{i,k})\mathcal{P}_{i,k}, \quad t_{i,k} \leq t \leq t_{i,k} + t_p, \quad (5.20)$$

with  $\mathbf{\Upsilon}_i(t)$  according to (5.13). Hence, for given basis functions array  $\mathbf{\Upsilon}_i(t)$  the motion during planning step  $k$  is fully described by  $\mathcal{P}_{i,k}$ . Similarly, segment  $k + 1$  is fully described by  $\mathcal{P}_{i,k+1}$ , segment  $k + 2$  by  $\mathcal{P}_{i,k+2}$  and so on, which completes the proof.  $\square$

**Theorem 5.2.** *The continuous time trajectory of a vehicle (assuming perfect tracking) that utilizes a receding horizon trajectory planner, in which the planned trajectories are constructed by means of B-splines that minimize a quadratic cost with linear equality constraints in its control points, can be represented by an equivalent discrete-time system and initial trajectory planning  $\mathcal{P}_{0,0}$  for vehicle  $i = 0$ , and initial planning  $\mathcal{P}_{i,0}$  and initial planning of preceding vehicle  $\mathcal{P}_{i-1,0}$  for vehicle  $i > 0$ .*

*Proof.* Using the result of Theorem 5.1 the continuous time trajectory can be represented fully by the series of control points  $\mathcal{P}_{i,k}, k \in \{0, \dots, \infty\}$ . Since trajectories should be feasible, they should be initialized in the state of the system at the time instance of the start of the planning  $t_{i,k}$ . The state at the start of the next planning step  $t_{i,k+1}$  can therefore be described by

$$\mathbf{S}_{i,k}(t_{i,k+1}) = \mathbf{\Upsilon}_i(t_p)\mathcal{P}_{i,k} = \mathbf{\Upsilon}_i(0)\mathcal{P}_{i,k+1} \quad (5.21)$$

The presented planning algorithms can be written in the form (5.17), with solutions in the form of (5.18). Provided that  $\mathbf{w}$  and  $\mathbf{q}$  are linear in  $\mathcal{P}_{i-1,k}$  and  $\mathcal{P}_{i,k-1}$  respectively (e.g., when the equality constraint involves the previously planned trajectory via  $\mathcal{P}_{i-1,k}$ ), the solution (5.18) can also be written as (5.19), which in combination with initial condition  $\mathcal{P}_{i,0}$  and  $\mathcal{P}_{i-1,0}$ ,  $i > 0$  is equivalent to a discrete time system with the control points of the consecutive B-spline trajectories as states. These control points can then be used to construct the piecewise B-spline trajectory, which completes the proof.  $\square$

### 5.2.1 Lead Vehicle

The dynamics of the lead vehicle can take different forms. First, in Section 3.3.4, dynamics for the lead vehicle are discussed that attempt to minimize the difference with respect to the previously generated trajectory, evaluated at the Greville abscissae. These dynamics are used to evaluate the performance of the following vehicle in terms of string stable behavior. A second algorithm for the lead vehicle is presented in Section 4.1.1, which is used to track a given velocity. In the remainder of this section, these two planning algorithms are formulated as discrete-time dynamic systems.

#### Trajectory Matching Planner

The lead vehicle described in Section 3.3.4 and [122] is set-up as an initial condition response. The control points of each consecutive planning cycle attempts to minimize the error

$$J_0 = \sum_{j=3}^n (s_{0,k}(t_{0,k} + \mu_j) - s_{0,k-1}(t_{0,k} + \mu_j))^2. \quad (5.22)$$

However, note that in this case the trajectory needs to be extrapolated for any plan step  $t_p > 0$ , as the new trajectory extends further in time as the previous one. The extrapolation of the trajectories by means of a constant acceleration assumption is denoted as

$$s_i(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \Upsilon_i(t - t_{i,k}) \mathcal{P}_{i,k}, \quad t_{i,k} < t, \quad (5.23)$$

where  $\Upsilon_i(t - t_{i,k})$  is defined as in (3.34). The cost (5.22) can thus be written as

$$J_0 = \left\| \underbrace{\begin{bmatrix} \mathcal{B}_{n,p}(\mu_3) \\ \vdots \\ \mathcal{B}_{n,p}(\mu_n) \end{bmatrix} \mathcal{P}_{i,k}}_{s_{0,k}(t_{0,k} + \mu_j), j \in \{3, \dots, n\}} - \underbrace{\begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \Upsilon_0(t_p + \mu_3) \\ \vdots \\ \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \Upsilon_0(t_p + \mu_n) \end{bmatrix} \mathcal{P}_{0,k-1}}_{s_{0,k-1}(t_{0,k} + \mu_j), j \in \{3, \dots, n\}} \right\|_2. \quad (5.24)$$

In addition to this minimization criterion and continuity constraint (5.16), also the terminal acceleration is constrained at zero, and the terminal velocity of each plan step is kept the same which is achieved by

$$\begin{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \Upsilon_0(T_0) \\ \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \Upsilon_0(T_0) \end{bmatrix} \mathcal{P}_{0,k} = \begin{bmatrix} 0 \\ \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \Upsilon_0(T_0) \end{bmatrix} \mathcal{P}_{0,k-1}. \quad (5.25)$$

Hence, this scenario can be represented by (5.17) with solution (5.18) for control points at plan step  $k$ , when substituting

$$\mathbf{V} = \begin{bmatrix} \mathcal{B}_{n,p}(\mu_3) \\ \vdots \\ \mathcal{B}_{n,p}(\mu_n) \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} [1 \ 0 \ 0] \Upsilon_0(t_p + \mu_3) \\ \vdots \\ [1 \ 0 \ 0] \Upsilon_0(t_p + \mu_n) \end{bmatrix} \mathcal{P}_{0,k-1}, \quad (5.26a)$$

$$\mathbf{Z} = \begin{bmatrix} \Upsilon_0(0) \\ [0 \ 0 \ 1] \Upsilon_0(T_0) \\ [0 \ 1 \ 0] \Upsilon_0(T_0) \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} \Upsilon_0(t_p) \\ 0 \\ [0 \ 1 \ 0] \Upsilon_0(T_0) \end{bmatrix} \mathcal{P}_{0,k-1}, \quad (5.26b)$$

and can also be written as (5.19),

$$\mathcal{P}_{0,k} = \mathbf{A}_0 \mathcal{P}_{0,k-1}, \quad (5.27a)$$

$$\mathbf{A}_0 = [\mathbf{I}_{n+1} \ 0] \begin{bmatrix} \mathbf{V}^\top \mathbf{V} & \mathbf{Z}^\top \\ \mathbf{Z} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{V}^\top \begin{bmatrix} [1 \ 0 \ 0] \Upsilon_0(t_p + \mu_3) \\ \vdots \\ [1 \ 0 \ 0] \Upsilon_0(t_p + \mu_n) \end{bmatrix} \\ \begin{bmatrix} \Upsilon_0(t_p) \\ 0 \\ [0 \ 1 \ 0] \Upsilon_0(T_0) \end{bmatrix} \end{bmatrix}. \quad (5.27b)$$

### Velocity-controlled Planner

Instead of attempting to match a previously planned trajectory, it is also possible to track a given desired velocity, while penalizing large accelerations and jerk by means of weights. This second type of planner is described in Section 4.1.1, where multiple trajectories are generated. However, assuming all generated trajectories pass the feasibility and collision test, the same optimal trajectory is selected each time step. Therefore, this lead vehicle planner can be translated to the following minimization

$$\begin{aligned} \min_{\mathcal{P}_{0,k}} & \left\| \begin{bmatrix} Q_{\dot{s}} \Delta_{10} \\ Q_{\ddot{s}} \Delta_{20} \\ Q_{\dddot{s}} \Delta_{30} \end{bmatrix} \mathcal{P}_{0,k} - \begin{bmatrix} Q_{\dot{s}} \mathbf{1}_{n \times 1} \\ \mathbf{0}_{n-1 \times 1} \\ \mathbf{0}_{n-2 \times 1} \end{bmatrix} u \right\|_2, \\ \text{s.t.} \quad & \Upsilon_0(0) \mathcal{P}_{0,k} = \Upsilon_0(t_p) \mathcal{P}_{0,k-1}, \end{aligned} \quad (5.28)$$

where  $u$  is the scalar input representing the desired velocity,  $Q_{\dot{s}} \geq 0$ ,  $Q_{\ddot{s}} \geq 0$ ,  $Q_{\dddot{s}} \geq 0$  are weights for velocity, acceleration and jerk respectively. Note that this minimization is applied directly at the control points, rather than on the resulting spline. However, as the control polygon, constructed by the control points, dictate the shape of the spline, and when the control polygon converges, the spline does

as well. This results in the following:

$$\mathbf{V} = \begin{bmatrix} Q_s \Delta_{10} \\ Q_s \Delta_{20} \\ Q_s \Delta_{30} \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} Q_s \mathbf{1}_{n \times 1} \\ \mathbf{0}_{n-1 \times 1} \\ \mathbf{0}_{n-2 \times 1} \end{bmatrix} u, \quad (5.29a)$$

$$\mathbf{Z} = \Upsilon_0(0), \quad \mathbf{q} = \Upsilon_0(t_p) \mathcal{P}_{0,k-1}, \quad (5.29b)$$

which can be written in the form of (5.19)

$$\mathcal{P}_{0,k} = \mathbf{A}_0 \mathcal{P}_{0,k-1} + \mathbf{B}_0 u, \quad (5.30a)$$

$$\mathbf{A}_0 = [\mathbf{I}_{n+1} \quad \mathbf{0}] \begin{bmatrix} \mathbf{V}^\top \mathbf{V} & \Upsilon_0(0)^\top \\ \Upsilon_0(0) & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \Upsilon_0(t_p) \end{bmatrix}, \quad (5.30b)$$

$$\mathbf{B}_0 = [\mathbf{I}_{n+1} \quad \mathbf{0}] \begin{bmatrix} \mathbf{V}^\top \mathbf{V} & \Upsilon_0(0)^\top \\ \Upsilon_0(0) & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{V}^\top \begin{bmatrix} Q_s \mathbf{1}_{n \times 1} \\ \mathbf{0}_{n-1 \times 1} \\ \mathbf{0}_{n-2 \times 1} \end{bmatrix} \\ \mathbf{0} \end{bmatrix}. \quad (5.30c)$$

### 5.2.2 Follower Vehicle

For any follower vehicle using the approach described in Section 3.3, with a solution for control points (3.32), the process of finding a trajectory can also be written in the form of (5.17) with solution (5.18). Recall that the following objective is minimized.

$$\min_{\mathcal{P}_{i,k}} \left\| \begin{bmatrix} [1 \quad 0 \quad 0] \Upsilon_{i-1}(\mu_3) \\ \vdots \\ [1 \quad 0 \quad 0] \Upsilon_{i-1}(\mu_n) \end{bmatrix} \mathcal{P}_{i-1,k} - c_i - L_i - \Omega \mathcal{P}_{i,k} \right\|_2, \quad (5.31a)$$

$$\text{s.t. } \Upsilon_i(0) \mathcal{P}_{i,k} = \Upsilon_i(t_p) \mathcal{P}_{i,k-1}, \quad (5.31b)$$

$$\Omega = \begin{bmatrix} \mathcal{B}_{n,p}(\mu_3) + h_i \mathcal{B}_{n-1,p-1}(\mu_3) \Delta_{10} \\ \vdots \\ \mathcal{B}_{n,p}(\mu_n) + h_i \mathcal{B}_{n-1,p-1}(\mu_n) \Delta_{10} \end{bmatrix}, \quad (5.31c)$$

which has a solution in the form of (5.18) as

$$\mathbf{V} = \Omega, \quad \mathbf{w} = \begin{bmatrix} [1 \quad 0 \quad 0] \Upsilon_{i-1}(\mu_3) \\ \vdots \\ [1 \quad 0 \quad 0] \Upsilon_{i-1}(\mu_n) \end{bmatrix}^\top \mathcal{P}_{i-1,k} - c_i - L_i, \quad (5.32a)$$

$$\mathbf{Z} = \Upsilon_i(0), \quad \mathbf{q} = \Upsilon_i(t_p) \mathcal{P}_{i,k-1}. \quad (5.32b)$$

Due to the dependency of  $\mathbf{w}$  on  $\mathcal{P}_{i-1,k}$  this can also be written in the form of (5.19) as

$$\mathcal{P}_{i,k} = \mathbf{A}_i \mathcal{P}_{i,k-1} + \mathbf{B}_i \mathcal{P}_{i-1,k} + \mathbf{E}_i, \quad (5.33a)$$

$$\mathbf{A}_i = [\mathbf{I}_{n+1} \quad \mathbf{0}] \begin{bmatrix} \mathbf{\Omega}^\top \mathbf{\Omega} & \mathbf{\Upsilon}_i(0)^\top \\ \mathbf{\Upsilon}_i(0) & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{\Upsilon}_i(t_p) \end{bmatrix}, \quad (5.33b)$$

$$\mathbf{B}_i = [\mathbf{I}_{n+1} \quad \mathbf{0}] \begin{bmatrix} \mathbf{\Omega}^\top \mathbf{\Omega} & \mathbf{\Upsilon}_i(0)^\top \\ \mathbf{\Upsilon}_i(0) & \mathbf{0} \end{bmatrix}^{-1} \mathbf{\Omega}^\top \begin{bmatrix} [1 \quad 0 \quad 0] \mathbf{\Upsilon}_{i-1}(\mu_3) \\ \vdots \\ [1 \quad 0 \quad 0] \mathbf{\Upsilon}_{i-1}(\mu_n) \\ \mathbf{0} \end{bmatrix}^\top, \quad (5.33c)$$

$$\mathbf{E}_i = [\mathbf{I}_{n+1} \quad \mathbf{0}] \begin{bmatrix} \mathbf{\Omega}^\top \mathbf{\Omega} & \mathbf{\Upsilon}_i(0)^\top \\ \mathbf{\Upsilon}_i(0) & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Omega}^\top \mathbf{1}_{n-2 \times 1} (-c_i - L_i) \\ \mathbf{0} \end{bmatrix}. \quad (5.33d)$$

With the dynamics described lead vehicle dynamics of (5.27) or (5.30), and the discrete-time follower vehicle dynamics (5.33), the behavior of the string of vehicles is defined.

### 5.3 String Stability in the Frequency Domain

With the discrete-time dynamics of the vehicles described, it is now possible to analyze string stability, for arbitrary inputs (i.e., the control points which are not fixed due to continuity requirements) of the lead vehicle. First, a coordinate transformation is used to transform the discrete-time state vector  $\mathcal{P}_{i,k}$ , to state coordinates that have an equilibrium in the origin, by removing the inherent drift due to the curvilinear distance  $s_i(t)$  being included in the state vector. This transformed state vector is used to define transfer functions that describe the relation between relevant signals. Additionally, a reduction in dimension is used to eliminate the control points of the acceleration spline that do not directly impact the output on the executed part of the trajectory  $t \in [t_{i,k}, t_{i,k} + t_p]$ . Finally, the transfer function  $\Gamma_i(z)$ , which relates the  $\mathcal{L}_2$ -norm of the acceleration signal of vehicle  $i - 1$  to that of vehicle  $i$ , is shown to depend not only on the planning algorithm of vehicle  $i$ , but also on the planning algorithm of all vehicles in front of it. Recognize that with initial conditions equal to zero (5.2) can also be evaluated as

$$\|\mathbf{y}_i(t) - g(\mathbf{x}_i)\|_{\mathcal{L}_p} \leq \alpha \left( \|\mathbf{u}_r(t)\|_{\mathcal{L}_p} \right), \quad t \in (0, \infty), \quad (5.34)$$

when considering the  $\mathcal{L}_p$ -norm of the signal over the time interval  $t \in [-\infty, \infty]$

$$\|\mathbf{y}_i(t)\|_{\mathcal{L}_p} = \sqrt[p]{\int_{-\infty}^{\infty} \mathbf{y}_i^p(t) dt}, \quad (5.35)$$

and assuming a system initially at rest. This approach is used in this section to prevent the influence of initial conditions to the problem.

### 5.3.1 Coordinate Transformation

The approach presented in [137] is followed, by introducing the following coordinate transformation

$$\mathbf{x}_{i,k} = \begin{bmatrix} s_{i,k} \\ v_{i,k} \\ \mathbf{a}_{i,k} \end{bmatrix} = \begin{bmatrix} s_{i,k} \\ v_{i,k} \\ a_{0,i,k} \\ \mathbf{a}_{1,i,k} \\ \mathbf{a}_{2,i,k} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ & & \Delta_{20} & \end{bmatrix}}_{\Phi} \mathbf{P}_{i,k}, \quad (5.36)$$

$$\bar{\mathbf{x}}_{i,k} = \begin{bmatrix} \bar{s}_{i,k} \\ \bar{v}_{i,k} \\ \bar{\mathbf{a}}_{i,k} \end{bmatrix} = \Phi \mathbf{P}_{i,k} - (\bar{s}_i + kt_p \bar{v}) \mathbf{u}_1 - \bar{v} \mathbf{u}_2, \quad (5.37)$$

$$\bar{s}_i = \bar{s}_{i-1} - (L_i + c_i) - h_i \bar{v}, \quad (5.38)$$

$$\bar{s}_0 = \lim_{k \rightarrow \infty} (s_{0,k} - k\bar{v}), \quad (5.39)$$

where  $\mathbf{u}_1 = [1 \ 0 \ \dots \ 0]^\top$  and  $\mathbf{u}_2 = [0 \ 1 \ 0 \ \dots \ 0]^\top$ ,  $a_{0,i,k} \in \mathbb{R}^1$ ,  $\mathbf{a}_{1,i,k} \in \mathbb{R}^r$  and  $\mathbf{a}_{2,i,k} \in \mathbb{R}^{(n-2)-r}$  can be vector valued. The distinction between  $a_{0,i,k}$ ,  $\mathbf{a}_{1,i,k}$  and  $\mathbf{a}_{2,i,k}$  is as follows. If the basis function  $N_{j,n-2,p-2}(t)$  corresponding to acceleration control point  $P_{j,i,k}^{(2)}$  is not supported on the executed part of the trajectories (i.e.,  $N_{j,n-2,p-2}(t) = 0$ ,  $t \in [t_{i,k}, t_{i,k} + t_p]$ ), then that control point is contained in  $\bar{\mathbf{a}}_{2,i,k}$ , otherwise it is either contained in  $\bar{a}_{0,i,k}$  (only the first acceleration control point that is fixed due to continuity) or contained in  $\bar{\mathbf{a}}_{1,i,k}$ . For example, if  $t_p = T_i$ , then all acceleration control points have non-zero support somewhere on  $t \in [t_{i,k}, t_{i,k} + T_i]$ , and  $\bar{\mathbf{a}}_{2,i,k}$  is an empty array. The interpretation of values  $\bar{s}_0$  and  $\bar{v}$  is illustrated in Figure 5.2. It can be seen that  $\bar{v}$  represents the velocity to which the velocity of each vehicle in the string converges, while  $\bar{s}_i$  represents position for  $k = 0$  related to the new equilibrium velocity  $\bar{v}$ .

The state described by (5.36) has an equilibrium in the origin, that represents steady state vehicle following with a velocity of  $\bar{v}$ . The state vector  $\bar{\mathbf{x}}_{i,k}$  contains information related to the position, velocity, and acceleration of vehicle  $i$  at the start of plan step  $k$ , denoted by  $\bar{s}_{i,k}$ ,  $\bar{v}_{i,k}$ , and  $\bar{a}_{0,i,k}$ . The remaining control points for the acceleration spline are denoted by  $\mathbf{a}_{1,i,k}$  and  $\mathbf{a}_{2,i,k}$ . To derive the dynamics for  $\bar{\mathbf{x}}_{i,k}$  first define  $\mathbf{z}_{i,k}$  such that  $\bar{\mathbf{x}}_{i,k} = \Phi \mathbf{z}_{i,k}$  as

$$\mathbf{z}_{i,k} = \mathbf{P}_{i,k} - (\bar{s}_i + kt_p \bar{v}) \mathbf{1} - \bar{v} \Xi_{10|\mathbf{U}_i^{(1)}} \mathbf{1}, \quad (5.40)$$

and where  $\mathbf{1} = [1 \ \dots \ 1]^\top$  and  $\Xi_{10|\mathbf{U}_i^{(1)}} \in \mathbb{R}^{n+1 \times n}$  represents the matrix to integrate the velocity control points  $\mathbf{P}_i^{(1)}$  to obtain the offsetted position control



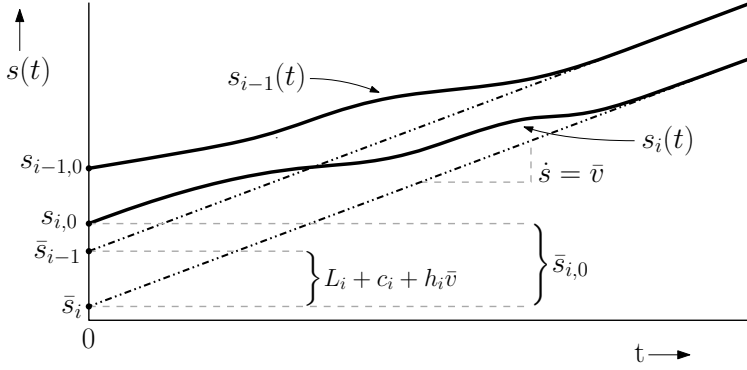


Figure 5.2: Interpretation of elements of coordinate transformation (5.37)

points  $\mathcal{P}_{i,k} - P_{0,i,k} = \Xi_{10|\mathbf{U}_i^{(1)}} \mathcal{P}_i^{(1)}$ . Then by using (5.19)

$$\begin{aligned} \mathbf{z}_{i,k} &= \mathcal{P}_{i,k} - (\bar{s}_i + kt_p \bar{v}) \mathbb{1} - \bar{v} \Xi_{10|\mathbf{U}_i^{(1)}} \mathbb{1} \\ &= (\mathbf{A}_i \mathcal{P}_{i,k-1} + \mathbf{B}_i \mathcal{P}_{i-1,k} + \mathbf{E}_i) - (\bar{s}_i + kt_p \bar{v}) \mathbb{1} - \bar{v} \Xi_{10|\mathbf{U}_i^{(1)}} \mathbb{1}, \end{aligned} \quad (5.41)$$

then substituting

$$\mathcal{P}_{i,k-1} = \mathbf{z}_{i,k-1} + (\bar{s}_i + (k-1)t_p \bar{v}) \mathbb{1} + \bar{v} \Xi_{10|\mathbf{U}_i^{(1)}} \mathbb{1}, \quad (5.42)$$

$$\mathcal{P}_{i-1,k} = \mathbf{z}_{i-1,k} + (\bar{s}_{i-1} + kt_p \bar{v}) \mathbb{1} + \bar{v} \Xi_{10|\mathbf{U}_i^{(1)}} \mathbb{1}, \quad (5.43)$$

in (5.41), results in

$$\begin{aligned} \mathbf{z}_{i,k} &= \mathbf{A}_i \mathbf{z}_{i,k-1} + \mathbf{B}_i \mathbf{z}_{i-1,k} + \mathbf{A}_i \left( (\bar{s}_i + (k-1)t_p \bar{v}) \mathbb{1} + \bar{v} \Xi_{10|\mathbf{U}_i^{(1)}} \mathbb{1} \right) \\ &\quad + \mathbf{B}_i \left( (\bar{s}_{i-1} + kt_p \bar{v}) \mathbb{1} + \bar{v} \Xi_{10|\mathbf{U}_i^{(1)}} \mathbb{1} \right) \\ &\quad + \mathbf{E}_i - (\bar{s}_i + kt_p \bar{v}) \mathbb{1} - \bar{v} \Xi_{10|\mathbf{U}_i^{(1)}} \mathbb{1}. \end{aligned} \quad (5.44)$$

Then using  $\bar{s}_{i-1} = \bar{s}_i + (L_i + c_i) + h \bar{v}$  and the properties under Lemma B.2 in Appendix B

$$\begin{aligned} \mathbf{z}_{i,k} &= \mathbf{A}_i \mathbf{z}_{i,k-1} + \mathbf{B}_i \mathbf{z}_{i-1,k} + \mathbf{A}_i \left( (\bar{s}_i + kt_p \bar{v}) \mathbb{1} + \bar{v} \Xi_{10|\mathbf{U}_i^{(1)}} \mathbb{1} \right) \\ &\quad - \mathbf{A}_i t_p \bar{v} \mathbb{1} + \mathbf{B}_i \left( (\bar{s}_{i-1} + kt_p \bar{v}) \mathbb{1} + \bar{v} \Xi_{10|\mathbf{U}_i^{(1)}} \mathbb{1} \right) \\ &\quad + \mathbf{E}_i - (\bar{s}_i + kt_p \bar{v}) \mathbb{1} - \bar{v} \Xi_{10|\mathbf{U}_i^{(1)}} \mathbb{1} \\ &= \mathbf{A}_i \mathbf{z}_{i,k-1} + \mathbf{B}_i \mathbf{z}_{i-1,k} + \bar{v} (\mathbf{A}_i + \mathbf{B}_i) \Xi_{10|\mathbf{U}_i^{(1)}} \mathbb{1} \\ &\quad + \bar{v} (\mathbf{I} - \mathbf{A}_i - \mathbf{B}_i) \Xi_{10|\mathbf{U}_i^{(1)}} \mathbb{1} - \bar{v} \Xi_{10|\mathbf{U}_i^{(1)}} \mathbb{1} \\ &= \mathbf{A}_i \mathbf{z}_{i,k-1} + \mathbf{B}_i \mathbf{z}_{i-1,k}. \end{aligned} \quad (5.45)$$

Using these dynamics for  $\mathbf{z}_{i,k}$ , it can be seen that

$$\begin{aligned}\bar{\mathbf{x}}_{i,k} &= \Phi \mathbf{z}_{i,k} = \Phi \mathcal{P}_{i,k} - (\bar{s}_i + kt_p \bar{v}) \Phi \mathbb{1} - \bar{v} \Phi \Xi_{10|U_i^{(1)}} \mathbb{1} \\ &= \Phi \mathcal{P}_{i,k} - (\bar{s}_i + kt_p \bar{v}) \mathbb{u}_1 - \bar{v} \mathbb{u}_2,\end{aligned}\quad (5.46)$$

which consists of states  $\bar{\mathbf{x}}_{i,k} = [\bar{s}_{i,k} \ \bar{v}_{i,k} \ \bar{a}_{0,i,k} \ \bar{\mathbf{a}}_{1,i,k}^\top \ \bar{\mathbf{a}}_{2,i,k}^\top]^\top$ . Note that  $\Xi_{10|U_i^{(1)}} \mathbb{1} \bar{v}$  represents the position control points for steady state driving at velocity  $\bar{v}$ , and thus  $\bar{v} \Phi \Xi_{10|U_i^{(1)}} \mathbb{1}$  represents the transformed state vector  $[0 \ \bar{v} \ 0 \dots 0]^\top$ . Additionally, The state  $\bar{\mathbf{x}}_{i,k}$  has dynamics as

$$\bar{\mathbf{x}}_{i,k} = \bar{\mathbf{A}}_i \bar{\mathbf{x}}_{i,k-1} + \bar{\mathbf{B}}_i \bar{\mathbf{x}}_{i-1,k}, \quad (5.47a)$$

$$\bar{\mathbf{A}}_i = \Phi \mathbf{A}_i \Phi^{-1}, \quad \bar{\mathbf{B}}_i = \Phi \mathbf{B}_i \Phi^{-1}. \quad (5.47b)$$

### 5.3.2 Transfer Function Description

Next, the discrete-time transfer function that relates  $\tilde{\mathbf{y}}_{i,k}$  to  $\tilde{\mathbf{y}}_{i-1,k}$  is derived, which is used to assess string stability of the continuous-time signals via (5.7) and (5.11). Note that the discrete time series of elements in  $\bar{\mathbf{x}}_{i,k}$  are not sampled versions of their corresponding continuous-time signals. Instead, using the discrete-time series  $\bar{\mathbf{x}}_{i,k}$ , the continuous-time signals can be reconstructed *exactly*, as illustrated in Figure 5.1. The new state vector  $\bar{\mathbf{x}}_{i,k}$  is used to describe derive the desired transfer function. The first part of this state vector is independent of the external input  $\bar{\mathbf{x}}_{i-1,k}$  due to continuity requirements (see Appendix B), and is computed via,

$$\begin{bmatrix} \bar{s}_{i,k} \\ \bar{v}_{i,k} \\ \bar{a}_{0,i,k} \end{bmatrix} = \bar{\mathbf{A}}_{i,11} \begin{bmatrix} \bar{s}_{i,k-1} \\ \bar{v}_{i,k-1} \\ \bar{a}_{0,i,k-1} \end{bmatrix} + \bar{\mathbf{A}}_{i,12} \begin{bmatrix} \bar{\mathbf{a}}_{1,i,k-1} \\ \bar{\mathbf{a}}_{2,i,k-1} \end{bmatrix}, \quad (5.48)$$

where  $\bar{\mathbf{A}}_{i,11}$  and  $\bar{\mathbf{A}}_{i,12}$  represent matrix partitions of  $\bar{\mathbf{A}}_i$ . The second part (the acceleration control points that are not fixed due to continuity requirements) does depend on the external input

$$\begin{aligned} \begin{bmatrix} \bar{\mathbf{a}}_{1,i,k} \\ \bar{\mathbf{a}}_{2,i,k} \end{bmatrix} &= \bar{\mathbf{A}}_{i,21} \begin{bmatrix} \bar{s}_{i,k-1} \\ \bar{v}_{i,k-1} \\ \bar{a}_{0,i,k-1} \end{bmatrix} + \bar{\mathbf{A}}_{i,22} \begin{bmatrix} \bar{\mathbf{a}}_{1,i,k-1} \\ \bar{\mathbf{a}}_{2,i,k-1} \end{bmatrix} \\ &\quad + \bar{\mathbf{B}}_{i,21} \begin{bmatrix} \bar{s}_{i-1,k} \\ \bar{v}_{i-1,k} \\ \bar{a}_{0,i-1,k} \end{bmatrix} + \bar{\mathbf{B}}_{i,22} \begin{bmatrix} \bar{\mathbf{a}}_{1,i-1,k} \\ \bar{\mathbf{a}}_{2,i-1,k} \end{bmatrix}. \end{aligned} \quad (5.49)$$

Then, using the  $z$ -transform (i.e.,  $X(z) := \mathcal{Z}(x_k)$ ), the transfer function of control points that are influenced by the external outputs can be found, which are related

by transfer function  $F_i(z)$

$$\begin{bmatrix} A_{1,i}(z) \\ A_{2,i}(z) \end{bmatrix} = \underbrace{\begin{bmatrix} F_{i,11}(z) & F_{i,12}(z) \\ F_{i,21}(z) & F_{i,22}(z) \end{bmatrix}}_{F_i(z)} \begin{bmatrix} A_{1,i-1}(z) \\ A_{2,i-1}(z) \end{bmatrix}, \quad (5.50)$$

$$F_i(z) = (\mathbf{I} - \bar{\mathbf{A}}_{i,21}(z\mathbf{I} - \bar{\mathbf{A}}_{i,11})^{-1}\bar{\mathbf{A}}_{i,12}z^{-1} - \bar{\mathbf{A}}_{i,22}z^{-1})^{-1} (\bar{\mathbf{B}}_i^{21}(z\mathbf{I} - \bar{\mathbf{A}}_{i,11})^{-1}\bar{\mathbf{A}}_{i,12} + \bar{\mathbf{B}}_i^{22}), \quad (5.51)$$

where  $F_{i,11}(z) \in \mathbb{C}^{r \times r}$ ,  $F_{i,12}(z) \in \mathbb{C}^{r \times (n-2)-r}$ ,  $F_{i,21}(z) \in \mathbb{C}^{(n-2)-r \times r}$  and  $F_{i,22}(z) \in \mathbb{C}^{(n-2)-r \times (n-2)-r}$ . To make a clear distinction between matrices and transfer functions, all transfer functions are denoted by regular fonts, as opposed to bold fonts (which are reserved for matrices). Note that (5.51) has been derived using the following transfer functions for the acceleration control points:

$$\begin{bmatrix} S_i(z) \\ V_i(z) \\ A_{0,i}(z) \end{bmatrix} = \underbrace{(z\mathbf{I} - \bar{\mathbf{A}}_{i,11})^{-1}\bar{\mathbf{A}}_{i,12}}_{K_i(z)} \begin{bmatrix} A_{1,i}(z) \\ A_{2,i}(z) \end{bmatrix}, \quad (5.52)$$

$$\begin{bmatrix} A_{0,i}(z) \\ A_{1,i}(z) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ & \mathbf{I} & 0 \end{bmatrix} K_i(z) \begin{bmatrix} A_{1,i}(z) \\ A_{2,i}(z) \end{bmatrix}, \quad (5.53)$$

with  $K_i(z) \in \mathbb{C}^{3 \times (n-2)}$ . The  $z$ -transform  $Y_i(z) := \mathcal{Z}(\tilde{\mathbf{y}}_{i,k})$ , of the non-scalar output  $\tilde{\mathbf{y}}_{i,k}$  in (5.9) (related to the  $\mathcal{L}_2$ -norm of the acceleration signal) is unaffected by  $A_{2,i}(z)$  (per definition of  $\mathbf{a}_{2,i,k}$ ), and is defined as

$$Y_i(z) = \Psi_{\tilde{s},a} \begin{bmatrix} A_{0,i}(z) \\ A_{1,i}(z) \end{bmatrix}, \quad (5.54)$$

$$Y_i(z) = \Psi_{\tilde{s},a} \begin{bmatrix} 0 & 0 & 1 \\ & \mathbf{I} & 0 \end{bmatrix} K_i(z) \underbrace{F_i(z)F_{i-1}(z) \dots F_1(z)F_0(z)}_{[A_{1,i}^\top(z), A_{2,i}^\top(z)]^\top} \begin{bmatrix} A_{1,0}(z) \\ A_{2,0}(z) \end{bmatrix}, \quad (5.55)$$

$$\Psi_{\tilde{s}} = \begin{bmatrix} \Psi_{\tilde{s},a} & \mathbf{0}_{r+1 \times (n-2)-r} \\ \mathbf{0}_{(n-2)-r \times r+1} & \mathbf{0}_{(n-2)-r \times (n-2)-r} \end{bmatrix},$$

where  $\Psi_{\tilde{s}} \in \mathbb{R}^{n-1 \times n-1}$  is according to the definition described in (5.7), and  $\Psi_{\tilde{s},a} \in \mathbb{R}^{r+1 \times r+1}$  is a partition that is associated with coefficients that *do* effect the acceleration  $\tilde{s}_{i,k}(t)$  on the interval  $t \in [t_{k,i}, t_{k,i} + t_p]$ . The objective is to relate  $Y_i(z)$  directly to  $Y_{i-1}(z)$ , as this could be used to verify string stability. Since  $Y_{i-1}(z)$  only depends on  $A_{0,i-1}(z)$  and  $A_{1,i-1}(z)$ ,  $A_{2,i-1}(z)$  should still be computed. Assume there exists a transfer function  $H_i(z) \in \mathbb{C}^{(n-2)-r \times r}$  such that

$$A_{2,i}(z) = H_i(z)A_{1,i}(z). \quad (5.56)$$

Then, since  $A_{2,i-1}(z) = H_{i-1}(z)A_{1,i-1}(z)$ ,

$$\begin{aligned} \begin{bmatrix} A_{1,i}(z) \\ A_{2,i}(z) \end{bmatrix} &= F_i(z) \begin{bmatrix} A_{1,i-1}(z) \\ A_{2,i-1}(z) \end{bmatrix} = F_i(z) \begin{bmatrix} \mathbf{I} \\ H_{i-1}(z) \end{bmatrix} A_{1,i-1}(z), \\ &= \begin{bmatrix} F_i^{11}(z) & F_i^{12}(z)H_{i-1}(z) \\ F_i^{21}(z) & F_i^{22}(z)H_{i-1}(z) \end{bmatrix} A_{1,i-1}(z), \end{aligned} \quad (5.57)$$

and by equating the two expressions for  $A_{2,i}(z)$

$$H_i(z) [F_i^{11}(z) + F_i^{12}(z)H_{i-1}(z)] = [F_i^{21}(z) + F_i^{22}(z)H_{i-1}(z)], \quad (5.58)$$

then solving for  $H_i(z)$  yields

$$H_i(z) = [F_i^{21}(z) + F_i^{22}(z)H_{i-1}(z)] [F_i^{11}(z) + F_i^{12}(z)H_{i-1}(z)]^{-1}. \quad (5.59)$$

The transfer function  $H_0(z)$  depends on the selected planning algorithm of the lead vehicle and will be discussed later. However, for a given  $H_0(z)$ ,  $H_i(z)$  can be computed. This in turn can be used to compute the full information from  $Y_{i-1}(z)$  by

$$\begin{bmatrix} A_{1,i-1}(z) \\ A_{2,i-1}(z) \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I} \\ 0 & H_{i-1}(z) \end{bmatrix} \underbrace{\Psi_{s,a}^{-1} Y_{i-1}(z)}_{[A_{0,i-1}^\top(z) \quad A_{1,i-1}^\top(z)]^\top}, \quad (5.60)$$

then by pre-multiplying by  $F_i$  and using (5.54)

$$\begin{bmatrix} A_{1,i}(z) \\ A_{2,i}(z) \end{bmatrix} = F_i(z) \begin{bmatrix} 0 & \mathbf{I} \\ 0 & H_{i-1}(z) \end{bmatrix} \Psi_{s,a}^{-1} Y_{i-1}(z), \quad (5.61)$$

$$\begin{aligned} \begin{bmatrix} A_{0,i}(z) \\ A_{1,i}(z) \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 1 \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} K_i(z) F_i(z) \begin{bmatrix} 0 & \mathbf{I} \\ 0 & H_{i-1}(z) \end{bmatrix} \Psi_{s,a}^{-1} Y_{i-1}(z), \\ Y_i(z) &= \underbrace{\Psi_{s,a} \begin{bmatrix} 0 & 0 & 1 \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} K_i(z) F_i(z) \begin{bmatrix} 0 & \mathbf{I} \\ 0 & H_{i-1}(z) \end{bmatrix} \Psi_{s,a}^{-1} Y_{i-1}(z)}_{\Gamma_i(z)}. \end{aligned} \quad (5.62)$$

Then following (5.11), string stability is achieved if

$$\|\Gamma_i(z)\|_{h_\infty} \leq 1, \quad i > 0. \quad (5.63)$$

Note that the transfer function  $\Gamma_i(z)$  of vehicle  $i$  depends on the planning algorithm of all vehicles in front of it (i.e., with indices smaller than  $i$ ) via (5.59). Hence, string stability of vehicle  $i$ , also depends on the planning algorithm of all vehicles in front of it.

### Dependency on Lead Vehicle

Next the transfer function  $H_0(z)$  is derived, which depends on the specific trajectory planning algorithm of the lead vehicle. Since the transfer function  $\Gamma_i(z)$  depends on the entire platoon ahead,  $H_0(z)$  is required to determine the string stability of the platoon at vehicle  $i$ . The control points of a general lead vehicle follow the dynamics

$$\bar{\mathbf{x}}_{0,k} = \bar{\mathbf{A}}_0 \bar{\mathbf{x}}_{0,k-1} + \bar{\mathbf{B}}_0 \mathbf{u}_{r,k-1}, \quad (5.64)$$

where  $\mathbf{u}_{r,k} \in \mathbb{R}^{n-2}$  represent the degrees of freedom related to control points that are not fixed due to continuity constraints. The  $z$ -transform of these free control points is computed by means of

$$\begin{bmatrix} A_{1,0}(z) \\ A_{2,0}(z) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} (z\mathbf{I} - \bar{\mathbf{A}}_0)^{-1} \bar{\mathbf{B}}_0}_{F_0(z)} U_r(z), \quad (5.65)$$

where  $U_r(z) = \mathcal{Z}(\mathbf{u}_{r,k})$ . In the example of the lead vehicle described by (5.27), the lead vehicle is set-up to match the planning of the previous plan step as good as possible. As a result, it does not have an external input (i.e.,  $\bar{\mathbf{B}}_0 = \mathbf{0}$ ). However, in practice the lead vehicle occasionally needs to select a different optimal trajectory in the next time step, due to changing conditions. To take these possible new trajectories into consideration,  $\bar{\mathbf{B}}_0 \neq \mathbf{0}$  can be used to represent these cases. Note that in fact the specific choice for  $\bar{\mathbf{B}}_0$  is not that important, as long as it satisfies the properties listed in Appendix B. Thus it is chosen as

$$\bar{\mathbf{B}}_0 = \begin{bmatrix} \mathbf{0}_{3 \times n-2} \\ \mathbf{I}_{n-2} \end{bmatrix}. \quad (5.66)$$

This ensures that the control points  $\mathbf{a}_{1,i,k}$  and  $\mathbf{a}_{2,i,k}$  can all independently be chosen by the lead vehicle ( $\mathbf{a}_{0,i,k}$  is still fixed due to the continuity constraint). The transfer function  $H_0(z)$  then depends on the specific choice for the lead vehicle planning algorithm, after which  $H_i(z)$  and  $\Gamma_i(z)$  can be computed.

### 5.3.3 Evaluation and Discussion

As an example, a planning algorithm is chosen with  $n = 6$  and  $p = 5$ ,  $t_p/T_i < 0.5$ , such that  $\mathbf{a}_{2,i,k} \in \mathbb{R}^1$ . This is the result of only the basis function  $N_{n,p}(t) = 0, t \in [t_{i,k}, t_{i,k} + t_p]$ , corresponding to the last control point. Now for the lead vehicle of (5.26), the last row of  $\bar{\mathbf{A}}_0$  equals zero, due to the constraint (5.25), where the terminal acceleration is equal to zero. Therefore,  $\bar{a}_{2,i,k} = 0, k > 0$  and for a given  $Y_0(z)$  and  $A_{0,0}(z)$ ,  $A_{1,0}(z)$  can be computed directly. Then  $A_{2,0}(z)$  is computed via  $A_{2,0}(z) = H_0(z)A_{1,0}(z)$ . Since  $A_{2,0}(z) = 0, H_0(z) = 0, \forall z$  can be found, which can then be used to compute  $H_i(z)$  for any  $i$ .

With  $H_0(z)$  of the lead vehicle, it is now possible to compute  $\Gamma_i(z)$  for each vehicle  $i$ , and test whether or not the trajectory planning algorithm is string stable for arbitrary input signals. The numerical case is evaluated with parameters  $h_i = 0.5$  s,  $\forall i$ ,  $T_i = 5$  s,  $\forall i$  and  $t_p = \frac{1}{5}$  s, which are the same parameters as used in Section 3.3.4.

It can then be shown that all following vehicles are in-fact string unstable for given inputs to the lead vehicle. In fact, an example can be shown in which an input for the lead vehicle is used that results in string unstable behavior. To find an input for the lead vehicle, that results in string unstable behavior, the eigenvalues of  $\Gamma_i^*(z)\Gamma_i(z)$  (where  $\Gamma_i^*(z)$  denotes the complex conjugate) can be determined. Specifically, for the presented parameters  $\Gamma_1(1)$  takes values as

$$|\Gamma_1(1)| \approx \begin{bmatrix} 0 & 13.87 & 0.1496 & 21.22 \\ 0 & 1.0 & 0.002293 & 0.6066 \\ 0 & 0.00461 & 0.628 & -24.52 \\ 0 & -1.956 \cdot 10^{-6} & 0.001127 & 0.7754 \end{bmatrix}, \quad (5.67)$$

and which has singular values  $\sigma(\Gamma_1(1)) = [32.502 \quad 10.523 \quad 0.038195 \quad 0]$ , some of which are larger than one. These singular values indicate that there exist inputs to the lead vehicle, such that the platoon behaves string unstable. Then the eigenvector corresponding to the highest eigenvalue of  $\Gamma_1^*(1)\Gamma_1(1)$  is found as  $[0 \quad 0.2961 \quad -0.009667 \quad 0.9551]^T$ . To translate this to an input for the lead vehicle, consider (5.9) and (5.55), from which it can be derived that

$$\tilde{\mathbf{y}}_{i,k} = \Psi_{\tilde{s},a} \begin{bmatrix} \bar{a}_{0,i,k} \\ \bar{\mathbf{a}}_{1,i,k} \end{bmatrix}. \quad (5.68)$$

This is used to find the following direction in which the inputs (i.e., the free acceleration control points) are applied

$$\begin{bmatrix} \bar{\mathbf{a}}_{1,0,k} \\ \bar{a}_{2,0,k} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I} \\ 0 & \mathbf{0} \end{bmatrix} \underbrace{\Psi_{\tilde{s},a}^{-1} \begin{bmatrix} 0 \\ 0.2961 \\ -0.009667 \\ 0.9551 \end{bmatrix}}_{[\bar{a}_{0,0,k} \quad \bar{\mathbf{a}}_{1,0,k}^T]^T}, \quad (5.69)$$

where the final acceleration control points are chosen as zero  $\bar{a}_{2,i,k} = 0$ ,  $\forall k$ . To ensure that string stability can be checked, it is required that  $\|\tilde{s}_0(t)\|_{\mathcal{L}_2}$  is finite. However, a constant input  $\mathbf{u}_{r,k} = \mathbf{u}_r$ ,  $\forall k$  does not satisfy this requirement. Instead, a slow decay of the magnitude of the input is applied, such that the

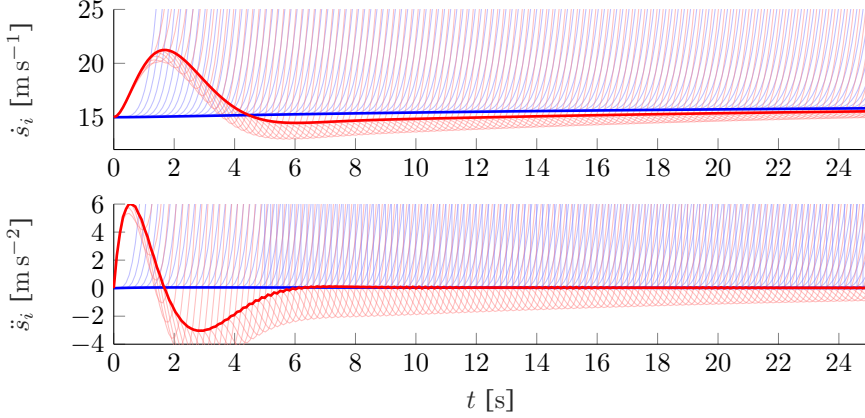


Figure 5.3: String unstable example, with lead vehicle according to (5.70). Lead vehicle trajectory (—), follower vehicle (—), corresponding planned trajectories (—) and (—).

control points of the lead vehicle are found by means of

$$\mathcal{P}_{0,k} = \mathbf{A}_0 \mathcal{P}_{0,k-1} + \underbrace{\Phi^{-1} \begin{bmatrix} \mathbf{0}_{3 \times 4} \\ \mathbf{I}_4 \end{bmatrix}}_{\mathbf{B}_0} \underbrace{\begin{bmatrix} 0 & \mathbf{I} \\ 0 & 0 \end{bmatrix} \Psi_{\ddot{s},a}^{-1} \begin{bmatrix} 0 \\ 0.2961 \\ -0.009667 \\ 0.9551 \end{bmatrix}}_{\mathbf{u}_{r,k}} \beta^k, \quad (5.70)$$

where  $\beta^k$ , with  $\beta < 1$ , ensures that the  $\|\ddot{s}_0(t)\|_{\mathcal{L}_2}$  remains bounded. The sequential communicated trajectories of the lead vehicle are shown in Figure 5.3, for  $\beta = 0.999$  and where the input are scaled by  $10^{-3}$  to ensure that the resulting acceleration is somewhat reasonable in magnitude. It can directly be seen that a huge amplification of the acceleration occurs (to a point where the acceleration of the lead vehicle is hard to distinguish from the figure) and that string stability in the  $\mathcal{L}_2$  sense is not achieved. Although this results seems somewhat discouraging, it must be noted that this input example does not represent an intended use case. Specifically, the inputs of the consecutive plan steps of the lead vehicle are not temporal consistent. Specifically, The light colored lines represent the planned trajectories of each vehicle. Clearly, these are not temporal consistent, as each new plan step is more or less shifted over  $t_p$ .

## 5.4 Temporally Consistent Lead Vehicle

Although the analysis in Section 5.3 suggests that the algorithm is in fact string unstable, the example in which the behavior is string-unstable is actually not very

representative of real world scenarios. By design, the communicated trajectory of the preceding vehicle at time step  $k$ , is actually expected to be close to some *optimal* trajectory. As a result, the trajectory at time step  $k+1$  is expected to again be close to the trajectory planned in time step  $k$ , due to the principle of optimality [138]. The example that has been found in Section 5.3, does not satisfy this criterion, as communicated trajectories in consecutive time steps are not close to one another. In this section, two specific classes of trajectories for the lead vehicle are considered, instead of any arbitrary input signal with bounded  $\mathcal{L}_2$ -norm as was the case in Section 5.3.

### 5.4.1 Lead Vehicle Minimum Jerk

The planning algorithm for the lead vehicle described in Section 5.2.1 by (5.26) with dynamics (5.27) attempts to match its previously planned trajectory as close as possible. However, the trajectory of the first plan step has not been discussed. In Section 3.3.4, minimum jerk maneuvers are used for this planning algorithm. In this section, an analytical solution is presented that corresponds with these scenarios. The lead vehicle that transitions from a given velocity  $v_0$  to a new velocity  $v_T$  for time horizon  $T_0$  can be described by means of B-splines. In fact, this can again be written in the form of (5.17), by writing the integral of the squared jerk in a similar form as the integral of the squared acceleration as (5.7). This results in,

$$\min_{\mathcal{P}_{0,0}} \int_0^{T_0} \ddot{s}_0^2(t) dt = \min_{\mathcal{P}_{0,0}} \mathcal{P}_{0,0}^T \Delta_{30}^T \underbrace{(\mathbf{Q}_{s^2}(T_0) - \mathbf{Q}_{s^2}(0))}_{\Psi_s^T \Psi_s} \Delta_{30} \mathcal{P}_{0,0}, \quad (5.71a)$$

$$\text{s.t.} \quad \begin{bmatrix} s_0 & v_0 & a_0 \\ v_T & a_T \end{bmatrix}^T = \begin{bmatrix} \mathbf{r}_0(0) \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{r}_0(T_0) \end{bmatrix} \mathcal{P}_{0,0}, \quad (5.71b)$$

where  $\mathbf{Q}_{s^2}(t)$  can be computed by means of the approach described in Appendix A and where  $\Psi_s$  exists as  $(\mathbf{Q}_{s^2}(T_0) - \mathbf{Q}_{s^2}(0))$  is positive definite due to the definition of the  $\mathcal{L}_2$ -norm of the jerk. The solution to (5.71) can then be found as

$$\mathcal{P}_{0,0} = [\mathbf{I}_{n+1} \quad 0] \begin{bmatrix} \Delta_{30}^T \Psi_s^T \Psi_s \Delta_{30} & \mathbf{C}_c^T \\ \mathbf{C}_c & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \Delta_{30}^T \Psi_s^T \mathbf{0}_{n-2 \times 1} \\ s_0 \\ v_0 \\ a_0 \\ v_T \\ a_T \end{bmatrix},$$

$$\mathbf{C}_c = \begin{bmatrix} \mathbf{r}_0(0) \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{r}_0(T_0) \end{bmatrix}, \quad (5.72a)$$



The following solution is found for degree  $p = 5$  and number of control points  $n = 6$  as

$$\mathcal{P}_{0,0} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & \frac{1}{10}T_0 & 0 & 0 & 0 \\ 1 & \frac{3}{10}T_0 & \frac{1}{40}T_0^2 & 0 & 0 \\ 1 & \frac{9}{20}T_0 & \frac{1}{15}T_0^2 & \frac{1}{20}T_0 & -\frac{1}{60}T_0^2 \\ 1 & \frac{1}{2}T_0 & \frac{1}{12}T_0^2 & \frac{1}{5}T_0 & -\frac{7}{120}T_0^2 \\ 1 & \frac{1}{2}T_0 & \frac{1}{12}T_0^2 & \frac{2}{5}T_0 & -\frac{1}{12}T_0^2 \\ 1 & \frac{1}{2}T_0 & \frac{1}{12}T_0^2 & \frac{1}{2}T_0 & -\frac{1}{12}T_0^2 \end{bmatrix} \begin{bmatrix} s_0 \\ v_0 \\ a_0 \\ v_T \\ a_T \end{bmatrix}. \quad (5.73)$$

Then the discrete time series of control points, based on initial planning  $\mathcal{P}_{0,0}$  and output related to the  $\mathcal{L}_2$ -norm of the acceleration are described by

$$\mathcal{P}_{0,k} = \mathbf{A}_0^k \mathcal{P}_{0,0}, \quad (5.74)$$

$$\|\ddot{s}_i(t)\|_{\mathcal{L}_2}^2 = \sum_{k=0}^{\infty} \left( \int_0^{t_p} \ddot{s}_i^2(t) dt \right) = \sum_{k=0}^{\infty} \left( \tilde{\mathbf{y}}_{i,k}^T \tilde{\mathbf{y}}_{i,k} \right), \quad (5.75)$$

$$\tilde{\mathbf{y}}_{i,k} = \Psi_{\ddot{s}} \Delta_{20} \mathcal{P}_{i,k}, \quad (5.76)$$

for the lead vehicle of (5.26) that matches the previously planned trajectory as close as possible. Note that

$$\sum_{k=0}^{\infty} \tilde{\mathbf{y}}_{i,k}^T \tilde{\mathbf{y}}_{i,k} = \sum_{k=0}^{\infty} \left( \mathcal{P}_{i,k}^T \Delta_{20}^T \Psi_{\ddot{s}}^T \Psi_{\ddot{s}} \Delta_{20} \mathcal{P}_{i,k} \right), \quad (5.77)$$

$$\sum_{k=0}^{\infty} \tilde{\mathbf{y}}_{0,k}^T \tilde{\mathbf{y}}_{0,k} = \mathcal{P}_{0,0}^T \sum_{k=0}^{\infty} \left( \mathbf{A}_0^T \Delta_{20}^T \Psi_{\ddot{s}}^T \Psi_{\ddot{s}} \Delta_{20} \mathbf{A}_0^k \right) \mathcal{P}_{0,0}. \quad (5.78)$$

Recognise that this resembles the discrete time Observability Gramian  $\mathbf{W}_o = \sum_{k=0}^{\infty} (\mathbf{A}^T)^k \mathbf{C}^T \mathbf{C} \mathbf{A}^k$ , for which solutions can be found [139] as the solution to

$$\mathbf{A}^T \mathbf{W}_o \mathbf{A} - \mathbf{W}_o = -\mathbf{C}^T \mathbf{C}. \quad (5.79)$$

Hence, (5.78) can be found as

$$\sum_{k=0}^{\infty} \tilde{\mathbf{y}}_{0,k}^T \tilde{\mathbf{y}}_{0,k} = \mathcal{P}_{0,0}^T \mathbf{W}_o \mathcal{P}_{0,0}, \quad (5.80)$$

where  $\mathbf{W}_o$  is the unique solution to

$$\mathbf{A}_0^T \mathbf{W}_o \mathbf{A}_0 - \mathbf{W}_o = -\Delta_{20}^T \Psi_{\ddot{s}}^T \Psi_{\ddot{s}} \Delta_{20}. \quad (5.81)$$

Now assume that the string is initially in equilibrium while driving in steady state, such that  $a_i = a_0 = 0$ ,  $\forall i$ ,  $v_i(0) = v_0$ ,  $\forall i$  and

$$s_i(0) = s_{i-1}(0) - h_i \dot{s}_i(0) - L_i - c_i = s_0 - \sum_{j=0}^i (h_j v_0 + c_j + L_j), \quad \forall i. \quad (5.82)$$

Then the control point related to initial acceleration of the lead vehicle is equal to zero:  $a_0 = 0$ . When also assuming that the lead vehicle attempts to achieve a new equilibrium with velocity  $v_T = \bar{v}$  such that  $a_T = 0$ , (5.73) can be written as

$$\mathcal{P}_{0,0} = \begin{bmatrix} 0 & 0 \\ \frac{1}{10} & 0 \\ \frac{3}{10} & 0 \\ \frac{9}{20} & \frac{1}{20} \\ \frac{1}{2} & \frac{1}{5} \\ \frac{1}{2} & \frac{2}{5} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} v_0 \\ \bar{v} \end{bmatrix} T_0 + s_0. \quad (5.83)$$

### Follower Vehicle

For the following vehicle, the dynamics are given by

$$\mathcal{P}_{i,k} = \mathbf{A}_i \mathcal{P}_{i,k-1} + \mathbf{B}_i \mathcal{P}_{i-1,k} + \mathbf{E}_i, \quad k > 0. \quad (5.84)$$

Assume that the string is in equilibrium while driving at a constant velocity, such that  $v_i(0) = v_0$ ,  $\forall i$  and  $s_i(0) = s_0 - \sum_{j=0}^i (h_j v_0 + c_j + L_j)$ ,  $\forall i$ , and  $a_i = a_0 = 0$ ,  $\forall i$ . The steady state assumption for the initial condition allows the initial control points  $\mathcal{P}_{i,0}$  to be related to  $\mathcal{P}_{i-1,0}$ , via

$$\mathcal{P}_{i,0} = \left( \begin{bmatrix} \mathbf{I}_3 & 0 \\ 0 & \mathbf{0}_{n-2} \end{bmatrix} + \mathbf{B}_i \right) \mathcal{P}_{i-1,0} - \begin{bmatrix} \mathbf{1}_{3 \times 1} \\ \mathbf{0}_{n-2 \times 1} \end{bmatrix} (h_i v_0 + c_i + L_i), \quad (5.85)$$

where the first three rows of  $\mathbf{B}_i$  are zero rows. Note that this equation represents that the three initial control points, which fix the initial position, velocity and acceleration, are fixed by means of the assumption of steady state driving in which the spacing policy is satisfied. The remaining control points, which represent the planning of the vehicle for that particular time step, are fixed through the planning algorithm.

Then the output  $\tilde{\mathbf{y}}_{i,k}$  can be found via  $\mathcal{P}_{i,k}$ , which in turn can be found by the solution to the full interconnected system of vehicles. Considering the coordinate transformation in Section 5.3.1, a lumped state vector that includes the states of

all vehicles can be written as

$$\underbrace{\begin{bmatrix} \bar{\mathbf{x}}_{0,k} \\ \bar{\mathbf{x}}_{1,k} \\ \vdots \\ \bar{\mathbf{x}}_{i,k} \end{bmatrix}}_{\bar{\mathbf{x}}_k} = \underbrace{\begin{bmatrix} \bar{\mathbf{A}}_0 & & & \\ \bar{\mathbf{B}}_1 \bar{\mathbf{A}}_0 & \bar{\mathbf{A}}_1 & & \\ \vdots & \vdots & \ddots & \\ \bar{\mathbf{B}}_i \dots \bar{\mathbf{B}}_1 \bar{\mathbf{A}}_0 & \bar{\mathbf{B}}_i \dots \bar{\mathbf{B}}_2 \bar{\mathbf{A}}_1 & \dots & \bar{\mathbf{A}}_i \end{bmatrix}}_{\bar{\mathbf{A}}} \underbrace{\begin{bmatrix} \bar{\mathbf{x}}_{0,k-1} \\ \bar{\mathbf{x}}_{1,k-1} \\ \vdots \\ \bar{\mathbf{x}}_{i,k-1} \end{bmatrix}}_{\bar{\mathbf{x}}_{k-1}}. \quad (5.86)$$

With initial conditions of the individual elements as

$$\bar{\mathbf{x}}_{i,0} = \bar{\mathbf{A}}_i \bar{\mathbf{x}}_{i,-1} + \bar{\mathbf{B}}_i \bar{\mathbf{x}}_{i-1,0}, \quad \bar{\mathbf{x}}_{0,0} = \Phi \mathcal{P}_{0,0} - \begin{bmatrix} \bar{s}_0 \\ \bar{v} \\ \mathbf{0}_{n-1 \times 1} \end{bmatrix}, \quad (5.87)$$

where  $\mathcal{P}_{0,0}$  is according to (5.85). Additionally,  $\bar{\mathbf{x}}_{i,-1}$  is found by means of the assumption that the platoon is in steady state vehicle following with a velocity  $v_0$  for  $k < 0$  and transitioning to the new steady state velocity  $\bar{v}$  for  $k \geq 0$ , according to the state transformation (5.37). Then the following can be written

$$\begin{aligned} \bar{\mathbf{x}}_{i,-1} &= \begin{bmatrix} \bar{s}_{i,-1} \\ \bar{v}_{i,-1} \\ \bar{\mathbf{a}}_{i,-1} \end{bmatrix} = \begin{bmatrix} s_{i,-1} - (\bar{s}_i - t_p \bar{v}) \\ v_0 - \bar{v} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} s_{i,0} - t_p v_0 - (\bar{s}_i - t_p \bar{v}) \\ v_0 - \bar{v} \\ \mathbf{0} \end{bmatrix}, \\ &= \begin{bmatrix} \bar{s}_{0,0} \\ 0 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} -t_p - \sum_{j=1}^i h_j \\ 1 \\ \mathbf{0} \end{bmatrix} (v_0 - \bar{v}), \end{aligned} \quad (5.88)$$

where it is used that  $s_{i,0} = s_{0,0} - \sum_{j=1}^i (h_j v_0 + L_i + c_i)$  and  $\bar{s}_i = \bar{s}_0 - \sum_{j=1}^i (h_j \bar{v} + L_i + c_i)$ . As a result, (5.87) can be written as

$$\begin{aligned} \bar{\mathbf{x}}_{i,0} &= \begin{bmatrix} \bar{s}_{i,0} \\ \bar{v}_{i,0} \\ \bar{\mathbf{a}}_{i,0} \end{bmatrix} = \bar{\mathbf{A}}_i \left( \begin{bmatrix} \bar{s}_{0,0} \\ 0 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} -t_p - \sum_{j=1}^i h_j \\ 1 \\ \mathbf{0}_{n-1 \times 1} \end{bmatrix} (v_0 - \bar{v}) \right) + \bar{\mathbf{B}}_i \bar{\mathbf{x}}_{i-1,0}, \\ &= \begin{bmatrix} \bar{s}_{0,0} \\ 0 \\ \mathbf{0} \end{bmatrix} + \bar{\mathbf{A}}_i \begin{bmatrix} -t_p - \sum_{j=1}^i h_j \\ 1 \\ \mathbf{0}_{n-1 \times 1} \end{bmatrix} (v_0 - \bar{v}) + \bar{\mathbf{B}}_i \bar{\mathbf{x}}_{i-1,0}. \end{aligned} \quad (5.89)$$

and  $\bar{\mathbf{x}}_{0,0}$  is given in (5.87). To eliminate the influence of  $\bar{s}_{0,0}$  on the dynamics of the state vector of the platoon, instead define the lumped state vector for vehicles

0 until  $i$  as,

$$\begin{bmatrix} \bar{s}_{0,k} - \bar{s}_{1,k} - h_1 \bar{v}_{1,k} \\ \bar{v}_{0,k} - \bar{v}_{1,k} \\ \bar{\mathbf{a}}_{0,k} \\ \vdots \\ \bar{s}_{i-1,k} - \bar{s}_{i,k} - h_i \bar{v}_{i,k} \\ \bar{v}_{i-1,k} - \bar{v}_{i,k} \\ \bar{\mathbf{a}}_{i-1,k} \\ \bar{s}_{i,k} \\ \bar{v}_{i,k} \\ \bar{\mathbf{a}}_{i,k} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I} & \chi_1 & 0 & \dots & 0 \\ 0 & \mathbf{I} & \chi_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \mathbf{I} & \chi_{i-1} \\ 0 & \dots & \dots & 0 & \mathbf{I} \end{bmatrix}}_{\tilde{\chi}} \begin{bmatrix} \bar{\mathbf{x}}_{0,k} \\ \bar{\mathbf{x}}_{1,k} \\ \vdots \\ \bar{\mathbf{x}}_{i-1,k} \\ \bar{\mathbf{x}}_{i,k} \end{bmatrix}, \quad (5.90)$$

$$\chi_j = \begin{bmatrix} -1 & -h_j & \mathbf{0} \\ 0 & -1 & \mathbf{0} \\ 0 & 0 & \mathbf{0} \end{bmatrix}.$$

Then by observing the system matrix of the state vector,  $\tilde{\chi} \tilde{\mathbf{A}} \tilde{\chi}^{-1}$ , it can be seen that  $\bar{s}_{i,k}$  and  $\bar{v}_{i,k}$  are not influenced by any of the other vehicle states. Additionally, they do not influence the output of the final vehicle via  $\bar{\mathbf{a}}_{i,k}$ . As a result, the following lumped state vector is used, that removes the states  $\bar{s}_{i,k}$  and  $\bar{v}_{i,k}$ , which results in

$$\tilde{\mathbf{x}}_k = \chi_b \tilde{\chi} \bar{\mathbf{x}}_k, \quad \chi_b = \begin{bmatrix} \mathbf{I}_{(i-1)(n+1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{(n-1)} \end{bmatrix}, \quad (5.91)$$

with associated dynamics

$$\tilde{\mathbf{x}}_k = \underbrace{\chi_b \tilde{\chi} \tilde{\mathbf{A}} \tilde{\chi}^{-1} \chi_b^\top}_{\tilde{\mathbf{A}}} \tilde{\mathbf{x}}_{k-1}. \quad (5.92)$$

With the states  $\bar{s}_{i,k}$  and  $\bar{v}_{i,k}$  removed, the lumped state vector now has stable dynamics, where all eigenvalues of  $\tilde{\mathbf{A}}$  are smaller than one. The outputs for lead vehicle and following vehicles are found via

$$\tilde{y}_{i,k} = \tilde{\mathbf{C}}_i \tilde{\mathbf{x}}_k, \quad \tilde{\mathbf{C}}_i = \begin{cases} \begin{bmatrix} \mathbf{0}_{n-1 \times 2} & \Psi_{\bar{s}} \end{bmatrix}, & \text{if } i = 0, \\ \begin{bmatrix} \mathbf{0}_{n-1 \times i(n+1)} & \Psi_{\bar{s}} \end{bmatrix}, & \text{otherwise,} \end{cases} \quad (5.93)$$

which results in a  $\mathcal{L}_2$  norm of

$$\sum_{k=0}^{\infty} \tilde{\mathbf{y}}_{i,k}^\top \tilde{\mathbf{y}}_{i,k} = \tilde{\mathbf{x}}_0^\top \mathbf{W}_{o,i} \tilde{\mathbf{x}}_0, \quad (5.94)$$

where  $\mathbf{W}_{o,i}$  is the unique solution to

$$\tilde{\mathbf{A}}^\top \mathbf{W}_{o,i} \tilde{\mathbf{A}} - \mathbf{W}_{o,i} = -\tilde{\mathbf{C}}_i^\top \tilde{\mathbf{C}}_i. \quad (5.95)$$

Then by the steady state vehicle following assumption for the initial condition of (5.85) and (5.82), it can be seen that

$$\begin{aligned}\bar{v}_{i-1,0} - \bar{v}_{i,0} &= (v_0 - \bar{v}) - (v_0 - \bar{v}) = 0, \\ \bar{s}_{i-1,0} - \bar{s}_{i,0} - h_i \bar{v}_{i,0} &= (s_{i-1,0} - \bar{s}_{i-1} - 0t_p \bar{v}) - (s_{i,0} - \bar{s}_i - 0t_p \bar{v}) - h_i \bar{v}_{i,0} \\ &= (s_{i-1,0} - s_{i,0}) + (\bar{s}_i - \bar{s}_{i-1}) - h_i \bar{v}_{i,0} \\ &= ((L_i + c_i) + h_i v_{i,0}) + (-(L_i + c_i) - h \bar{v}) - h_i (v_{i,0} - \bar{v}) = 0,\end{aligned}$$

such that

$$\tilde{\mathbf{x}}_0 = [0, 0, \bar{\mathbf{a}}_{0,0}^\top, 0, 0, \bar{\mathbf{a}}_{1,0}^\top, \dots, 0, 0, \bar{\mathbf{a}}_{i-1,0}^\top, \bar{\mathbf{a}}_{i,0}^\top]^\top, \quad (5.96)$$

where  $\bar{\mathbf{a}}_{i,0}$  is found by means of (5.89) and  $\tilde{\mathbf{x}}_0 \propto (v_0 - \bar{v})$  (which can be seen from (5.88) and (5.89)). Hence,  $\tilde{\mathbf{x}}_0$  is independent of  $s_{0,0}$ , and can be written as function of  $(v_0 - \bar{v})$ . Additionally,  $[1, 0, \dots, 0] \bar{\mathbf{a}}_{i,0} = 0$  (i.e., the first element of  $\bar{\mathbf{a}}_{i,0}$  is zero), due to the steady state assumption for the initial condition.

This result can be summarized in the following theorem

**Theorem 5.3.** *Assume a vehicle platoon utilizing a stable trajectory planning method, in which the planned trajectories are constructed by means of B-splines and in which the lead vehicle can be represented by an autonomous system (i.e., without external input). Then the equivalent discrete-time system (5.86), uniquely determines the string-stability properties, in the  $\mathcal{L}_2$ -sense, of the system.*

*Proof.* Using the results of Theorem 5.2 and a coordinate transformation of the control points for all vehicles in the string,  $\tilde{\mathbf{x}}_k$ , with corresponding initial conditions,  $\tilde{\mathbf{x}}_0$ , the dynamics of the lumped system (5.86) can be used to compute the time-series of the discrete-time states.

Using (5.7), the integral of the squared continuous-time acceleration signal of vehicle  $i$  can be computed, by summing over the contribution of each individual planning step  $k$ . The contribution of the  $k$ -th planning step to the infinite integral of the squared continuous-time acceleration signal is quadratic in the control points as described in (5.93).

The  $\mathcal{L}_2$ -norm of the acceleration signal of vehicle  $i$  can then be computed by (5.94) and (5.95), provided that the eigenvalues of reduced lumped system with dynamics (5.92) are all smaller than one.

With the  $\mathcal{L}_2$ -norm of the acceleration signal of vehicle  $i$  and vehicle  $i - 1$ , the string-stability criterion (5.8) can then be tested, which completes the proof.  $\square$

## Numerical Evaluation

The numerical case is evaluated with parameters  $p = 5$ ,  $n = 6$ ,  $h_i = 0.5$  s,  $\forall i$ ,  $T_i = 5$  s,  $\forall i$  and  $t_p = \frac{1}{5}$  s, which are the same parameters as used in

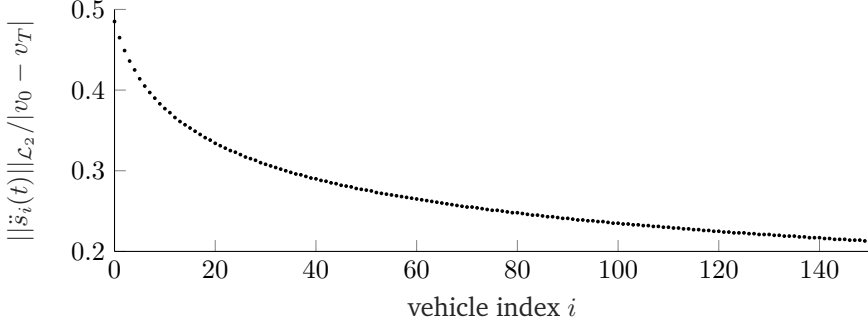


Figure 5.4:  $\mathcal{L}_2$ -norm squared for vehicles  $i$ , following a lead vehicle with dynamics of (5.26) and initial condition (5.73).

Section 3.3.4, Section 5.3.3 and Section 5.4.1. Appendix A can be used to find an exact expression of  $\Psi_{\ddot{s},a}$  (5.55), which is numerically evaluated as

$$\Psi_{\ddot{s},a} \approx \begin{bmatrix} 0.39738 & 0.047376 & 0.0012859 & 0.000013215 \\ 0 & 0.030871 & 0.0012656 & 0.00001565 \\ 0 & 0 & 0.00033314 & 6.8685 \cdot 10^{-6} \\ 0 & 0 & 0 & 1.1737 \cdot 10^{-6} \end{bmatrix}. \quad (5.97)$$

Additionally, it can be seen from (5.88) and (5.89) that  $\bar{\mathbf{a}}_{i,0} \propto (v_0 - \bar{v}), \forall i$ . Specifically, for the numerical example

$$\bar{\mathbf{a}}_{0,0} = \frac{1}{5} \begin{bmatrix} 0 \\ -1 \\ -2 \\ -5 \end{bmatrix} (v_0 - \bar{v}), \quad \bar{\mathbf{a}}_{1,0} = \frac{1}{1532} \begin{bmatrix} 0 \\ -93 \\ -\frac{3103}{5} \\ -377 \end{bmatrix} (v_0 - \bar{v}),$$

$$\bar{\mathbf{a}}_{2,0} = \frac{1}{23470240} \begin{bmatrix} 0 \\ -89621 \\ -7538809 \\ -6902463 \end{bmatrix} (v_0 - \bar{v}),$$

and so on, for vehicles further upstream. Consequently,  $\|\ddot{s}_i(t)\|_{\mathcal{L}_2} \propto (v_0 - \bar{v}), \forall i$ . Then it is possible to plot the norm of  $\|\ddot{s}_i(t)\|_{\mathcal{L}_2}$  proportional to  $(v_0 - \bar{v})$  as function of vehicle index  $i$ , the result of which can be seen in Figure 5.4. The figure shows that indeed the  $\mathcal{L}_2$ -norm of the acceleration decreases over vehicle index  $i$ , such that the string of vehicles using this planning algorithm behaves string stable for a lead vehicle that uses (5.73) to determine the change of velocity, for any initial velocity  $v_0$  and terminal velocity  $\bar{v}$ . Note that based on the figure, this claim can only be made for platoons up to a length of 150 vehicles. However, the trend in the plot suggests that this would also hold for larger platoons, and could be tested

if so desired by the same approach, but at an increased computational cost, due to the larger dimensional observability Gramian  $W_{o,i}$ . The analysis here is limited to 150 vehicles, which for driving at  $\bar{v} = 100$  km/h, with time gap  $h_i = 0.3$  s and assuming a vehicle length of  $L_i = 3$  m and standstill distance  $c_i = 2$  m would span a length of two kilometres, which is typically not very likely to occur in practice. Therefore, the analysis up to 150 vehicles is sufficient for practical application.

### 5.4.2 Lead Vehicle Velocity Controlled Trajectory Planner

A similar analysis can be made for a platoon that follows a lead vehicle with a second type of trajectory planner, described in Section 4.1.1 with discrete-time dynamics described by (5.29). For this lead vehicle the following series of control points are found, with input  $u_k$

$$\mathcal{P}_{0,k} = \mathbf{A}_0 \mathcal{P}_{0,k-1} + \mathbf{B}_0 u_k, \quad (5.98)$$

where  $u_k = v_0$  for  $k < 0$ , and  $u_k = \bar{v}$  for  $k \geq 0$ . Again the same coordinate transformation to state  $\bar{\mathbf{x}}_{i,k}$  can be done as in Section 5.3.1, where for  $k < 0$  the interconnected system is in steady state according to (5.88) and (5.89).

$$\bar{\mathbf{x}}_{i,k} = \mathbf{x}_{i,k} - \begin{bmatrix} \bar{s}_i + kt_p \bar{v} \\ \bar{v} \\ \mathbf{0}_{n-1 \times 1} \end{bmatrix} = \Phi \mathcal{P}_{i,k} - \begin{bmatrix} \bar{s}_i + kt_p \bar{v} \\ \bar{v} \\ \mathbf{0}_{n-1 \times 1} \end{bmatrix}. \quad (5.99)$$

Note that it is indeed valid to again use the coordinate transformation of Section 5.3.1, as the origin (i.e.,  $\bar{\mathbf{x}}_{0,k} = \mathbf{0}_{n+1 \times 1}$ ) is again an equilibrium for  $u_k = \bar{v}$ ,  $k \geq 0$ . This is true since the solution of  $\mathbf{x}_{0,k}$  converges for  $u_k = \bar{v}$  (see also Figure 5.2) to

$$\lim_{k \rightarrow \infty} \mathbf{x}_{0,k} = \begin{bmatrix} \bar{s}_0 + kt_p \bar{v} \\ \bar{v} \\ \mathbf{0}_{n-1 \times 1} \end{bmatrix}. \quad (5.100)$$

As a result, the same coordinate transformation can be used for this case, where the lead vehicle has an input. Similarly, the initial condition for  $\bar{\mathbf{x}}_{0,k}$  can be found using a similar approach as in Section 5.4.1, where

$$\bar{\mathbf{x}}_{0,0} = \bar{\mathbf{A}}_0 \bar{\mathbf{x}}_{0,-1} + \bar{\mathbf{B}}_0 (u_0 - \bar{v}) = \bar{\mathbf{A}}_0 \bar{\mathbf{x}}_{0,-1} + \bar{\mathbf{B}}_0 (\bar{v} - \bar{v}) = \bar{\mathbf{A}}_0 \bar{\mathbf{x}}_{0,-1}. \quad (5.101)$$

Then using (5.88) to see that

$$\bar{\mathbf{x}}_{0,-1} = \begin{bmatrix} \bar{s}_{0,0} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -t_p \\ 1 \\ \mathbf{0}_{n-1 \times 1} \end{bmatrix} (v_0 - \bar{v}), \quad (5.102)$$

$$\bar{\mathbf{x}}_{0,0} = \bar{\mathbf{A}}_0 \bar{\mathbf{x}}_{0,-1} = \begin{bmatrix} \bar{s}_{0,0} \\ 0 \\ 0 \end{bmatrix} + \bar{\mathbf{A}}_0 \begin{bmatrix} -t_p \\ 1 \\ \mathbf{0}_{n-1 \times 1} \end{bmatrix} (v_0 - \bar{v}). \quad (5.103)$$

Then the lumped interconnected system has dynamics for  $k \geq 0$  as

$$\bar{\mathbf{x}}_k = \bar{\mathbf{A}}\bar{\mathbf{x}}_{k-1}, \quad (5.104)$$

where initial conditions for the individual following vehicles are found by means of equation (5.89), but with  $\bar{\mathbf{x}}_{0,0}$  according to (5.103). To eliminate the effect of  $\bar{s}_{0,0}$ , a second coordinate transformation is done, which is similar to (5.90), with the exception that the state  $v_i$  is not omitted. Hence,  $\tilde{\mathbf{x}}_k$  is now defined as

$$\tilde{\mathbf{x}}_k = \chi_b \tilde{\chi} \bar{\mathbf{x}}_k, \quad \chi_b = \begin{bmatrix} \mathbf{I}_{(i-1)(n+1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_n \end{bmatrix}, \quad (5.105)$$

where the lower right identity matrix is of size  $n$  rather than  $n-1$ . The associated dynamics  $\tilde{\mathbf{A}}$  are given by (5.92), with initial condition similar to (5.106) but with  $\bar{v}_{i,0} = (v_0 - \bar{v})$  included, i.e.,

$$\tilde{\mathbf{x}}_0 = [0, 0, \bar{\mathbf{a}}_{0,0}^\top, 0, 0, \bar{\mathbf{a}}_{1,0}^\top, \dots, 0, 0, \bar{\mathbf{a}}_{i-1,0}^\top, (v_0 - \bar{v}), \bar{\mathbf{a}}_{i,0}^\top]^\top. \quad (5.106)$$

The output is found by means of

$$\tilde{\mathbf{y}}_{i,k} = \tilde{\mathbf{C}}_i \tilde{\mathbf{x}}_k, \quad \tilde{\mathbf{C}}_i = \begin{cases} \begin{bmatrix} \mathbf{0}_{n-1 \times 2} & \Psi_{\tilde{s}} \end{bmatrix}, & \text{if } i = 0, \\ \begin{bmatrix} \mathbf{0}_{n-1 \times i(n+1)+1} & \Psi_{\tilde{s}} \end{bmatrix}, & \text{otherwise,} \end{cases} \quad (5.107)$$

and the corresponding squared  $\mathcal{L}_2$ -norm is found by (5.94).

### Numerical Evaluation

String stability for a platoon of vehicles, following a lead vehicle described by (5.29) subjected to a step input, can be assessed using the method described above. Again the same parameters as in Section 3.3.4 and Section 5.3.3 are chosen:  $p = 5$ ,  $n = 6$ ,  $h_i = 0.5$  s,  $\forall i$ ,  $T_i = 5$  s,  $\forall i$  and  $t_p = \frac{1}{5}$  s. Appendix A is used to find  $\Psi_{\tilde{s}}$  and is given numerically by

$$\Psi_{\tilde{s}} = \begin{bmatrix} 0.39738 & 0.047376 & 0.0012859 & 0.000013215 & 0 \\ 0 & 0.030871 & 0.0012656 & 0.00001565 & 0 \\ 0 & 0 & 0.00033314 & 6.8685 \cdot 10^{-6} & 0 \\ 0 & 0 & 0 & 1.1737 \cdot 10^{-6} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (5.108)$$

All initial conditions in  $\bar{\mathbf{x}}_{0,k}$  are proportional to  $(v_0 - \bar{v})$ , such that the analysis can be made independently of initial and final velocity. The results can be found in Figure 5.5, which shows that indeed the  $\mathcal{L}_2$ -norm of the acceleration  $\ddot{s}_i(t)$  decreases in the upstream direction  $\|\ddot{s}_i(t)\|_{\mathcal{L}_2} \leq \|\ddot{s}_{i-1}(t)\|_{\mathcal{L}_2}$ .



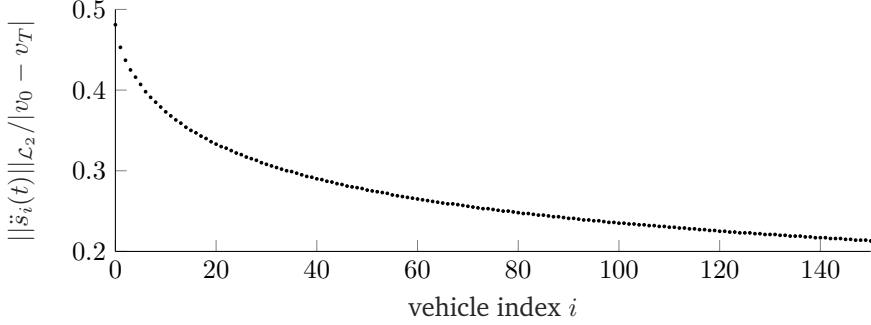


Figure 5.5:  $\mathcal{L}_2$ -norm squared for vehicles  $i$ , following a lead vehicle described by (5.29) for input  $u_k = v_0$ ,  $k < 0$  and  $u_k = \bar{v}$ ,  $k \geq 0$ .

### 5.4.3 Planning Delay for Following Vehicles

The discrete time setting does not allow for easily inclusion of an arbitrary planning delay (e.g., the time between the planning of a trajectory of vehicle  $i - 1$  and the planning of a trajectory of vehicle  $i$  where that information is used). However, a delay  $\zeta_i$  with a value equal to an integer multiple of the planning interval  $t_p$

$$\zeta_i = j_i t_p, \quad j_i \in \mathbb{N}^+, \quad (5.109)$$

can be used in this setting. In this case, the dynamics of the follower vehicles are adjusted as,

$$\mathcal{P}_{i,k} = \mathbf{A}_i \mathcal{P}_{i,k-1} + \mathbf{B}_i \mathcal{P}_{i-1,k-j} + \mathbf{E}_i, \quad (5.110)$$

where  $\mathbf{A}_i$  and  $\mathbf{E}_i$  are identical to (5.33), and  $\mathbf{B}_i$  is an adjusted variant where the arguments of  $\mathbf{r}(t)$  are offsetted by  $\zeta_j$

$$\mathbf{A}_i = [\mathbf{I}_{n+1} \quad \mathbf{0}] \begin{bmatrix} \Omega^\top \Omega & \mathbf{r}_i(0)^\top \\ \mathbf{r}_i(0) & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{r}_0(t_p) \end{bmatrix}, \quad (5.111a)$$

$$\mathbf{E}_i = [\mathbf{I}_{n+1} \quad \mathbf{0}] \begin{bmatrix} \Omega^\top \Omega & \mathbf{r}_i(0)^\top \\ \mathbf{r}_i(0) & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \Omega^\top \mathbf{1}_{n-2 \times 1}(-c_i - L_i) \\ \mathbf{0} \end{bmatrix}, \quad (5.111b)$$

$$\mathbf{B}_i = [\mathbf{I}_{n+1} \quad \mathbf{0}] \begin{bmatrix} \Omega^\top \Omega & \mathbf{r}_i(0)^\top \\ \mathbf{r}_i(0) & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \Omega^\top \begin{bmatrix} [1 \ 0 \ 0] \mathbf{r}_{i-1}(\mu_3 + \zeta_i) \\ \vdots \\ [1 \ 0 \ 0] \mathbf{r}_{i-1}(\mu_n + \zeta_i) \\ \mathbf{0} \end{bmatrix}^\top \end{bmatrix}. \quad (5.111c)$$

Note that in the scenario with time delay (5.109), the vehicles start to react on a change in velocity of the lead vehicle in different plan steps, due to the delay of

information. Assuming the platoon is in steady state for  $k < 0$ , the lead vehicle  $i = 0$ , adjusts its planning at  $k = 0$  (and the ‘delay’ of vehicle  $i = 0$  is set to  $j_0 = 0$ ), then the first following vehicle,  $i = 1$  reacts after  $j_1$  plan steps (with  $j_1$  from (5.109)), vehicle  $i = 2$  reacts another  $j_2$  steps later on the lead vehicle, and so on. Hence, vehicle  $i$  bases its planning at plan step  $k$  on information of the preceding vehicle of  $k - j_i$ . As a result, information of a change in velocity of the lead vehicle at  $k = 0$  will be included in plan step  $k = \sum_{l=0}^i j_l$  of vehicle  $i$ . Since vehicle  $i$  is not accelerating in the first  $k < \sum_{l=0}^i j_l$  plan steps, these trajectories do not contribute to the  $\mathcal{L}_2$ -norm of the acceleration. Now assume a homogeneous delay along the entire length of the string for simplicity (i.e.,  $j_i = j, \forall i$  and  $\zeta = jt_p$ ), such that

$$\|\ddot{s}_i(t)\|_{\mathcal{L}_2} = \sum_{k=0}^{\infty} \left( \mathcal{P}_{i,k}^T \Delta_{20}^T \Psi_s^T \Psi_s \mathcal{P}_{i,k} \right) = \sum_{k=i \cdot j}^{\infty} \left( \mathcal{P}_{i,k}^T \Delta_{20}^T \Psi_s^T \Psi_s \mathcal{P}_{i,k} \right). \quad (5.112)$$

Instead of using the initial control points (5.85) (as is used for the case without planning delay), the control points of follower vehicle  $i$  can be computed for time step  $k = i \cdot j$  as

$$\mathcal{P}_{i,i \cdot j} = \left( \begin{bmatrix} \mathbf{I}_3 & 0 \\ 0 & \mathbf{0}_{n-2} \end{bmatrix} + \mathbf{B}_i \right) \mathcal{P}_{i-1,i \cdot j - j} - \begin{bmatrix} \mathbf{1}_{3 \times 1} \\ \mathbf{0}_{n-2 \times 1} \end{bmatrix} ((h_i - \zeta_i)v_0 + c_i + L_i), \quad (5.113)$$

where the term  $\zeta_i v_0$  relates to the distance travelled during the delay. This is only found in the first three control points, that are fixed by means of continuity. Again a similar coordinate transformation can be made

$$\underbrace{\begin{bmatrix} \bar{\mathbf{x}}_{0,k} \\ \bar{\mathbf{x}}_{1,k+j} \\ \vdots \\ \bar{\mathbf{x}}_{i,k+i \cdot j} \end{bmatrix}}_{\bar{\mathbf{x}}_k} = \underbrace{\begin{bmatrix} \bar{\mathbf{A}}_0 & & & \\ \bar{\mathbf{B}}_1 \bar{\mathbf{A}}_0 & \bar{\mathbf{A}}_1 & & \\ \vdots & \vdots & \ddots & \\ \bar{\mathbf{B}}_i \dots \bar{\mathbf{B}}_1 \bar{\mathbf{A}}_0 & \bar{\mathbf{B}}_i \dots \bar{\mathbf{B}}_2 \bar{\mathbf{A}}_1 & \dots & \bar{\mathbf{A}}_i \end{bmatrix}}_{\bar{\mathbf{A}}} \underbrace{\begin{bmatrix} \bar{\mathbf{x}}_{0,k-1} \\ \bar{\mathbf{x}}_{1,k+j-1} \\ \vdots \\ \bar{\mathbf{x}}_{i,k+i \cdot j-1} \end{bmatrix}}_{\bar{\mathbf{x}}_{k-1}}, \quad (5.114)$$

such that  $\bar{\mathbf{x}}_k$  consists of time shifted state vectors,  $\bar{\mathbf{x}}_{i,i \cdot j}$ , of the vehicles in the platoon. As a result,  $\bar{\mathbf{x}}_k$  contains the states of the vehicles that have become aware of the change at plan step  $k$  of the lead vehicle. The ‘initial’ conditions  $\bar{\mathbf{x}}_{i,i \cdot j}$  are again found by considering that the platoon is initially in steady state at velocity  $v_0$ , and ends in steady state with velocity  $\bar{v}$ . Note that (5.88), still holds to determine  $\bar{\mathbf{x}}_{i,0}$ . However, in this case the initial condition of interest is  $\bar{\mathbf{x}}_{i,i \cdot j}$  instead of  $\bar{\mathbf{x}}_{i,0}$ . Hence,  $\bar{\mathbf{x}}_{i,i \cdot j}$  needs to be computed. Note that each following vehicle remains driving at  $v_0$  until  $k = i \cdot j$ , such that even for  $k = i \cdot j$  expression

(5.88) can be used.

$$\bar{\mathbf{x}}_{i,i \cdot j-1} = \begin{bmatrix} \bar{s}_{0,0} \\ 0 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} (i \cdot j - 1)t_p - \sum_{j=1}^i h_j \\ 1 \\ \mathbf{0}_{n-1 \times 1} \end{bmatrix} (v_0 - \bar{v}), \quad (5.115)$$

$$\bar{\mathbf{x}}_{i,i \cdot j} = \bar{\mathbf{A}}_i \bar{\mathbf{x}}_{i,i \cdot j-1} + \bar{\mathbf{B}}_i \bar{\mathbf{x}}_{i,i \cdot j-j}. \quad (5.116)$$

Additionally,  $\bar{\mathbf{x}}_{0,0}$  is computed by the method described in Section 5.4.1 or Section 5.4.2, depending on which type of planner the lead vehicle in front of the platoon has.

Note that a slightly different coordinate transformation should be used, as the states associated with  $\bar{s}_{i,k}$  in (5.90), no longer converge to zero. To see this, note that  $\bar{s}_{i-1,k+(i-1)j} - \bar{s}_{i,k+ij} - h_i \bar{v}_{i,k+ij}$  in this new state vector of (5.114), which does not converge to zero, is analogous to  $\bar{s}_{i-1,k} - \bar{s}_{i,k} - h_i \bar{v}_{i,k}$  in (5.90). Instead, the following coordinate transformation is used

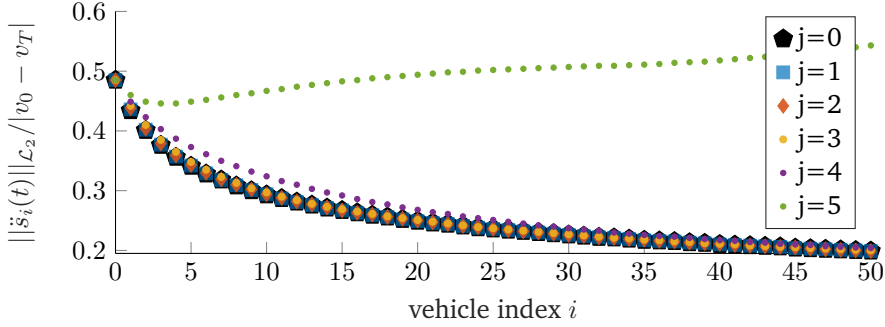
$$\begin{bmatrix} \bar{s}_{0,k} - \bar{s}_{1,k+j} - (h_1 - \zeta_1) \bar{v}_{1,k+j} \\ \bar{v}_{0,k} - \bar{v}_{1,k+j} \\ \bar{\mathbf{a}}_{0,k} \\ \vdots \\ \bar{s}_{i-1,k+(i-1)j} - \bar{s}_{i,k+ij} - (h_i - \zeta_i) \bar{v}_{i,k+ij} \\ \bar{v}_{i-1,k+(i-1)j} - \bar{v}_{i,k+ij} \\ \bar{\mathbf{a}}_{i-1,k+(i-1)j} \\ \bar{s}_{i,k+ij} \\ \bar{v}_{i,k+ij} \\ \bar{\mathbf{a}}_{i,k+ij} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I} & \chi_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \chi_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & & \ddots & \mathbf{I} & \chi_{i-1} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\tilde{\chi}} \bar{\mathbf{x}}_k$$

$$\chi_j = \begin{bmatrix} -1 & -(h_j - \zeta_j) & \mathbf{0} \\ 0 & -1 & \mathbf{0} \\ 0 & 0 & \mathbf{0} \end{bmatrix}, \quad (5.117)$$

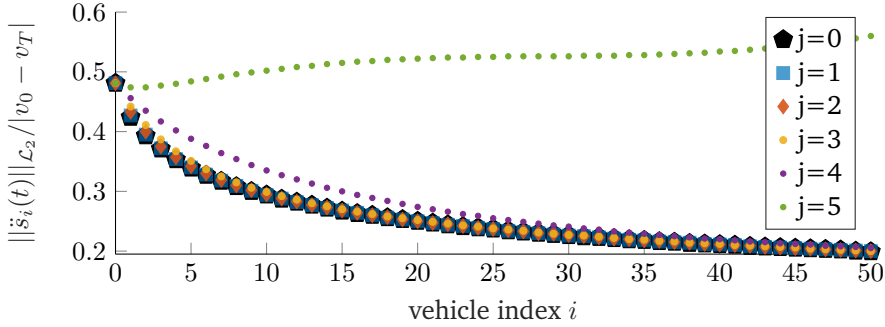
with  $\bar{\mathbf{x}}_k$  as defined in (5.114) and where  $\chi_j$  is only equal to that of (5.90), in case the delay  $\zeta_i = 0$ . Note that this highlights that without the delay compensation, the spacing policy is not satisfied, even in steady state. The dynamics are then again given by (5.92). Depending on which lead vehicle is used  $\chi_b$  is identical to either (5.96) or (5.106), and the output matrix  $\tilde{\mathbf{C}}_i$  is given by either (5.93) or (5.107).

## Numerical Evaluation

Using this approach, the propagation of the  $\mathcal{L}_2$ -norm over the string can be assessed for scenarios with a planning delay. Again the same example as used in Section 5.4.1 and Section 5.4.2 is chosen, with parameters:  $p = 5$ ,  $n = 6$ ,  $T_i = 5$  s,  $\forall i$  and  $t_p = \frac{1}{5}$  s. The exception is the time gap, which is now taken as  $h_i = 1$  s,  $\forall i$ , for a reason that will become apparent in the result.



(a) following a lead vehicle described by (5.26).



(b) following a lead vehicle described by (5.26).

Figure 5.6:  $\mathcal{L}_2$ -norm squared for vehicles  $i$ , , with delay  $\zeta_i = jt_p$ , and time gap  $h_i = 1, \forall i$ .

The propagation of the  $\mathcal{L}_2$ -norm over the string can be illustrated, similar to Figure 5.4 and Figure 5.5, involving the minimum jerk trajectory tracking lead vehicle and velocity controlled lead vehicle, respectively. A slightly smaller string of 50 vehicle is used, to speed up computations. The results for this scenario with planning delay can be seen in Figure 5.6a and Figure 5.6b, for which the case without plan delay can be found in Figure 5.4 and Figure 5.5 respectively. It can be seen that for both lead vehicles string stability is achieved up to a delay of  $\zeta = 4t_p$ . Moreover, it can also be seen that for both lead vehicles, the  $\mathcal{L}_2$ -norm propagation is very similar. So for two different lead vehicles, with different planning algorithms, the string of vehicles behaves in a very similar way. It should be noted that despite the planning algorithms being different, both can be considered as approximately temporal consistent.

To find string stable combinations of delay  $\zeta_i$  and  $h_i = h, \forall i$ , a bisection search is done, in which  $h$  can be increased or decreased for a fixed  $\zeta_i = \zeta, \forall i$ , until the system is just string stable with a tolerance of  $\epsilon = 10^{-4}$  in  $h$ , for both the delay compensated algorithm as well as the non-compensated case. This is similar to the analysis of Section 3.3.4, but instead of varying the delay, now the time gap is varied. The results can be found in Figure 5.7, which is slightly more

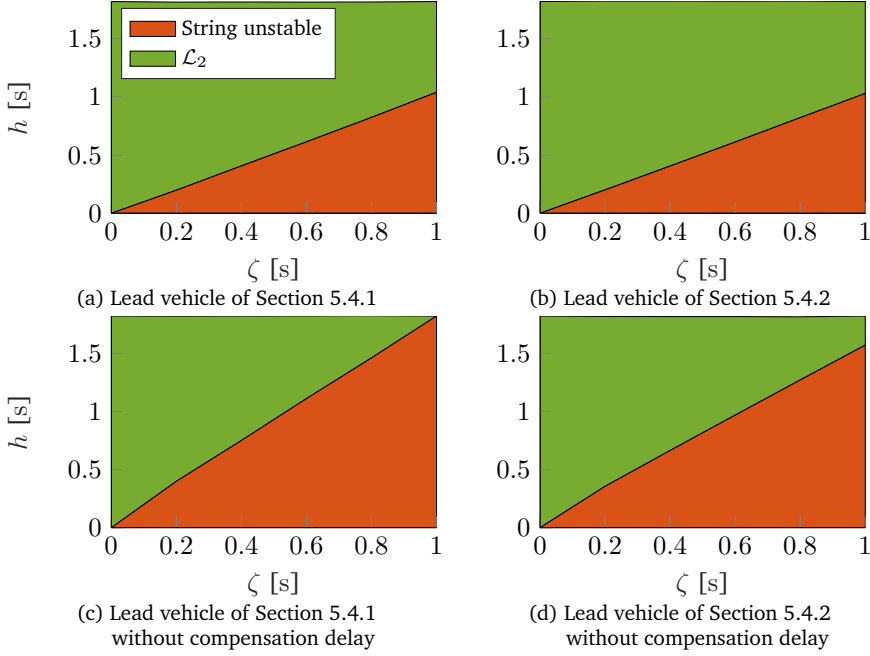


Figure 5.7: Bisection search to find  $h_i$  that results in string stable behavior for a string of 50 vehicles. Search is performed at  $\zeta_i = jt_p$ ,  $j \in \{0, \dots, 5\}$  and  $t_p = 0.2$  s.

optimistic (i.e., requiring a lower  $h$  for the same delay) than the analysis used in Section 3.3.4, which can be explained by the fact that here, only transitions to a new velocity are considered, whereas before, also cases where an initial perturbation or a random acceleration input for the first plan step are considered.

Interestingly, this method also allows to check the usefulness of delay compensation in (5.112) (i.e., the addition of  $\zeta_j$  in the argument of  $\Upsilon_{i-1}(t)$ ). Without this delay compensation, also (5.117) is adjusted (e.g.,  $\zeta_j$  in  $\chi_j$  is removed) to ensure that the dynamics of  $\tilde{\mathbf{x}}_k$  again converge to zero. In addition, it also affects the minimum  $h_i$  required for attenuation of the  $\mathcal{L}_2$ -norm of the acceleration signal. The results of the bisection search are illustrated in Figure 5.7, which shows that a significantly higher time gap  $h_i$  is required to attenuate the  $\mathcal{L}_2$ -norm of the acceleration signal in the upstream direction of the platoon. Clearly, being aware of the delay has significant benefits, as was also demonstrated in [140].

## 5.5 Summary and Discussion

In summary, this chapter considers string stability using a more rigorous mathematical approach. First, it is demonstrated that the planning algorithm with B-splines can be described by discrete dynamics. These dynamics describe how the control points of plan step  $k$ , are related to those of plan step  $k - 1$  and the control points of the preceding vehicle of plan step  $k$ . Although the dynamics of these control points are in discrete time (related to the discrete update intervals of the planning), the signals of interest are still defined in continuous time. Specifically, the acceleration signal of each vehicle is still continuous. It is shown that the  $\mathcal{L}_2$ -norm of this continuous-time signal can be described by the output of the discrete time system.

The discrete-time description of the trajectory planner is then used to consider string stability in the frequency domain, for arbitrary input to the lead vehicle. As a result, the planning algorithm appears to yield string unstable behavior. When finding a specific example to demonstrate this string unstable behavior, it is found that this occurs when the lead vehicle consistently communicates a planned trajectory, but then consistently does not execute it (i.e., temporally inconsistent planning). However, these cases are less relevant, as vehicles should in general be expected to execute a trajectory that is at least similar to the communicated planning.

Therefore, also inputs to the lead vehicle that describe a single change in velocity are considered. This can either be with the lead vehicle whose trajectory matches that of the previous plan step and the velocity-controlled lead vehicle, both of which are approximately temporal consistent. An exact expression for the  $\mathcal{L}_2$ -norm of the acceleration is then derived, which is used to consider vehicle following for both lead vehicles. It is demonstrated, that a string of vehicles following either lead vehicle is string stable for any initial and terminal velocity. Although the planning algorithm is designed to handle arbitrary planning delay, the discrete-time description only allows for delays of integer multiples of the update time, as synchronous planning for all vehicles is assumed. It can be shown that in the case of such delays, the discrete-time description is still valid, and an exact expression for the  $\mathcal{L}_2$ -norm can still be found. This is used to first demonstrate how this delay influences the attenuation of acceleration in the string. Next, a bisection search is used to find for the minimum string-stable time gap for a given delay. Again it is shown that even in the case of large planning delays, string stability is retained. It is demonstrated that being aware of the planning delay helps to reduce the required time gap for attenuation. Additionally, it is shown that being aware of the planning delay is necessary to satisfy the spacing policy in steady state driving.

Interestingly, there is thus a difference in behavior for temporal inconsistent and approximate temporal consistent lead vehicles. Clearly, for a single velocity change for lead vehicles, the string behaves string stable, whereas this is not the

case for arbitrary inputs to the lead vehicle. This supports the design requirement of temporal consistency used in Chapter 3. Note however, that once the platoon has converged to the new steady state velocity, a second velocity change could be performed, which would again be string stable. This suggests that there might be a certain dwell time (related to the convergence rate of the interconnected system) for a switch in the input to the lead vehicle, which could still result in string stable behavior. This would be an interesting topic for follow up research. Nevertheless, the developed algorithm showed to be string stable for single velocity changes with approximate temporally consistent planners, which in itself is a very useful property.

# Chapter 6

## Experimental Implementation

The framework described in Chapter 3 and Chapter 4 enables an automated vehicle to operate both autonomously as well as cooperatively. This framework was shown in simulations to be versatile in traffic scenarios and capable of ad-hoc forming and braking of platoons. Additionally, Chapter 5 presented a more rigorous string stability analysis, which shows that for temporal consistent lead vehicles, the resulting cooperative trajectories are indeed string stable. Since simulations and rigorous analysis both show great potential for the developed framework, the next step is to implement the framework on an experimental setup. This chapter discusses the implementation of the framework on two full-scale vehicles, equipped with a real-time processing unit to host the framework. First, in Section 6.1, the experimental setup is described. Next, Section 6.2 discusses all practical details that are involved with real-time implementation of the framework. Section 6.3 and Section 6.4 present the experimental results of the lead vehicle trajectory planner of Chapter 4 and the cooperative trajectory planner of Chapter 3 respectively.

### 6.1 Experimental Setup

The experiments are performed using two Renault Twizy's that are automated as part of the i-CAVE project [52], which are illustrated in Figure 6.1. This section discusses both the Twizy platform as well as the instrumentation that is used. These small electric vehicles are equipped with a 13 kW electric drive train. A high fidelity model for this vehicle was developed in [141], which covers a detailed torque map for both regenerative braking as well as driving. It should be noted that these torque maps are not those as created by the OEM. Instead, they have been modified to increase the usability of the drivetrain for automated driving, by increasing the available torque in both regular driving as well as regenerative





Figure 6.1: Two automated Renault Twizy's of the i-CAVE project

use.

These Renault Twizy's have been equipped with actuation and instrumentation to facilitate cooperative and autonomous driving. The vehicles are equipped with real-time computers, which are programmed by means of Simulink real-time. The the control system running at these computers is updated at a frequency of 100 Hz. These real-time computers that have access to the vehicles native CAN-bus that provides the control system with the vehicle's velocity by means of motor RPM. A Bosch mm5.10 IMU is used for acceleration and yaw rate measurements. A u-blox EVK-M8T GPS system provides the vehicles with a means of localization as well as a common clock. A Bosch MRRevo14F radar sensor is used for distance and relative velocity measurements. Two Wi-Fi ITS-G5 routers from Rendits Routers [142] are used for V2V communication, which enable the Simulink platform to transmit ITS-G5 messages, such as the CAM [143] and DENM messages [144], between the vehicles. Additionally, these Rendits routers are capable to transmit a custom message that can be used to communicate arbitrary information between the vehicles. This custom message is used to communicate the coefficients of the planned trajectory  $\mathcal{P}_{i,k}$  and  $\mathcal{R}_{i,k}$ , as well as the planning time  $t_{i,k}$  and time horizon  $T_i$ . An overview of the sensors used specifically for the experiments presented in this chapter can be found in Table 6.1. A more elaborate description of these vehicles and the additional instrumentation can be found in [65], which also discusses the state estimation that is used for host tracking, together with a

Table 6.1: Vehicle sensors of the i-CAVE Renault Twizy's, used for the implementation of the framework presented in this thesis.

Sensor Type	Sensor used on Twizy	Signals
Acceleration	Bosch mm5.10 IMU [145]	3DOF Acceleration Yaw Rate
GPS	u-blox EVK-M8T [146]	Longitude Latitude Heading Speed over ground Clock
Vehicle Sensors	Native CAN-bus	Motor RPM
Steering Sensor	Bosch LWS3 [147]	Steering Angle Steering Rate
Front Radar	Bosch MRRevo14F Radar [148]	Distance Angle vertical position Height Class
V2X communication	Rendits Router [142]	iCLCM-Message [149] DENM Message [144] CAM Message [143] Custom Message

more accurate description of all the sensors.

## 6.2 Implementation Details

The framework described in Chapter 4, requires the host vehicle to be able to position itself within a map. This map can either be loaded a-priori or received wirelessly, if such a map exists for the environment. Additionally, it can be constructed in real-time by means of the computer vision system. When the vehicle is capable of positioning itself and obstacles in such a map, it can then construct trajectories, as described in Chapter 4. Although the Renault Twizy's used in the i-CAVE project are equipped with steering actuator, steering angle and steering angle velocity sensor, and are capable of making controlled steering movements, the capabilities of the computer vision system are at the time of writing extremely limited making the construction of Highly-Automated-Driving maps (HAD maps) by means of computer vision infeasible. In addition, no HAD map is available for the testing locations, making positioning by means of GPS within a known map equally infeasible. Therefore, the implementation of the trajectory planner is limited to longitudinal driving, while lateral positioning is done by the driver. Recall that the planning algorithms of Chapters 3 and 4 plan

trajectories in curvilinear coordinate  $s$  for vehicle  $i$  as  $s_{i,k}(t)$ ,  $t \in [t_{i,k}, t_{i,k} + T_i]$ . These trajectories are planned at times  $t_{i,k}$ , span over a horizon  $T_i$  and are updated after a time of  $t_p$ . The planned trajectory are constructed by means of B-splines as

$$s_{i,k}(t) = \mathcal{B}_{n,p}(t - t_{i,k}) \mathcal{P}_{i,k}, \quad (6.1)$$

$$\dot{s}_{i,k}(t) = \mathcal{B}_{n-1,p-1}(t - t_{i,k}) \Delta_{10} \mathcal{P}_{i,k}, \quad (6.2)$$

$$\ddot{s}_{i,k}(t) = \mathcal{B}_{n-2,p-2}(t - t_{i,k}) \Delta_{20} \mathcal{P}_{i,k}, \quad (6.3)$$

where  $\mathcal{B}_{n,p}(t - t_{i,k})$  are basis functions of degree  $p$ , and control points  $\mathcal{P}_{i,k} \in \mathbb{R}^{n+1}$  and  $\Delta_{10}$  the hodograph relation according to (2.48).

### 6.2.1 Integration of Radar Information

The framework presented in Chapter 3 assumes that an accurate position planning,  $s_{i-1,k}(t)$ , of the rear bumper of the preceding vehicle (see also Figure 3.1 for reference) over the time horizon  $t \in [t_{i-1,k}, t_{i-1,k} + T_{i-1}]$  is communicated. It uses this planning to construct its own trajectory  $s_{i,k}(t)$  over the time horizon  $t \in [t_{i,k}, t_{i,k} + T_i]$ . However, in order to communicate an accurate planning, an accurate estimate of the curvilinear position (see Figure 2.2) is essential. If either the estimated position of the host vehicle or the preceding vehicle is inaccurate, or not defined in the same reference frame (i.e.,  $s_i = 0$  does not define the same position in the world frame as  $s_{i-1}(t)$ ), then the resulting planned trajectory might not achieve the desired spacing policy. To solve this issue, the on-board radar can be used in conjunction with the planned trajectories obtained by means of vehicle-to-vehicle (V2V) communication. Denote the communicated trajectory of the preceding vehicle by  $\tilde{s}_{i-1,k}(t)$ ,  $t \in [t_{i-1,k}, t_{i-1,k} + T_{i-1}]$ , then the corrected trajectory is constructed by means of the inter-vehicle distance  $d_i(t_{i,k})$ , measured by the on-board front facing radar of vehicle  $i$  as

$$s_{i-1,k}(t) = \tilde{s}_{i-1,k}(t) - (\tilde{s}_{i-1,k}(t_{i,k}) - s_i(t_{i,k})) + (d_i(t_{i,k}) + L_i), \quad t \in [t_{i-1,k}, t_{i-1,k} + T_{i-1}]. \quad (6.4)$$

This ensures that the inter-vehicle distance  $s_{i-1,k}(t_{i,k}) - s_{i,k}(t_{i,k})$  corresponds with the measured inter-vehicle distance  $d_i(t_{i,k})$ , and can be used to achieve the desired spacing policy. The correct inter-vehicle distance  $d_i(t)$  should be extracted from the radar measurements.

### 6.2.2 Most Important Object (MIO) Identification

Radar measurements are thus used to increase robustness of the system when compared to only using V2V communication. However, it is important to relate

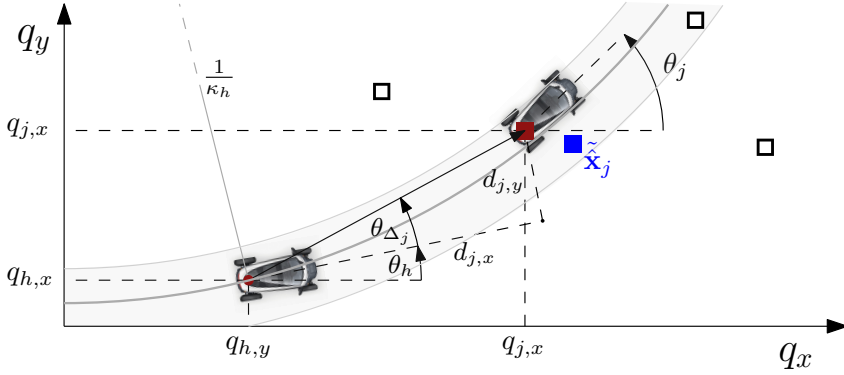


Figure 6.2: Schematic representation of MIO selection. Detected obstacles ( $\square$ ), MIO ( $\blacksquare$ ), V2V object position  $\tilde{x}_j$  ( $\blacksquare$ ).

the radar measurement with the corresponding messages of other vehicles. These vehicles can either be driving in the same lane as the host vehicle, or could be driving on neighbouring lanes, as for example was presented in Chapter 4. Since communicated trajectories include a specification on the planned lateral position  $\ell_{i,k}(t)$ , this can be used in conjunction with the computer vision system, which can be used to identify which lane (and hence which lateral coordinate  $\ell$ ) is associated with each measurement. However, the objective during the experimental validation is to follow a single vehicle (as only two equipped vehicles are operational). Therefore, instead of mapping multiple communicated messages, to multiple detections, only a single object needs to be identified, which is referred to as the Most Important Object (MIO). In this section, the selection of the MIO is presented.

Since the computer vision system of the experimental set-up is not operational at the time of writing, accurate localization relative to the road of the host vehicle and objects is difficult. Hence, the lateral component of the driving task is completely performed by the operator. This, in combination with the objective of only following a single preceding vehicle, significantly simplifies the task at hand of matching detections with communicated planned trajectories of other road users. Specifically, it reduces the objective of identifying a single preceding vehicle, which is approximately in front of the host vehicle, among detected objects. The selected method makes use of a *driving lane*, to create a region of interest, in combination with the information of V2V communication. This method is partly based on that of [65], which filters out stationary objects such as parked cars alongside of the road that are not valid objects to follow. However, since in this work, the host vehicle should also be capable of stopping behind a stationary vehicle, such as done in Section 3.4. Therefore, standstill vehicles need to be considered and cannot be filtered out. As a result, a different approach is required.

The Bosch MRR radar that is used consists of multiple beams and provides the real-time computer with object level data of up to 32 obstacles. The use of object level data, rather than a point cloud of detections, makes implementation significantly simpler. However, multiple objects are still detected. Hence, it is necessary to identify which of the detected objects corresponds with the preceding vehicle, with which a cooperative trajectory is to be constructed. Each of the detected objects consists of measurements of a lateral position and velocity, and a longitudinal position, velocity and acceleration estimate. By use of the elevation radar, the height and vertical position are also included. Additionally, a class (unknown, four wheel vehicle, two wheel vehicle, pedestrian, construction element) is included. Finally, the radar also provides a measure of probability of the detected class and a probability that the object indeed exists.

Consider the schematic representation of the MIO selection in Figure 6.2, which represents the selection of the MIO from all detected obstacles. Denote the position, yaw rate and curvature of the host vehicle as  $x_{h,x}(t)$ ,  $x_{h,y}(t)$ ,  $\dot{\theta}_h(t)$  and  $\kappa_h(t_{i,k})$  respectively. Denote the lateral and longitudinal component of the distance towards obstacle  $j$  expressed in the host vehicle reference frame as  $d_{j,x}(t)$  and  $d_{j,y}(t)$  respectively. Then the following can be written to relate the position of the host vehicle and the obstacle

$$\begin{bmatrix} q_{j,x}(t) \\ q_{j,y}(t) \end{bmatrix} = \begin{bmatrix} q_{h,x}(t) \\ q_{h,y}(t) \end{bmatrix} + \begin{bmatrix} \cos \theta_h(t) & -\sin \theta_h(t) \\ \sin \theta_h(t) & \cos \theta_h(t) \end{bmatrix} \begin{bmatrix} d_{j,x}(t) \\ d_{j,y}(t) \end{bmatrix} \quad (6.5)$$

$$\tan \theta_{\Delta_j}(t) = \frac{d_{j,y}(t)}{d_{j,x}(t)}. \quad (6.6)$$

In addition, the preceding vehicle communicates an estimate of its position (e.g., longitude and latitude) denoted by  $\hat{\mathbf{x}}_j(t) = [\hat{q}_{j,x}(t), \hat{q}_{j,y}(t)]$ . These state estimates are obtained by means of the state estimator described in [65], which is implemented on both experimental vehicles.

The procedure is now as follows. First, the driving lane is identified. Typically, this can be done by means of computer vision, or by means of a HAD map in combination with the lateral position in the Frenet frame,  $\ell$  (see also Figure 2.2). However, both of these methods are unavailable during these experiments. Therefore, instead the driving lane shall be identified by means of the curvature of the host vehicle  $\kappa_h(t)$ . Next, the driving lane is used to search for the most important object among the obstacles that have sufficient probability to exist (i.e.,  $\text{prob}(\text{obj}) > \epsilon_p$ ). This MIO must satisfy certain criteria. The measured distance to the object should correspond with the difference between the estimate of the host position and the estimate of the communicated position of the preceding vehicle. Hence, it is required that

$$\left\| \tilde{\mathbf{x}}_j(t) - \hat{\mathbf{x}}_h(t) \right\|_2 - \|\mathbf{d}_j(t)\|_2 \leq \epsilon_r. \quad (6.7)$$

Additionally, the ‘speed over ground’ of the obstacle can be estimated by measurement of the relative velocity components  $\dot{d}_{j,x}(t)$  and  $\dot{d}_{j,y}(t)$ . This estimate should be sufficiently close to the communicated velocity of the preceding vehicle  $\tilde{v}_j(t)$  as

$$|\hat{v}(t) - \tilde{v}_j(t)| \leq \epsilon_v, \quad (6.8)$$

$$\hat{v}_j(t) = \left\| \begin{bmatrix} v_h(t) \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{d}_{j,x}(t) \\ \dot{d}_{j,y}(t) \end{bmatrix} + \dot{\theta}_h(t) \begin{bmatrix} -d_{j,y}(t) \\ d_{j,x}(t) \end{bmatrix} \right\|_2. \quad (6.9)$$

The objects in the driving lane are first checked, starting with the closest until (6.7), (6.8) are satisfied. If no detection is found that satisfies both criteria, the search is extended to outside the driving lane. During the experiments, the parameters tuned manually which resulted in  $\epsilon_p = 0.5$ ,  $\epsilon_r = 2$  m and  $\epsilon_v = 2 \text{ m s}^{-1}$ .

### 6.2.3 Drivetrain Input

It should be noted that the model (2.6) is the result of feedback linearisation. As a result, the torque input to the vehicle is computed by means of the required tractive force  $F_{x,i,k}(t)$ , which is computed as

$$F_{x,i,k}(t) = m_{\text{eff},i} \ddot{s}_{i,k}(t) + F_r(\hat{s}_i(t)), \quad (6.10)$$

where  $F_r(\hat{s}_i(t))$  denotes the resistive force as function of the current velocity  $\hat{s}_i(t)$ , and is evaluated at the estimated velocity  $\hat{s}_i(t)$ . This resistive force is identified for the Twizy’s and described in [141]. The throttle position (TPS) required to achieve this tractive force at the current motor RPM is then computed by means of inverted torque maps. For this throttle position the lower range 0% - 20% is assigned to regenerative braking, and the upper part 20% - 100% is assigned to driving torque.

In the implementation, the low-level input is computed directly from the planned trajectory  $s_{i,k}(t)$ . No low-level feedback control is used. The reason for this is the validity of the vehicle model (2.6), in combination with a small input-delay. The addition of a feedback controller would result in a higher order model for the drivetrain as seen by the trajectory planner. Moreover, it was noticed that the vehicle was able to follow the planned trajectory without feedback controller sufficiently accurate. A similar approach is used in [3], which utilizes a road-load estimator to compensate for different vehicle masses and road-slopes. In this work, the roadload compensation  $F_r(\hat{s}_i(t))$  is set to zero when the velocity  $|\hat{s}_i(t)| < 1$  and the desired acceleration  $|\hat{\ddot{s}}_{i,k}(t)| < 0.01$ . These bounds are tuned manually, to prevent the vehicle from applying a road load compensation torque at standstill, which could result in the vehicle crawling forward.

### 6.2.4 Initial Condition of Planning

The planning framework in Chapter 3 and Chapter 4, assume perfect tracking. New trajectories  $\mathbf{S}_{i,k}(t)$ ,  $t \in [t_{i,k}, t_{i,k} + T_i]$  are initialized in  $\mathbf{S}_i(t_{i,k})$ , which, as a result of the perfect tracking assumption, coincide with  $\mathbf{S}_{i,k-1}(t_{i,k})$ . However, in practice the perfect tracking assumption is mostly not valid. Therefore, a notable difference in the experimental validation with respect to the developed framework in Chapter 3 and Chapter 4 is that trajectories are initialised as:

$$\underbrace{\begin{bmatrix} s_{i,k}(t_{i,k}) \\ \dot{s}_{i,k}(t_{i,k}) \\ \ddot{s}_{i,k}(t_{i,k}) \end{bmatrix}}_{\mathbf{S}_{i,k}(t_{i,k})} = \begin{bmatrix} \hat{s}_i(t_{i,k}) \\ \hat{\dot{s}}_i(t_{i,k}) \\ \ddot{s}_{i,k-1}(t_{i,k}) \end{bmatrix}, \quad \forall k. \quad (6.11)$$

Note that the position and velocity are thus initialised at state estimates for curvilinear position  $\hat{s}_i(t)$  and curvilinear velocity  $\hat{\dot{s}}_i(t)$ , whereas the curvilinear acceleration is initialised in such a way that the planned acceleration is continuous. This prevents uncomfortable and rapidly changing inputs to the system, which would otherwise be the result of sensor noise of the acceleration sensor. Additionally, it prevents issues in which the vehicle is at an incline, which would also otherwise directly influence the planned trajectories.

## 6.3 Lead Vehicle Experiments

To validate the framework for cooperative trajectory planning, it is required to have a preceding vehicle that also constructs and communicates B-spline trajectories. Therefore, first the framework for autonomous driving of Chapter 4, that attempts to track a given reference velocity, is implemented on the experimental set-up. Note that as discussed, the experiments only include trajectory planning in the longitudinal direction (i.e.,  $s_{i,k}(t)$ ). During implementation, the reference velocity  $\nu_i(t)$  in (4.3), is implemented such that it takes different values at different plan steps  $k$ , which is denoted as  $\nu_{i,k}(t)$ ,  $t \in [t_{i,k}, t_{i,k} + T_i]$ . In the experiment, this is implemented by a user defined value  $v_{c,k}$ , that is used to assign values to  $\nu_i(t)$  in (4.3) as

$$\nu_{i,k}(t) = v_{c,k}, \quad t \in [t_{i,k}, t_{i,k} + T_i], \quad (6.12)$$

such that the reference velocity takes a constant value over the entire planning horizon during plan step  $k$ . Note that this results in temporal inconsistent behavior as  $\nu_{i,k}(t)$  itself is temporally inconsistent, due to instantaneous changes in  $v_{c,k}$  by the user. Typically, during normal operation, the reference velocity  $\nu_{i,k}(t)$  is temporally consistent, as it is intended to relate to a maximum legal velocity associated with the current road. However, for the implementation in the experiment, using the proposed method allows for easily changing velocity.

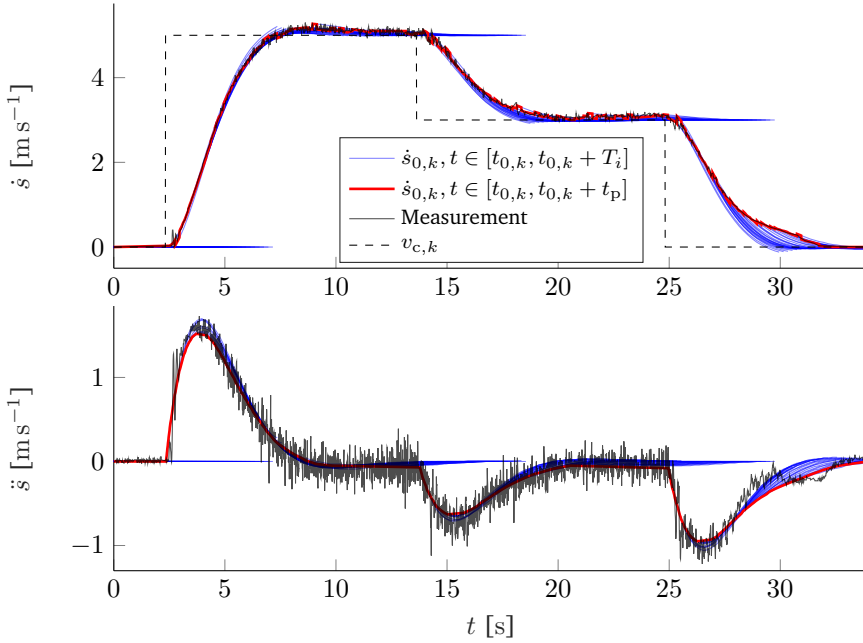


Figure 6.3: Lead vehicle planning to a velocity of  $\dot{s} = 5 \text{ m s}^{-1}$  and  $\dot{s} = 3 \text{ m s}^{-1}$ , and back to standstill

During the experiment, weight parameters have been chosen as  $\mathcal{Q}_s = [1, 0, 1]$ , similarly to the parameters chosen in Section 4.2. Note that the values are scaled with respect to the values chosen in Section 4.2, because no lateral cost needs to be considered during the experiment. Figure 6.3 and Figure 6.4 show two changes in reference velocity. These figures show the planned trajectories in blue. The red trace indicates the part of the planned trajectories that is executed before a new trajectory is planned (i.e.,  $t \in [t_{i,k}, t_{i,k} + T_i]$ ).

First consider Figure 6.3, which shows three gradual velocity transitions, with relatively small accelerations. It can be seen that the method of consecutively planning trajectories towards a target velocity  $v_{c,k}$  indeed achieves following the desired velocity. Moreover, it can be seen that the measurement (depicted in black) closely follows the pieces of the trajectories that are executed by the vehicle, which is depicted in red. A small exception can be seen when the vehicle is nearly at standstill. This discrepancy is expected to be related to a slightly inaccurate torque-map for the braking part (i.e., regenerative braking) at low velocities.

Next consider Figure 6.4, which shows a more aggressive velocity change. It can be seen that again, the measurement follows the executed part of the planned trajectories fairly well. An exception can be seen in the braking part of



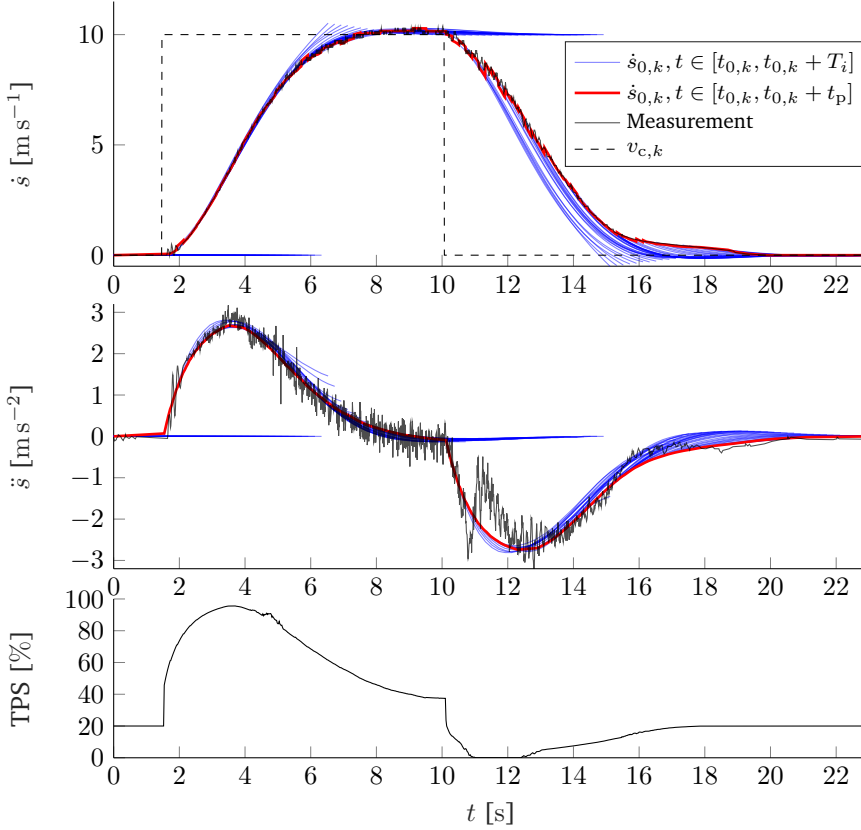


Figure 6.4: Lead vehicle planning to a velocity of  $\dot{s} = 10 \text{ m s}^{-1}$ , and back to standstill.

the measurement around approximately  $t = 11 \text{ s}$  in the figure. This occurs when the throttle position that is the result of the inverse torque map becomes 0%, which results in the highest possible braking torque of the drivetrain. As a result, the driveline has been saturated, and can no longer accurately track the selected trajectory. Additionally, note that the input to the driveline is fairly smooth. This is the result of initializing each plan step with in the evaluated acceleration of the previous plan step.

## 6.4 Cooperative Vehicle Following Experiments

With the functionality of the lead vehicle (e.g., the planning of autonomous longitudinal trajectories) as described in Chapter 4 validated, the next step is to validate the functionality of the cooperative trajectory planning of Chapter 3.

First, the planning delay  $\zeta$  is evaluated. The effect of the planning delay  $\zeta$  is discussed in both Chapter 3 and Chapter 5 and was shown to have a significant effect on the behavior of the cooperative trajectory planner. Next, the gap closing of Section 3.4 is considered, as this is essentially used for the ad-hoc forming of platoons. Additionally, since the lead vehicle can remain stationary for gap closing, the initial tests can be executed safely, as velocities are low and the planner behavior is predictable. After gap closing with the stationary lead vehicle, gap closing and gap opening are considered with the lead vehicle driving at a fixed velocity. Finally, cooperative vehicle following is considered with various time gap parameters, where the lead vehicle transitions from a given velocity to a different velocity. During all experiments the B-splines that are used to construct the planned trajectories are defined by means of  $p = 5$ ,  $n = 6$ ,  $T_i = 5$  s,  $t_p = 0.2$  s and weights  $\mathbf{Q}_s = [Q_s, Q_{\dot{s}}, Q_{\ddot{s}}] = [1, 0, 1]$ , such that all parameters are chosen identical to those of Section 6.3.

### 6.4.1 Planning Delay

During cooperative trajectory planning, the planning delay  $\zeta$  between the plan time  $t_{i-1,k}$  and time at which this information is used in the planning of the host vehicle  $t_{i,k}$  is an important factor in the performance of the cooperative trajectory planner. The analysis in Chapter 5 assumed synchronous planning, where  $t_{i-1,k} = t_{i,k}$ . In the experiments this is no longer the case, instead  $t_{i,k}$  is defined as the time at which vehicle  $i$  uses the information of vehicle  $i - 1$  generated at  $t_{i-1,k}$ . As a result, the planning delay  $\zeta$  originates from the combination of a communication delay,  $\zeta_c$ , and the asynchronicity of the planning of both vehicles. To recognize the effect of asynchronicity, consider the time at which the preceding vehicle started its most recent planning. The time at which the host vehicle can then make use of this information, is related to the point in time where it starts its new planning. This can be recognized in Figure 6.5. The accolades under the horizontal axes indicate the time range in which  $t_{i,k}$  can be found, related to the time  $t_{i-1,k}$ .

In the figure, the time at which the preceding vehicle updates are illustrated, as well as the earliest moment time at which this information is known to the host vehicle (i.e.,  $t_{i-1,k} + \zeta_c$ ). As already mentioned, the time  $t_{i-1,k} \neq t_{i,k}$ , and  $t_{i,k}$  is defined as the time at which vehicle  $i$  utilizes the information that vehicle  $i - 1$  has planned at time  $t_{i-1,k}$ . Therefore  $t_{i,k}$  is related to  $t_{i-1,k}$  as

$$t_{i,k} \in [t_{i-1,k} + \zeta_c, t_{i-1,k+1} + \zeta_c] = [t_{i-1,k} + \zeta_c, t_{i-1,k} + t_p + \zeta_c]. \quad (6.13)$$

In practice, the relative time between  $t_{i-1,k}$  and  $t_{i,k}$  is related to the time when the preceding vehicle and the host vehicle are turned on. As a result,  $\zeta$  can be thought of as uniformly distributed between  $\zeta_c$  and  $t_p + \zeta_c$ , denoted as

$$\zeta \in \mathcal{U}(\zeta_c, \zeta_c + t_p), \quad (6.14)$$

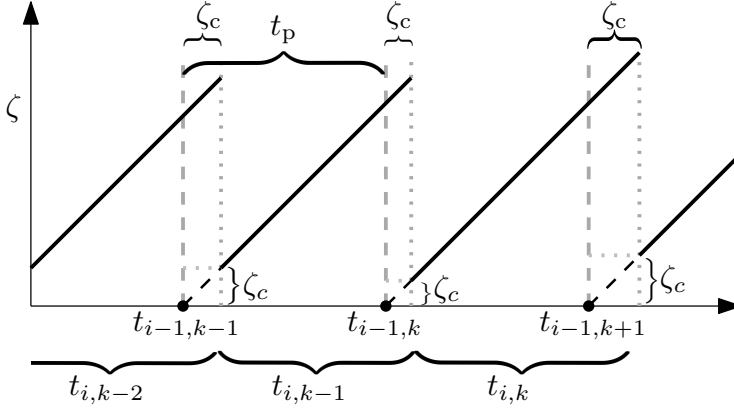


Figure 6.5: Schematic representation of planning delay. Time instances where preceding vehicle updates  $t_{i-1,k}$  (—), earliest moment at which this information is known to the host vehicle  $t_{i-1,k} + \zeta_c$  (····)

where  $\mathcal{U}(\min, \max)$  denotes a uniform probability distribution between  $\min$  and  $\max$ . Note that in practice, the communication delay  $\zeta_c$  itself is also uniformly distributed. However, the span of this distribution is typically significantly smaller (e.g.,  $\zeta_c \in [10, 30]$  ms), than the planning update time  $t_p$ . Therefore, the distribution of  $\zeta$  should be approximated well by (6.14).

In the experiments, both the communication delay  $\zeta_c$  and planning delay  $\zeta$  are determined by comparing the current time  $t_{i,k}$  for both vehicles, by means of the common clock. This common clock is available in the form of the GPS clock. The implementation of this clock on the i-CAVE Twizy's, as well as the communication delay measurement is described in detail in [65]. From each of the recorded measurements in this chapter, the planning delay is evaluated and recorded. A summary of these planning delays during experiments can be found in Figure 6.6.

The bottom figure illustrates the recorded planning delay  $\zeta$ , during all 40 executed experiments, each experiment indicated with a different color. During all these experiments, both vehicles update their planning with an update time  $t_p = 0.2$  s. The top left figure, illustrates a histogram of the planning delay  $\zeta$  during each of the plan steps of each of the experiments. From the figure at the bottom, it can be seen that the planning delay is relatively constant during each experiment. Some small jitter can be seen with a magnitude of 0.01 s. An seeming exception can be seen in one of the red lines around sample  $k = 450$ , where the delay is decreased significantly by approximately 0.2 s. At this point, the delay and jitter are such, that the host vehicle has become aware of a more recent planning of the lead vehicle. It should be noted that the histogram of  $\zeta$  in the top figure gives a somewhat distorted image of the distribution of  $\zeta$ . This is the result of  $\zeta$  not being uniformly distributed during each plan step, but rather

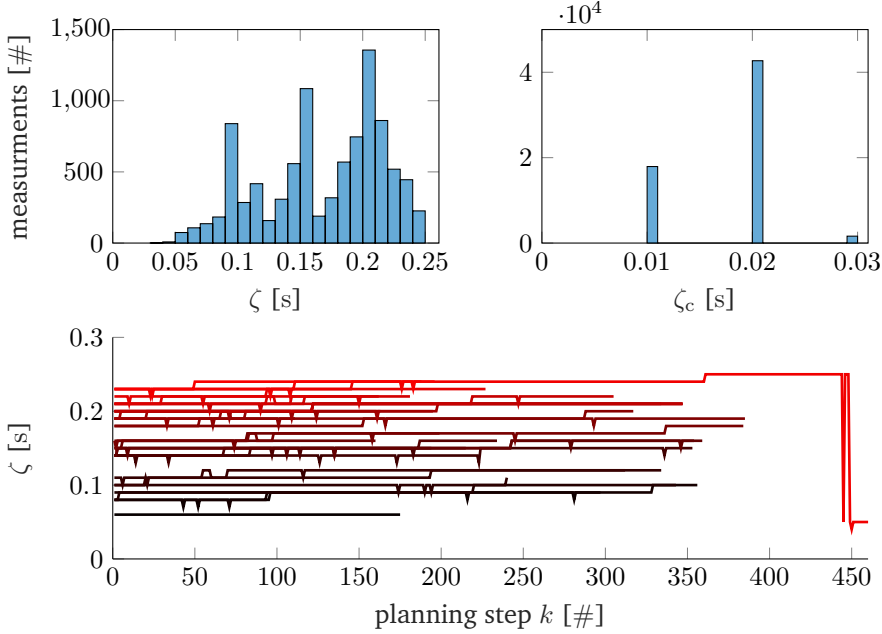


Figure 6.6: Planning delay  $\zeta = t_{i-1,k} - t_{i,k}$ , during all 40 experiments. Both vehicles have a planning update of  $t_p = 0.2$  s

during start-up only. Therefore, longer measurements have more impact on the histogram of  $\zeta$ . By recording more individual experiments (where the system turns on again and are all approximately equal in length), the distribution should start to resemble the distribution of (6.14) more, which is not the case with the data that is available. For the communication delay,  $\zeta_c$ , it should be noted that although the planning of vehicle  $i$  is updated each  $t_p$  seconds, this planning is communicated with vehicle  $i - 1$  more frequently (e.g., at a frequency of 25 Hz), which is mainly used for MIO selection. Consequently, the number of samples of communication delay is much larger. It can be seen that it only takes values of 10, 20 and 30 ms.

### 6.4.2 Gap Closing

The gap closing strategy of Section 3.4, where a reference standstill distance is chosen as a function of time  $c_{r,i,k}(t)$ , is found to be most suitable for implementation, as it can be used at any forward velocity (including standstill). It uses user defined parameters  $\underline{c}_i$  and  $\underline{h}_i$ , to parametrize the spacing policy. To validate the gap closing functionality more easily, these parameters can be adjusted in real-time. This can be used to instantaneously change the spacing

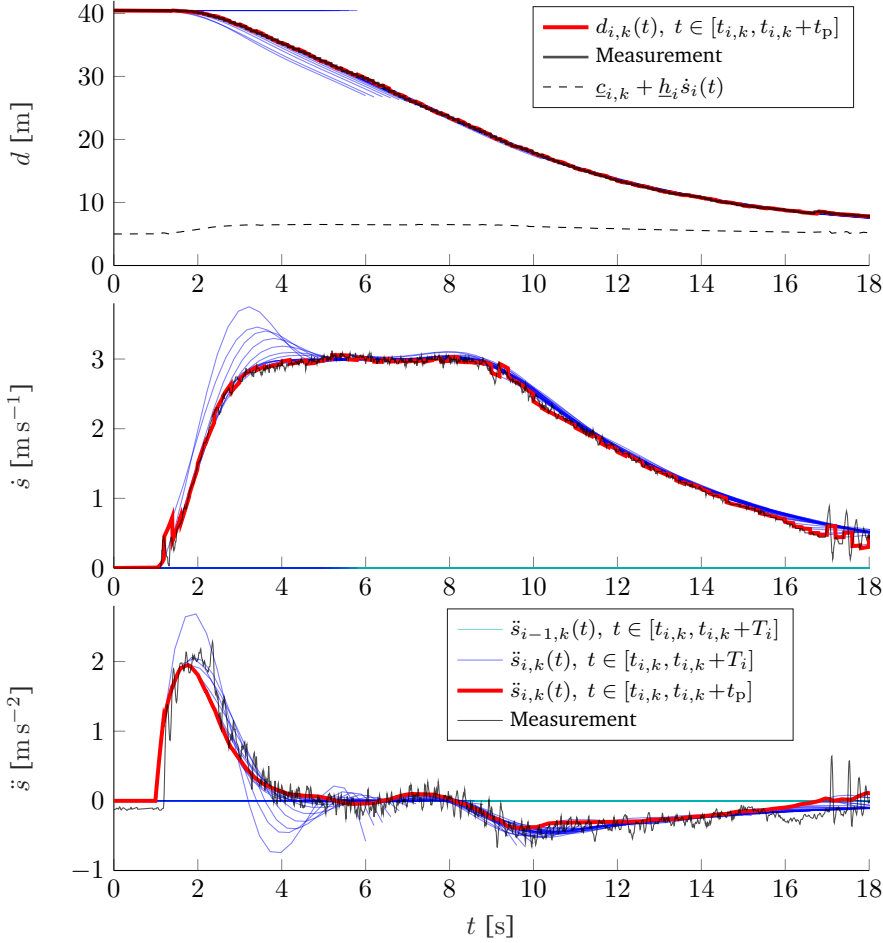


Figure 6.7: Gap closing behind a stationary vehicle. Closing velocity set to  $\underline{v}_{cl} = 3 \text{ m s}^{-1}$ .

policy while both vehicles are in motion, and put the system in a state where the error  $e_i(t) \neq 0$ . This allows to see how the system behaves in these scenarios. To achieve this, parameters  $\underline{c}_{i,k}$  is introduced, while the time gap is kept constant at  $\underline{h}_i$ . During all the experiments involving gap closing, the time gap is set to a constant value  $\underline{h}_i = 0.5 \text{ s}$ ,  $\forall k$ .

### Approaching a Stationary Vehicle

As an initial test, a stationary lead vehicle is considered. This lead vehicle does communicate its planned trajectory by means of the coefficients  $\mathcal{P}_{i-1,k} = s_{i-1}(0)\mathbf{1}_{n+1 \times 1}$ ,  $\forall k$ , such that it remains stationary. Gap closing

parameters are chosen as  $\phi_i = -0.1$ , such that the following vehicle will drive approximately 11% faster than the preceding vehicle to close the gap according to (3.43), and  $v_{cl} = 3 \text{ m s}^{-1}$ , the velocity with which a stationary vehicle is approached. The approach of a standstill vehicle is also demonstrated in Section 3.4, for which the results are shown in Figure 3.19a. The experimental results are shown in Figure 6.7. In the figure, the host vehicle first increases its velocity, until the velocity of  $v_{cl} = 3 \text{ m s}^{-1}$  is achieved. Then this velocity is maintained, until the vehicle is sufficiently close and starts to gradually decrease velocity. Note that some oscillations occur at the very end of the measurement. These are due to the road load compensator switching off at low velocities and desired accelerations.

### Lead Vehicle with Fixed Velocity

In the following experiments, the lead vehicle is no longer stationary. Both vehicles start of stationary, before the lead vehicles accelerates towards a fixed velocity of  $\dot{s}_i(t) = 10 \text{ m s}^{-1}$ . When both vehicles are in steady state at this new velocity, a change is made in the parameter  $\underline{c}_{i,k}$ , to introduce an error with respect to the desired spacing policy. This change is applied in both directions, decreasing  $\underline{c}_{i,k}$  results in a positive error, increasing  $\underline{c}_{i,k}$  results in a negative error.

First, a positive error is introduced, where the host vehicle needs to close a gap while moving. This is illustrated in Figure 6.8. At approximately  $t \approx 8.5 \text{ s}$  the value of  $\underline{c}_{i,k}$  is changed from 15 m to 5 m, resulting in a spacing error of  $e_i(t) = 10 \text{ m}$ . To overcome this error, the planner immediately initiates gap closing, by planning towards a new velocity, which scales with the average velocity of the preceding vehicles according to (3.49). In this example, the lead vehicle already starts to decelerate at approximately  $t \approx 12.4$  during the gap closing, such that only the part highlighted in gray in the figure is purely gap closing. Nevertheless, the presented approach is capable of adjusting to this change in velocity and planning of the lead vehicle.

A second example is illustrated in Figure 6.9. In this example, a negative error is introduced, where the host vehicle needs to increase the inter-vehicle distance. Initially, the standstill distance  $\underline{c}_{i,k}$  is 15 m. Then at  $t = 18 \text{ s}$  and  $t = 33 \text{ s}$  the standstill distance is changed to respectively  $\underline{c}_{i,k} = 10 \text{ m}$  and  $\underline{c}_{i,k} = 5 \text{ m}$ , such that a negative error of  $e_i = -5 \text{ m}$  occurs at both changes. The planner takes action by planning a new trajectory that increases the inter-vehicle distance to satisfy the new spacing policy. Notice that despite the spacing error being only half of the positive error in the previous scenario, the control input is similar in magnitude. This illustrates how the gap closing strategy ensures comfortable (e.g., low accelerations) gap closing, but in safety critical scenarios (where the host vehicle is too close to the vehicle ahead) still uses larger control inputs to correct the error.

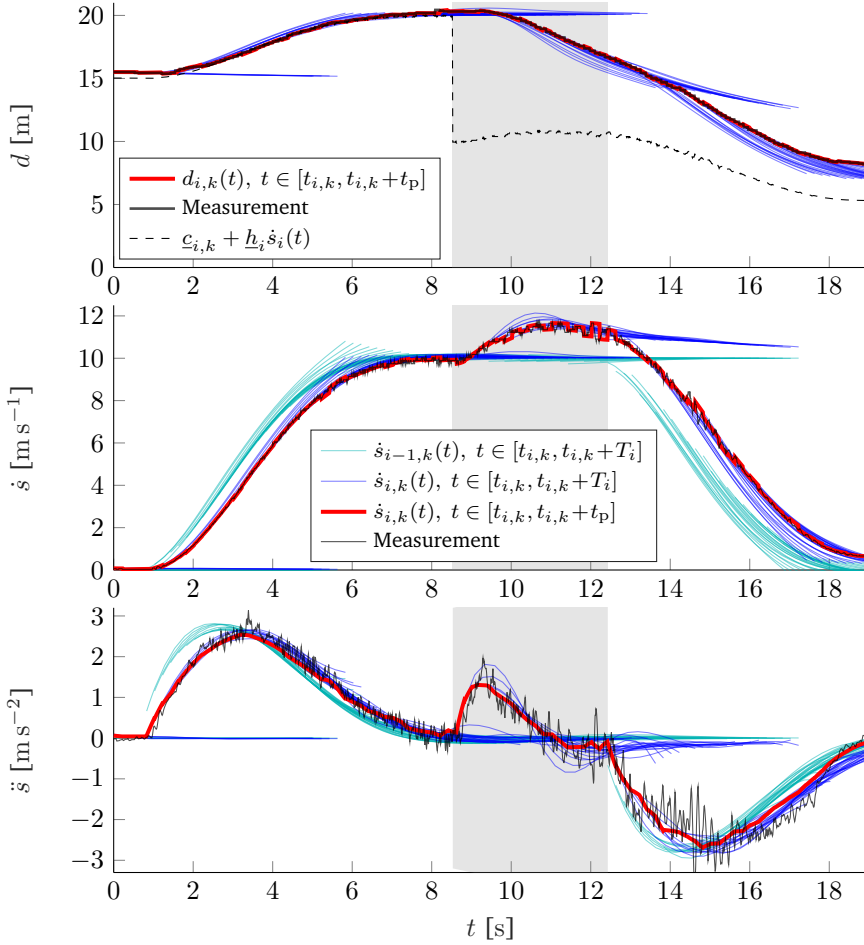


Figure 6.8: Gap closing behind a moving vehicle. Closing velocity set to  $\underline{v}_{cl} = 3 \text{ m s}^{-1}$ .

### 6.4.3 Vehicle Following

Aside from following a vehicle at a constant velocity, the planning algorithms should be capable of changing velocities successfully, for example in case of a changing speed limit. The experiments in this section are used to illustrate the functionality of the cooperative trajectory planner for various time gaps. The standstill distance is kept constant at  $\underline{c}_{i,k} = 5 \text{ m}$ . Various time gaps are used, but in each experiment a single time gap is used, such that  $\underline{h}_i$  only varies between experiments. During each experiment, the lead vehicle accelerates to a velocity of  $v_{c,k} = 10 \text{ m s}^{-1}$ , then maintains that velocity for a few seconds before decelerating to a lower velocity of  $v_{c,k} = 7 \text{ m s}^{-1}$ . For each experiment the mean planning

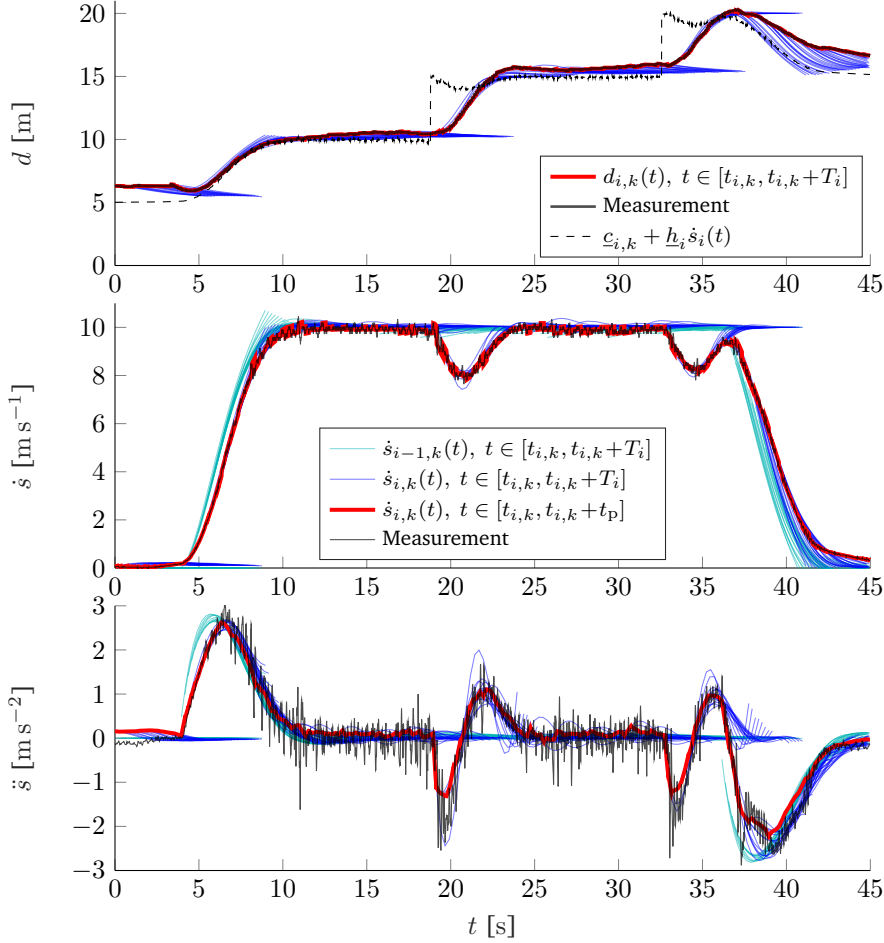


Figure 6.9: Gap opening transitioning from  $c_{i,k} = 5$  m to  $c_{i,k} = 10$  m to  $c_{i,k} = 15$  m

delay is denoted by  $\bar{\zeta}$ .

An overview of the experiments in this section are shown in Table 6.2. The result of these experiments are shown in Figure 6.10, Figure 6.11 and Figure 6.12 for decreasing time gaps  $h_i$ . Considering the combination of average planning delay  $\bar{\zeta}$  and time gap  $h_i$  and the results from Chapter 5, it is expected that the experiments illustrated in Figure 6.10 and Figure 6.11 would demonstrate string stable behavior. In the figures, some attenuation can indeed be observed by considering the peak planned acceleration, during each velocity transition. Additionally, the numerical approximation of the  $\mathcal{L}_2$ -norm of the planned acceleration also indicates string stable behavior in the  $\mathcal{L}_2$  sense. In



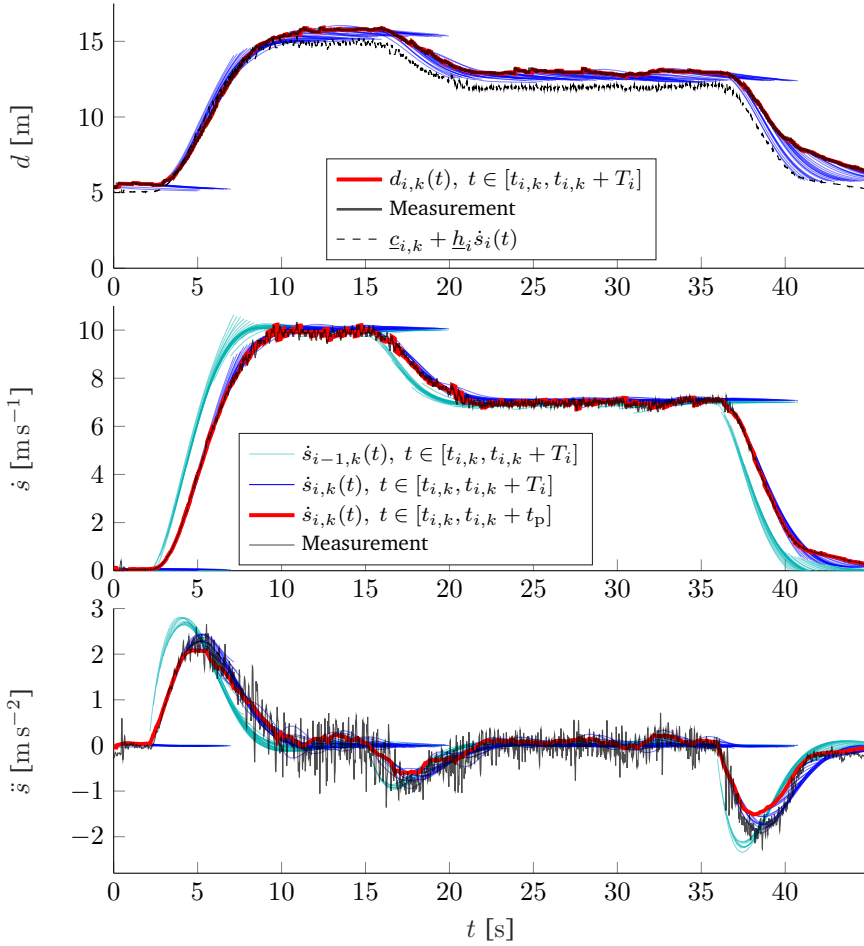
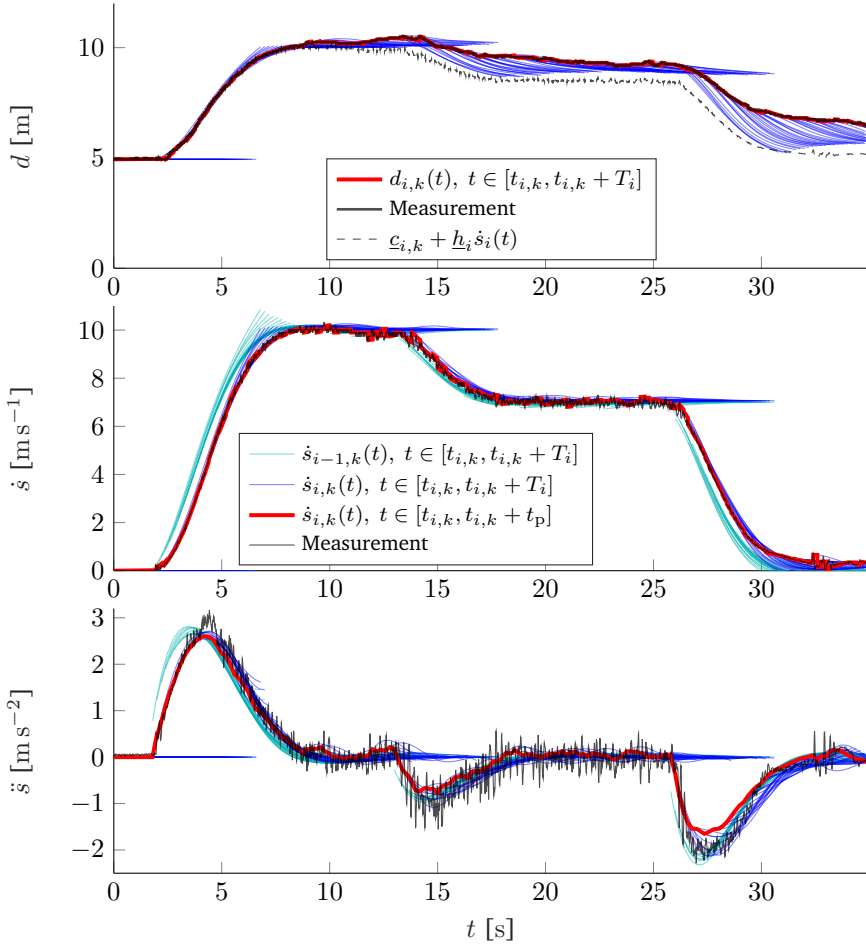


Figure 6.10: Vehicle following  $h_i = 1.0$  s,  $\bar{\zeta} = 0.0899$  s

Table 6.2: Overview of vehicle following experiments.

$h_i$ [s]	$\bar{\zeta}$ [s]	$\ \ddot{s}_{i,k}(t)\ _{\mathcal{L}_2} / \ \ddot{s}_{i-1,k}\ _{\mathcal{L}_2}$	Figure
1.0	0.0899	0.8152	6.10
0.5	0.2406	0.8946	6.11
0.4	0.1529	0.9062	×
0.3	0.0940	0.9096	×
0.2	0.1702	0.9344	×
0.1	0.1155	0.9287	6.12

Figure 6.11: Vehicle following  $h_i = 0.5$  s,  $\bar{\zeta} = 0.2406$  s

contrast, the combination of time gap and planning delay in the experiment illustrated in Figure 6.12, should result in string unstable behavior, as the delay is larger than the minimum time gap in Figure 5.7b. However, this instability can not be seen from the Figure 6.12, nor from the ratio of the numerical approximation of the  $\mathcal{L}_2$ -norm of the planned acceleration. It is clear however, that the attenuation of the  $\mathcal{L}_2$ -norm of the planned acceleration signal does decrease for decreasing time gap.

Some observations should be made regarding the performance of the implemented planner. First, it should be noted that the radar initially does not accurately track the position of the lead vehicle in Figure 6.12. Although in the majority of cases the radar did accurately track the lead vehicle, during some

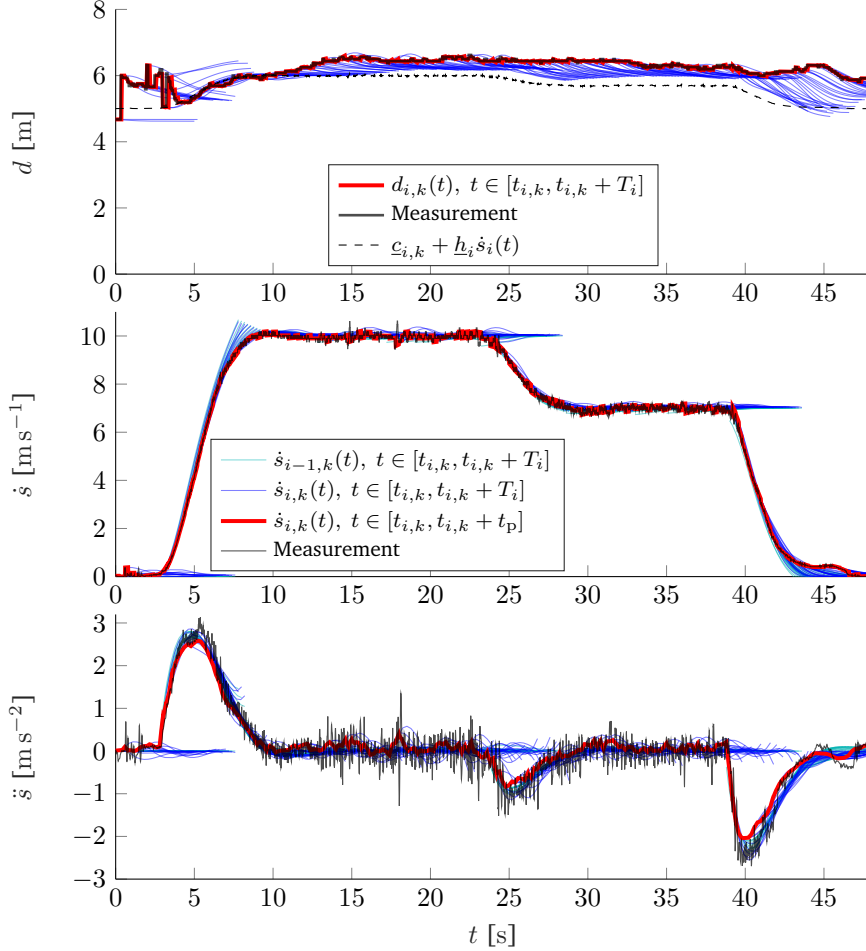


Figure 6.12: Vehicle following  $h_i = 0.1$  s,  $\bar{\zeta} = 0.1155$  s

experiments this type of jitter on the radar measurement can be seen. This can be improved by integrating additional sensors for the purpose of environmental sensing. Secondly, in most experiments, the measured spacing between the vehicles did not exactly match the desired spacing policy and a small positive error can be seen. This might be due to the due to the implementation of the low level controller, which does not include any feedback as described in Section 6.2.3. As a result, the correct drive torque might not be applied to the vehicle. A different explanation could be related to the error always being positive. Hence, it is possible that it is related to the gap closing strategy. When the vehicle is approximately driving with a spacing error equal to zero, a

disturbance that increases the error (host vehicle further away from preceding vehicle), the generated control input is lower than if the error was negative, as is already remarked while discussing the contents of Figure 6.8 and Figure 6.9. This asymmetry might result in the small bias that can be seen in some of the experiments, where an external disturbance cannot be overcome by the ‘weaker’ control input of the gap closing strategy. This also explains why the  $\mathcal{L}_2$ -norm of the planned acceleration of Figure 6.12, seemingly indicates string stable behavior, as the positive spacing error results in lower magnitude accelerations due to gap closing.

## 6.5 Summary and Discussion

In this chapter, experimental results are presented that are used to validate the framework presented in Chapter 3 and Chapter 4. First, an overview of the experimental set-up is provided, as well as some important details on the implementation. Next, experimental results are presented for both autonomous as well as cooperative driving. It is shown that the autonomous trajectory planner of Chapter 4 is indeed capable of tracking a desired velocity. Additionally, the cooperative trajectory planner of Chapter 3 is shown to be able to maintain a desired spacing policy, while following the lead vehicle. This is true for the regular planner, as well as the gap closing strategy that was implemented for improved passenger comfort. Note that the longitudinal part of the cooperative trajectory planning framework is the most significant contribution of this thesis. Therefore, despite only being able to test the longitudinal part of the trajectory planner, these experiments clearly indicate that this framework can be implemented in practice.

In conclusion, although some minor improvements can be made to the implementation itself, it is demonstrated that this framework can be successfully implemented in real-time to achieve both autonomous, as well as cooperative driving.



# Chapter 7

## Time Scaling

While cooperative trajectory planning shows significant benefits over more traditional feedback methods, it does suffer from a small drawback. Specifically, trajectory planning is computationally more expensive, as a result of which the rate at which planned trajectories are updated, is chosen to be smaller than the rate at which information is provided to the system. This limits the control bandwidth of achieving a certain spacing policy. This chapter presents a potential solution to relieve this bandwidth limitation of the trajectory planning method.

### 7.1 Scaling Approach

The cooperative trajectory planning method of Chapter 3 is capable of achieving string-stable behavior as well as comfortable gap closing. However, so far it was assumed that the preceding vehicle indeed executes its communicated trajectory. In practice, the preceding vehicle will most likely slightly deviate from its communicated trajectory, as the preceding vehicle is subject to various disturbances or its V2V communication might be impaired. These deviations can partly be compensated in the next plan cycle of the host vehicle. However, the update rate of the trajectory planning is typically low (e.g., 5 Hz), due to computational requirements of the algorithm. In this section the time scaling method is presented (see also [150] for more details), that is used as a safety feature against V2V impairments and inaccurate communicated or predicted trajectories of other traffic participants. This method adjusts the existing planned trajectory instead of replanning it entirely.

Recall that the planning algorithms of Chapters 3 and 4 plan trajectories in curvilinear coordinate  $s$  for vehicle  $i$  as  $s_{i,k}(t)$ ,  $t \in [t_{i,k}, t_{i,k} + T_i]$  by means of B-spline of degree  $p$  with  $n + 1$  control points. These trajectories are planned at

times  $t_{i,k}$  and span over a horizon  $T_i$ . The planned trajectory including velocity and acceleration is denoted as

$$\mathbf{S}_{i,k}(t) := \begin{bmatrix} s_{i,k}(t) \\ \dot{s}_{i,k}(t) \\ \ddot{s}_{i,k}(t) \end{bmatrix} = \begin{bmatrix} s_{i,k}(t) \\ v_{i,k}(t) \\ a_{i,k}(t) \end{bmatrix}, \quad t \in [t_{i,k}, t_{i,k} + T_i]. \quad (7.1)$$

Instead of replanning the trajectory, a method that adapts the existing trajectory is required. For this reason, the time scaling method is introduced. In time scaling, the planned trajectory  $\mathbf{S}_{i,k}(t)$  can be interpreted as a look-up table that is used by the host vehicle during the time period  $t \in [t_{i,k}, t_{i,k+1}]$ . This look-up table can be read-out at a different pace than real-time, to speed up or slow down the trajectory. For safety reasons, only retarding actions should be allowed for the time-scaling method, and decisions for accelerating are left to the planning algorithm. Additionally, the trajectories described by B-splines are limited in terms of rise time of the acceleration by the ratio  $T_i/n$ , due to the continuity requirements of the B-splines. This limitation applied to both the lead vehicle as well as the following vehicle. This might be a problem in emergency braking conditions. Instead of increasing the amount of control points  $n$ , which leads to higher computation time and higher communication bandwidth, also time scaling can be used to remain safe.

The time scaling method is defined as follows. The *spacing velocity*,  $v_{\text{sp},i}(t)$ , is determined, which is found by rewriting the spacing policy (3.4):

$$v_{\text{sp},i}(t) = \frac{d_i(t) - \underline{c}_i}{\underline{h}_i}, \quad (7.2)$$

$$d_i(t) = s_{i-1}(t) - s_i(t) - L_i, \quad (7.3)$$

where  $d_i(t)$  is the measured inter vehicle distance, and  $\underline{c}_i$  and  $\underline{h}_i$  user defined parameters as presented in Chapter 3. This velocity  $v_{\text{sp},i}(t)$  is then used to define a progression rate of read-out time  $\tau_i(t)$  as

$$\dot{\tau}_i(t) = \begin{cases} \frac{v_{\text{sp},i}(t)}{v_{i,k}(\tau_i(t))} & \text{if } \frac{v_{\text{sp},i}(t)}{v_{i,k}(\tau_i(t))} \leq 1, \text{ subject to } 0 < v_{i,k}(\tau_i(t)), \\ 1 & \text{otherwise,} \end{cases} \quad (7.4)$$

$$\tau_i(t) = \int_{t_{i,k}}^t \dot{\tau}_i(t) dt + \tau_i(t_{i,k}), \quad t \geq t_{i,k}, \quad (7.5)$$

which is set to the current time at the start of each plan step. For cases where the reference velocity is zero, time scaling is not required (i.e.,  $\dot{\tau}_i = 1$ ), as the vehicle does not move anyway. Note that, according to (7.4), this ‘new’ time  $\tau_i(t)$  never progresses faster than real-time, such that  $0 \leq \dot{\tau}_i(t) \leq 1, \forall t$ . This new ‘read out time’  $\tau_i$  is used to obtain the new scaled curvilinear position  $s_\tau(t)$  as

$$s_\tau(t) = s_{i,k}(\tau(t)), \quad (7.6)$$

and corresponding reference trajectory for the level vehicle control:

$$\mathbf{S}_{i,k,\tau}(t) = \begin{bmatrix} s_\tau(t) \\ v_\tau(t) \\ a_\tau(t) \end{bmatrix} = \begin{bmatrix} s_{i,k}(\tau(t)) \\ \dot{s}_{i,k}(\tau(t)) \\ \ddot{s}_{i,k}(\tau(t)) \end{bmatrix}. \quad (7.7)$$

By means of the chain rule, the time derivatives of  $s_{i,k}(\tau_i(t))$  are

$$\mathbf{S}_{i,k,\tau}(t) = \begin{bmatrix} s_{i,k}(\tau(t)) \\ v_{i,k}(\tau(t))\dot{\tau}(t) \\ a_{i,k}(\tau(t))\dot{\tau}(t)^2 + v_{i,k}(\tau(t))\ddot{\tau}(t) \end{bmatrix}, \quad t \in [t_{i,k}, t_{i,k} + T_i]. \quad (7.8)$$

Note that the time derivatives of the planned trajectory  $s_{i,k}(t)$  are ‘scaled’ by the time derivative of time  $\tau(t)$ , hence the name time scaling. Also note that the new desired velocity is indeed upper bounded by  $v_{sp,i}(t)$ , since:

$$v_\tau(t) = v_\tau(\tau_i(t))\dot{\tau}_i(t) = \begin{cases} v_{i,k}(\tau_i(t)) \frac{v_{sp,i}(t)}{v_{i,k}(\tau_i(t))} & \text{if } \frac{v_{sp,i}(t)}{v_{i,k}(\tau_i(t))} \leq 1, \\ v_{i,k}(\tau_i(t)) & \text{else} \end{cases}, \quad (7.9)$$

making it suitable for keeping a safe distance, when the preceding vehicle deviates from its communicated trajectory.

Although the time scaling policy is based on velocities, the scaled acceleration state is required for low level vehicle control as mentioned in Chapter 6. This scaled acceleration can be obtained by numerical differentiation of  $v_\tau(t)$ . However,  $v_\tau(t)$  can be discontinuous at planning updates, since the planned trajectories are initiated in the current measured vehicle state, and velocity measurements naturally contain noise. This makes numerical differentiation undesirable. Instead, the expression for  $a_\tau(\tau(t))$  in (7.8) is used. This requires computing  $\ddot{\tau}(t)$ . It should be noted that since  $\dot{\tau}$  is not continuously differentiable at point where (7.4) switches cases,  $\ddot{\tau}(t)$  is not well-defined everywhere. However, since it is only not well defined at points where timescaling enables or disables, this is not an issue in practice. Instead, a substitute variable  $\tilde{\ddot{\tau}}(t)$  is computed, using (7.4) as

$$\tilde{\ddot{\tau}}(t) = \begin{cases} \frac{v_{i,k}(\tau(t))\dot{v}_{sp,i}(t) - v_{sp,i}(t)a_{i,k}(\tau(t))\dot{\tau}(t)}{v_{i,k}(\tau(t))^2}, & \text{if } \frac{v_{sp,i}(t)}{v_{i,k}(\tau(t))} \leq 1, \\ 0, & \text{otherwise,} \end{cases} \quad (7.10)$$

$$\dot{v}_{sp,i}(t) = \frac{1}{h_i} \dot{d}_i(t), \quad (7.11)$$

where  $\dot{d}_i(t)$  can be measured from the on-board radar of the host vehicle. As remarked before, the time-scaling method should only perform retarding actions (i.e., reducing the velocity of the vehicle). However, noise in the velocity



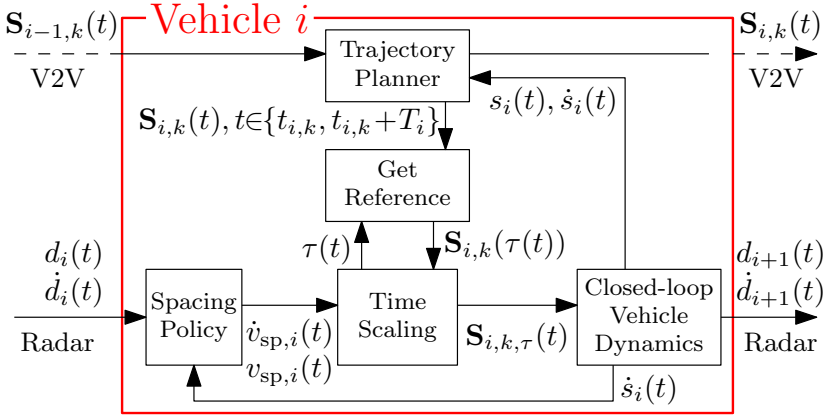


Figure 7.1: Schematic overview of trajectory planning with time scaling system.

measurement of the radar can still result in an increase in the acceleration term in (7.8) due to (7.10), even when  $v_{i,k}(\tau(t)) < v_{sp,i}(t)$ . The resulting scaled trajectories are then evaluated slightly different from (7.8) as

$$\mathbf{S}_{i,k,\tau}(t) = \begin{bmatrix} s_{i,k}(\tau(t)) \\ v_{\tau}(t) \\ a_{\tau}(t) \end{bmatrix}, \quad t \in [t_{i,k}, t_{i,k} + T_i], \quad (7.12a)$$

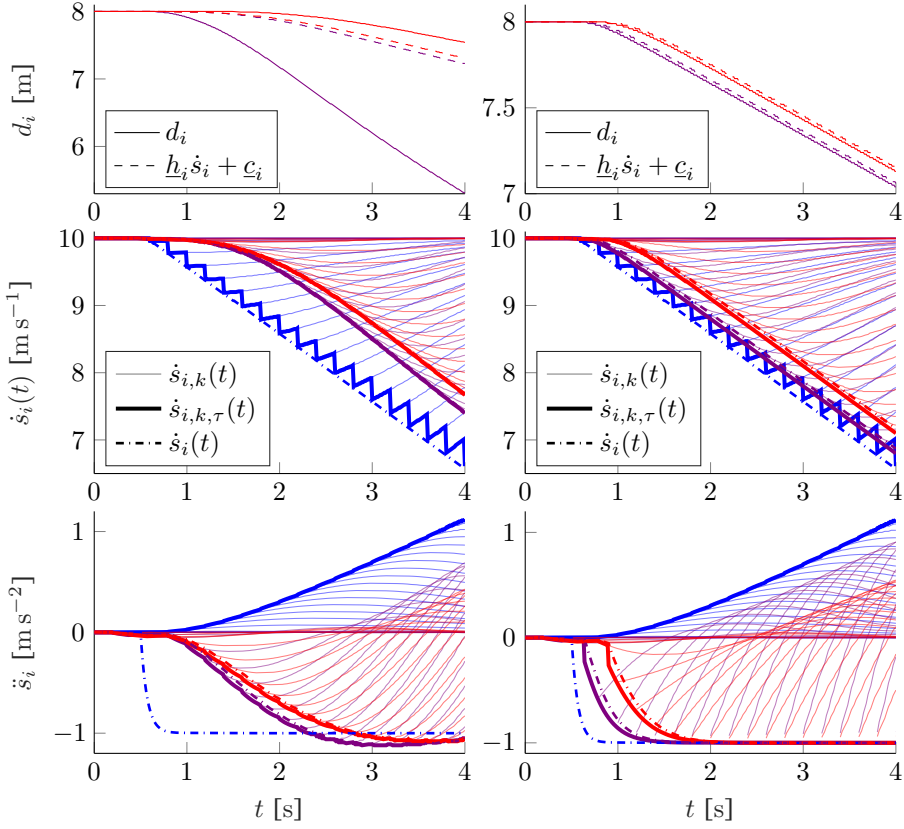
$$v_{\tau}(t) = v_{i,k}(\tau(t))\dot{\tau}(t), \quad (7.12b)$$

$$a_{\tau}(t) = \min [a_{i,k}(\tau(t)), (a_{i,k}(\tau(t))\dot{\tau}(t)^2 + v_{i,k}(\tau(t))\ddot{\tau}(t))]. \quad (7.12c)$$

The trajectory of plan step  $k$  is then initialized in the current scaled acceleration  $a_{\tau}(t_{i,k})$ . This ensures that in the next trajectory the scaled acceleration of the previous plan step is taken into account. This in turn informs the vehicles behind of the braking action. The resulting process is schematically illustrated in Figure 7.1, which shows the inputs to vehicle  $i$ , by both the radar, as well as the V2V communication. Note that additional information is utilized since the relative velocity  $\dot{d}_i(t)$  is now used, while in Chapter 6 only radar distance together with V2V messages were utilized. This additional source of information helps the vehicle to react faster to the changing condition.

## 7.2 Simulation Results

To illustrate in what scenarios time scaling might improve the behavior of the platoon, a simulation study is performed. Since time scaling is useful in particular in scenarios where a vehicle deviates from its reference trajectory, perfect tracking of the reference trajectory is no longer a sensible assumption. To



(a) Trajectory planning without time scaling.

(b) Time scaling enabled, with  $\dot{\tau}$  as in (7.4) and reference trajectories (7.12).

Figure 7.2: Comparison of trajectory planning with and without time scaling, Lead vehicle (—), first follower (—), second follower (—)

simulate vehicle suffering from a disturbance, a vehicle model is created. This vehicle model includes the powertrain and road load model as described in [141]. Additionally, a first order dynamical system like (2.6), is used to capture the first order dynamics of the drivetrain, with drivetrain parameter  $\eta_i = 0.0687$  s as identified in [65].

A scenario is considered where at  $t = 0.5$  s, the lead vehicle applies an input of  $u_i = -1$  m s<sup>-2</sup>, which can be interpreted as a failure of the automated system. While this vehicle starts to decelerate, it is unaware of the impairment and continues to plan to the desired velocity of 10 m s<sup>-1</sup>. This scenario is simulated for a string of three vehicles that make use of the planning method described in

Chapters 3 and 4, with parameters  $n = 6$ ,  $p = 5$ ,  $\underline{h}_i = 0.3$  s,  $\underline{c}_i = 5$  m. Figures 7.2a and 7.2b illustrate a scenario where time scaling is not used and where it is used, respectively. It can be seen that the lead vehicle indeed decelerates, as indicated by the actual acceleration  $\ddot{s}_i$  in the dash-dotted blue line. Note that since the lead vehicle does not have a vehicle in front of it,  $\dot{\tau}_0(t) = 1$ ,  $\forall t$  and  $s_{0,k,\tau}$  follows the planned trajectory  $s_{0,k}$ . However, also note the discrepancy between what this lead vehicle is planning and what it is executing, which is completely different. As a consequence, it also communicates a trajectory in which the vehicle accelerates, rather than the decelerating action that it is performing, which in turn is used by the vehicles driving behind it. In Figure 7.2a, where time scaling is not used, it can be seen that while the replanning of the following vehicles does result in a decrease in velocity, it is not sufficient. This results in the inter vehicle distance becoming significantly smaller than the desired inter vehicle distance as can be seen in the top plot.

In contrast, Figure 7.2b illustrates the same scenario, but with the time scaling mechanism active for both following vehicles. It can be seen that, similarly to Figure 7.2a, both following vehicles still plan their trajectories on faulty information from the preceding vehicle, as all the planned trajectories attempt to increase the acceleration and velocity. However, despite this incorrect planned trajectory, the vehicle much faster with time scaling than without time scaling. It can be seen that while both following vehicles suffer from a tracking error, the tracking error remains small.

### 7.3 Experimental Validation

The approach is also implemented on the experimental set-up. For the spacing policy  $\underline{h}_i = 0.5$  s and  $\underline{c}_i = 10$  m is used. A disturbance is applied to the lead vehicle by means of manually applying a braking torque to the vehicle, using the hydraulic brake system of the lead Twizy. Therefore, it is incapable of accurately tracking the planned and communicated trajectory. Figure 7.3 illustrates an experiment where the brake is applied, such that the lead vehicle is incapable of departing. It can be seen that the follower vehicle also plans to depart for approximately  $t \in [1, 14]$  s. However, as the host vehicle is already at the desired standstill distance, (7.4) will prevent progression of the readout time  $\tau$ . Hence, the host vehicle will not start departing, thus maintaining the initial inter-vehicle distance. Next, at  $t \approx 15$  s, the brake of the lead vehicle is released, and both vehicles depart. Then at  $t \approx 36$  s the hydraulic brake of the lead vehicle is again applied as a disturbance. It can be seen that the host vehicle again successfully maintains a safe distance, despite the lead vehicle not tracking its planned and communicated trajectory. However, one drawback is that the slightly noisy measurement of the relative velocity  $\dot{d}_i(t)$ , now directly affects the control input (i.e., the commanded acceleration) of the vehicle.

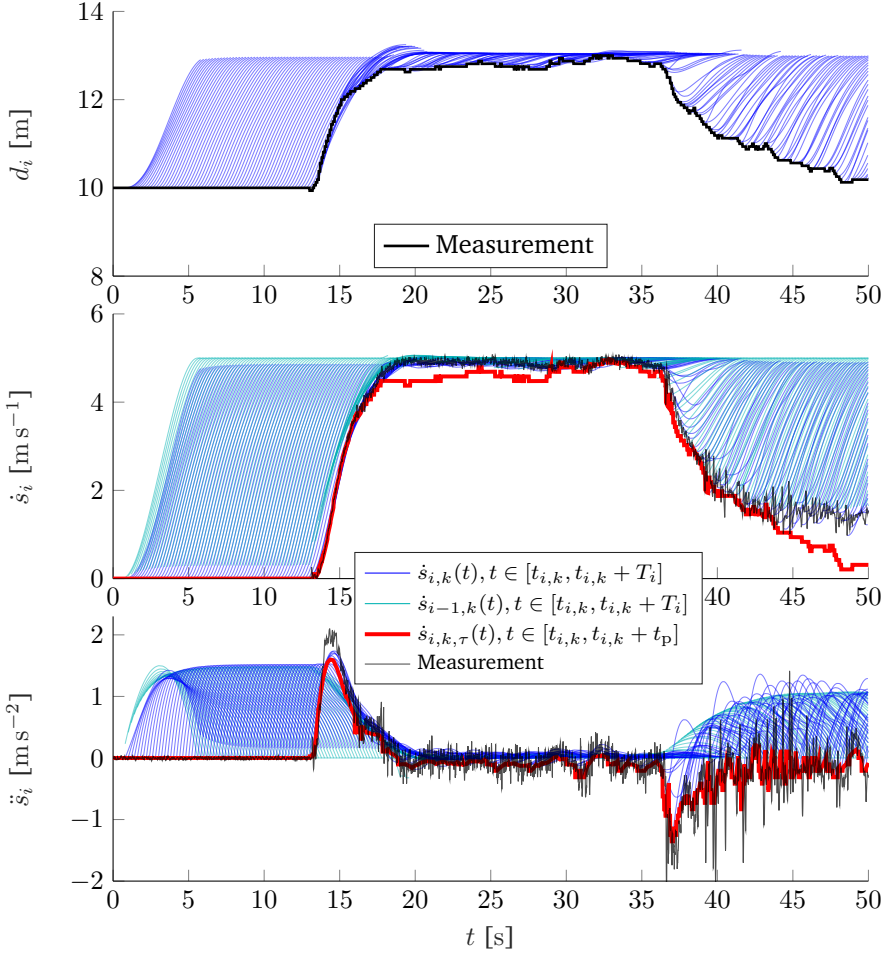


Figure 7.3: Experimental results of time scaling, with  $\dot{\tau}$  according to (7.4), and reference trajectories as in (7.12).

## 7.4 Summary and Discussion

The time scaling method is presented as an extension to the trajectory planning framework for automated cooperative vehicles. This method utilizes radar information in between plan steps, such that the automated vehicle is capable of adjusting its control input without replanning. Therefore, it helps to overcome the drawback of trajectory planning related to the limited up-date frequency. It is demonstrated that in case a discrepancy appears between the communicated and the executed trajectory of the preceding vehicle, the time scaling mechanism

helps to reduce the spacing error. Hence, it provides a method of graceful degradation of the cooperative trajectory planning framework. The discrepancy between the communicated and executed trajectory can be either due to a faulty or damaged lead vehicle, or due to a V2V impairment.

The time-scaling method utilises the same information as traditional Adaptive Cruise Control (ACC), being the inter vehicle distance and the inter vehicle velocity measured by the radar. Without accurate communicated planned trajectories of the preceding vehicle, the minimum time gap with which a vehicle can be followed without losing string stability increases. Therefore, the platoon of vehicles can demonstrate string unstable behavior due to the time scaling mechanism. However, it should be noted that if all vehicles track their communicated trajectories accurately, the control system does not rely on the time scaling mechanism to maintain a safe distance. Instead, it is implemented to remain safe in cases where other parts of the automated driving system are impaired. In cases where the follower vehicle needs the mechanism to maintain distance for a sustained amount of time, it might be beneficial to increase the time gap in the spacing policy. Increasing the time gap would account for the lack of accurate information of the preceding vehicle and potentially improves the behavior of the platoon in those scenarios in terms of string-stability.

# Chapter 8

## Conclusions and Recommendations

Vehicle automation is a research topic of great interest, due to the ever increasing demand of mobility. Both safety in mobility as well as throughput on roads can be increased by means of vehicle automation. Automated vehicles can be divided into two subclasses, which are cooperative vehicles and autonomous vehicles. Both of these subclasses have been researched extensively. This thesis aims to unify the research of both subclasses into a single trajectory planning framework. To do so, the following objectives are defined:

- To design a unified trajectory planning framework for automated and cooperative vehicles. A vehicle equipped with the aforementioned framework can operate both autonomously and in a cooperative fashion, using the same control topology, while broadcasting the same information in both modes.
- To ensure that cooperative trajectories are string-stable, such that the macroscopic traffic system benefits from the increased throughput.
- To validate experimentally the developed framework on full-scale vehicles.

The main conclusions from these objectives are provided in Section 8.1. Recommended topics for further research that are identified in this thesis are presented in Section 8.2.

## 8.1 Conclusion

An autonomous vehicle must be capable of handling a wide variety of scenarios, ranging from highway driving to urban scenarios. These vehicles make use of trajectory planners that can plan actions over a certain time horizon. The vehicle can check these trajectories for feasibility and collisions before executing it. Information about the environment of the vehicle is gathered by means of on-board sensors such as LIDAR (Light Detection And Ranging), radar and camera. In contrast, cooperative vehicles make use of vehicle to vehicle (V2V) communication to achieve some collective behavior. In the context of this thesis, the collective behavior refers to cooperative vehicle following, to form so called platoons. Correspondingly, cooperative vehicles refer to vehicles that perform cooperative vehicle following. An important requirement for cooperative vehicle following is string-stability, which refers to the attenuation of disturbances in the upstream direction of the platoon, thereby avoiding the common "harmonica" effect in traffic jams. The main benefits of cooperative vehicles include improved highway throughput and safety. Typically, these vehicles make use of feedback control to achieve string-stable vehicle following at small inter-vehicle distances. Ideally, an automated vehicle is both highly versatile to navigate in a high variety of traffic scenarios, as well of being capable of small inter-vehicle distance vehicle following. The main contribution of this thesis is therefore a unified framework, that is used for both cooperative as well as autonomous driving. This allows the vehicle to maintain the versatility of the trajectory planning framework of the autonomous vehicle, while also benefiting from the properties of the cooperative vehicle. Furthermore, this approach simplifies the ad-hoc forming and breaking of platoons, which greatly benefits the ease of implementation of the framework, even in mixed traffic.

In Chapter 3, cooperative trajectory planning is discussed. A comparison is made between trajectory construction in which only a terminal state is used to generate trajectories and where an optimization over a time horizon is used. Although it is shown that the use of only a terminal state is capable of a transition from one steady state to another, it is not well suited in case of initial perturbations. In contrast, trajectory construction by means of optimization over a time horizon is capable of both. This trajectory planning method employs B-splines to efficiently characterize and communicate the planned trajectories between vehicles by means of control points. An important property of this method is that it allows for a closed form computation, such that guarantees on the required computation time can be given. An extension is also presented, in which the spacing policy is time dependent to allow for more comfortable gap closing. In addition to the behavior of the platoon of vehicles, also the behavior of the planned trajectories in consecutive plan steps is considered. In practice, the planner is implemented in a receding horizon fashion, such that the planned trajectories are updated during execution. Since the cooperative trajectories are

based on a planned trajectories of the preceding vehicle, it makes sense that consecutive trajectories of a single vehicle should be similar. This way, the host vehicle successfully anticipates future actions of the preceding vehicle. The desired similarity of these consecutive trajectories is referred to as temporal consistency. It was shown by means of simulations that the construction of cooperative trajectories can be achieved by means of the derived method, satisfying the first part of the first objective.

The construction of autonomous trajectories is presented in Chapter 4. It shows how properties of B-splines can be used to satisfy acceleration constraints, by identifying a feasible subset of trajectory control points. A set of candidate trajectories is constructed, which includes variations in velocity and lateral position. Additionally, various transients are included for each velocity and lateral position, such that both fast and aggressive trajectories as well as more comfortable slower trajectories are considered. Some other trajectory planners in literature require additional bookkeeping, or a specific choice for the cost function that prefers aggressive trajectories over comfortable trajectories to achieve temporal consistency. The presented trajectory construction method here demonstrates close to temporal consistent behavior without such requirement. In addition to the construction of autonomous trajectories, Chapter 4 also presents a unified framework that allows the automated vehicle to switch between autonomous driving and cooperative driving. Simulations demonstrate how the behavior of a platoon in traffic can be influenced. In particular, it was shown that a single parameter determines the preference of a platoon to either overtake a slower vehicle, or to form a platoon with it. Combining the research of both Chapters 3 and 4 the first research objective is thus achieved.

However, a major the benefit of cooperative vehicle following is to have a platoon of vehicles that is string-stable. String-stability implies the attenuation of disturbances in upstream direction of the platoon and is well studied in literature. Chapter 5 presents a mathematical string-stability analysis that considers the acceleration of each vehicle in the platoon. A novel approach of this chapter is that the updating nature of the planner with B-splines can be represented by means of discrete-time dynamics. These discrete-time dynamics describe how the control points of the B-spline trajectories evolve from one plan step to the next. At the same time, these control points describe the relevant signals in continuous-time by recognizing that they consist of piecewise B-splines. It is shown that linear discrete-time dynamics can be used to represent the behavior of the cooperative planner, and obtain the  $\mathcal{L}_2$ -norm of resulting continuous time signals. This is used to find a transfer function, that relates the  $\mathcal{L}_2$ -norm of the acceleration of one vehicle in the string to the next. This analysis shows that there do exist ‘exotic’ inputs to the lead vehicle that result in string unstable behavior. An example is given where temporal inconsistent behavior of the lead vehicle results in amplification of the  $\mathcal{L}_2$ -norm of the acceleration in the upstream direction. However, it can be reasoned that this temporal inconsistency



of the lead vehicle typically would not occur in practice. Instead, further mathematical analysis, in which the behavior of the lead vehicle is limited to one of two different planners for the lead vehicle, yields different results. The difference in the consecutively planned trajectories of these planners remains small, such that they are close to temporal consistent. Using this analysis it can be shown that a platoon of vehicles following a lead vehicle with either of these lead vehicle trajectory planners indeed is string stable. This result emphasizes the importance of temporal consistency in cooperative vehicle following. Additionally, planning delay, where the following vehicle plans its trajectory at a later time than the preceding vehicle, is also considered. Using a similar analysis as before, the required time gap in the spacing policy is computed to achieve string-stable behavior for a given plan delay. It is shown that even for significant plan delays string stable behavior can still be achieved for these temporal consistent lead vehicles, which satisfies the second research requirement.

After the design of a unified framework for both cooperative as well as autonomous driving, and the theoretical analysis, Chapter 6 presents the implementation of this framework on two full scale experimental demonstrator platforms in the form of two instrumented Renault Twizy's. This chapter discusses how radar information is integrated to increase the robustness of the method against estimation errors of either host or preceding vehicle. Since environmental perception of the experimental platform is limited, only longitudinal trajectories are planned and executed for both the lead vehicle as well as the following vehicle, while lateral actuation is still performed by the operator. Both the construction of trajectories for the lead vehicle as well as cooperative trajectories are shown. Both vehicle following as well as gap closing scenarios are considered, validating the framework laid out in Chapter 3 and Chapter 4. By satisfying the third research objective, it is shown that the development of this framework can also successfully be implemented in practice.

Finally, Chapter 7, presents an extension to the trajectory planning framework for automated cooperative vehicles. This extension intends to overcome the drawback of control bandwidth limitations of the trajectory planning approach. The time scaling extension utilizes radar information in between plan steps, such that the automated vehicle is capable of adjusting its control input without replanning. Discrepancies between the communicated and executed trajectory can occur due to either a faulty or damaged lead vehicle, or due to a V2V impairment. It is demonstrated that in case of such a discrepancy between the communicated and the executed trajectory of the preceding vehicle, the time-scaling mechanism helps to reduce the spacing error. As such, it provides a method of graceful degradation of the cooperative trajectory planning framework.

## 8.2 Recommendations

During the development of the cooperative trajectory planner of Chapter 3, a choice was made to minimize a given criterion that includes samples over the entire planning horizon in such a way that it will yield a solution for any knot vector. Although this choice demonstrated satisfactory results, different approaches could be considered. Among others, a minimization over the *integral* of the squared error can be considered instead of samples, by making use of Appendix A. These alternative planning methods can also be evaluated by the mathematical analysis of Chapter 5 to check for string-stability. Potentially, B-spline trajectory planning algorithms can be found which further improve temporal consistency and string-stability.

Aside from the cooperative planner, the autonomous planner can also be improved further. Note that the combination of the longitudinal and lateral control points directly describe a geometric spline in the Frenet frame. Currently, the generation of a lateral trajectory is created such that it is completely independent of the construction of the longitudinal trajectory. Instead, a method can also be conceived that utilizes the information of the longitudinal trajectory while constructing the lateral trajectory. This can for example be used to explicitly take into account obstacles and other road users during the construction of the trajectory, rather than only while checking its feasibility. This might benefit temporal consistency of the lateral trajectories.

Interesting observations are made in relation to the string-stability analysis, specifically in relation to temporal consistency of the lead vehicle. It is shown that for lead vehicles with certain trajectory planners, string-stability can be achieved, while for others this is not the case. It is of great interest to further investigate the planning algorithm of the lead vehicle for which string-stability is achieved, with the objective of identifying requirements for this lead vehicle planner. Additionally, it should be recognized that for the presented lead vehicles, a transition from steady state driving at an initial velocity to another velocity does result in string stable behavior in the  $\mathcal{L}_2$  sense. This analysis considers continuous-time signals with infinite timespan that start from the time at which the transition takes place. In practice, more than one transition will take place on such a time span. It is therefore of interest if the analysis can be extended to include multiple arbitrary velocity changes. This could potentially be achieved by means of a dwell time, in which the lead vehicle is not allowed to make another velocity change. Further research on this topic might result in useful insights. Finally, observations in Chapter 6 showed that the spacing error might not converge for certain situations. Additional insights might be gained from including the  $\mathcal{L}_2$ -norm of the error signal in the string-stability analysis. In literature on feedback control for Cooperative Adaptive Cruise Control, the transfer function related to the error and the acceleration signal between one vehicle to the next are identical under the assumption of a homogeneous

platoon. For the presented cooperative trajectory planning method, this is not necessarily true. Therefore, it would be interesting to include a requirement on the string-stability of the error signal, which might change the design of the planning algorithm.

The experimental validation of Chapter 6 only includes the validation of longitudinal trajectories. While this does validate the main contribution of this thesis in the form of trajectory planning for cooperative vehicle following, it does not validate the full functionality of the unified framework of autonomous and cooperative driving. Most importantly, the experimental set-up would greatly benefit additional environmental perception sensors, such as LIDAR and computer vision. By including camera vision, the lateral position of the host vehicle relative to the road can successfully be identified, which can in turn be used for lateral trajectory planning and vehicle control. In addition, it increases the robustness of matching the V2V messages to detections of obstacles (i.e., Most Important Object (MIO) selection). LIDAR will further increase the accuracy of relative positioning of vehicles with respect to each other and the environment. Furthermore, it is important for trajectory planning that low-level control is sufficiently accurate. Therefore, the accuracy of the mapping of the drivetrain torque should be increased, especially for regenerative braking. Finally, a small input delay exists in the demonstrator platforms, that is currently ignored in the implementation. Trajectory planning has the advantage that future inputs are known to the control system. This information can potentially be utilized to minimize the negative effects of the input delay.

Finally, the time scaling approach could be improved by increasing the time gap in the spacing policy, whenever time scaling is active for a certain amount of time. The loss of accurate trajectory information of the preceding vehicle could potentially lead to string-instability. By increasing the time gap this instability might be prevented. This idea is supported by the fact that traditional Adaptive Cruise Control is also capable of preventing string-instability, provided that the time gap is sufficiently large, and does so without prior knowledge of the preceding vehicles trajectory.

# Appendix A

## Integral of the product of two B-splines

A method for computing the squared sum of a B-spline is presented in [113]. Consider the B-spline  $f(t)$ , of degree  $p_f$ , and  $n_f + 1$  control points and the B-spline  $g(t)$  of degree  $p_g$  and  $n_g + 1$  control points,

$$f(t) = \mathbf{N}_{n_f, p_f}(t) \mathcal{P}_f, \quad (\text{A.1})$$

$$g(t) = \mathbf{N}_{n_g, p_g}(t) \mathcal{P}_g, \quad (\text{A.2})$$

defined on a knotvectors  $\mathbf{U}_f$  and  $\mathbf{U}_g$ , where  $\mathbf{U}_f$  and  $\mathbf{U}_g$  span the same space in  $u$ . The objective is to find

$$\int_0^t f(t)g(t) dt, \quad (\text{A.3})$$

for  $t \leq u_n$  (i.e. for a  $t$  within the span of  $u$  on which  $f(t)$  and  $g(t)$  are defined. Note that the integral of a single B-spline is found by means of (2.53), which is described as

$$\mathbf{C}^{(-d)}(t) = \mathbf{N}_{n+d, p+d}(t) \mathcal{P}^{(-d)}, \quad (\text{A.4})$$

where  $\mathcal{P}^{(-d)} = \Xi_{d0} \mathcal{P}$ .

By following [113] a number of possibilities exist to evaluate the integral of the product of the two B-splines. By making use of integration by part the

following equations can be found in a recursive manner for  $p_f \leq p_g$

$$\begin{aligned}\int_a^b f(t)g(t) dt &= \left[ f(t) \int g(t) dt \right]_a^b - \int_a^b \left( \dot{f}(t) \int g(t) dt \right) dt, \\ \int_a^b \left( \dot{f}(t) \int g(t) dt \right) dt &= \left[ \dot{f}(t) \iint g(t) dt^2 \right]_a^b - \int_a^b \left( \ddot{f}(t) \iint g(t) dt^2 \right) dt, \\ \int_a^b \left( \ddot{f}(t) \iint g(t) dt^2 \right) dt &= \left[ \ddot{f}(t) \iiint g(t) dt^3 \right]_a^b - \int_a^b \left( \dddot{f}(t) \iiint g(t) dt^3 \right) dt,\end{aligned}$$

such that for step  $d$  we find the following

$$\begin{aligned}\int_a^b \left( f^{(d)}(t) \int \dots \int g(t) dt^d \right) dt &= \left[ f^{(d)}(t) \int \dots \int g(t) dt^{d+1} \right]_a^b \\ &\quad - \int_a^b \left( f^{(d+1)}(t) \int \dots \int g(t) dt^{d+1} \right) dt. \quad (\text{A.5})\end{aligned}$$

Then observe that the following can be written for the integral of the product of  $f(t)$  and  $g(t)$  following the recursive equations above

$$\begin{aligned}\int_a^b f(t)g(t) dt &= \sum_{i=0}^{p_f} \left[ (-1)^i f^{(i)}(t) \int \dots \int g(t) dt^{i+1} \right]_a^b \\ &\quad + (-1)^{p_f+1} \int_a^b \left( f^{(p_f+1)}(t) \int \dots \int g(t) dt^{p_f+1} \right) dt. \quad (\text{A.6})\end{aligned}$$

Note that as  $f(t)$  is a spline of degree  $p_f$  and hence  $f^{(p_f+1)}(t) = 0, \forall t$ , which can be used to determine  $\int_a^b \left( f^{(p_f+1)}(t) \int \dots \int g(t) dt^{p_f+1} \right) dt$ . However, note that in order to satisfy the principle of calculus (on which integration by parts is based), it is required that

$$\int_a^b f^{(p_f+1)}(t) dt = f^{(p_f)}(b) - f^{(p_f)}(a). \quad (\text{A.7})$$

Hence the jumps that occur in  $f^{(p_f)}(t)$ , due to the zero-order basis functions should be taken into account. To this extent, consider that the integral of Dirac function  $\delta(t)$  can be used

$$\int_a^b \delta(t - \hat{u}) f(t) dt = \begin{cases} f(\hat{u}), & a < \hat{u} < b \\ 0 & \hat{u} < a \text{ or } \hat{u} > b \end{cases}, \quad \hat{u} \neq b, \quad \hat{u} \neq a, \quad (\text{A.8})$$

now denote  $\int \dots \int g(t) dt^{p_f+1} = g^{-(p_f+1)}(t)$  such that

$$\int_a^b \left( f^{(p_f+1)}(t) g^{-(p_f+1)}(t) \right) dt = \sum_{j=0}^{n_f-p_f-1} \left( P_{j+1,f}^{(p_f)} - P_{j,f}^{(p_f)} \right) \int_a^b \left( \delta(t - u_{j+p_f+1,f}) g^{-(p_f+1)}(t) \right) dt, \quad (\text{A.9})$$

where the summation loops over all internal knots of knotvector  $\mathbf{U}_f$  which is then written as a matrix multiplication

$$\begin{aligned} \int_a^b \left( f^{(p_f+1)}(t) g^{-(p_f+1)}(t) \right) dt \\ = \mathcal{P}_f^\top \Delta_{p_f 0, f}^\top \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} [\mathbf{R}(t)]_a^b \Xi_{(p_f+1)0, g} \mathcal{P}_g, \end{aligned} \quad (\text{A.10})$$

$$\mathbf{R}(t) = \begin{bmatrix} \int_a^b (\delta(t - u_{p_f+1, f}) \mathbf{N}_{n_g+(p_f+1), p_g+(p_f+1), g}) dt \\ \vdots \\ \int_a^b (\delta(t - u_{n_f, f}) \mathbf{N}_{n_g+(p_f+1), p_g+(p_f+1), g}) dt \end{bmatrix}, \quad (\text{A.11})$$

where  $\Xi_{(p_f+1)0, g}$  denotes the transformation of control points  $\mathcal{P}_g$  to  $\mathcal{P}_g^{-(p_f+1)}$ , based on the knotvector  $\mathbf{U}_g$ , and  $\Delta_{p_f 0, f}$  denotes the transformation of control points  $\mathcal{P}_f$  to  $\mathcal{P}_f^{(p_f+1)}$  based on the knotvector  $\mathbf{U}_f$ .

Then (A.6) can be written as

$$\int_a^b f(t) g(t) dt = \mathcal{P}_f^\top (\mathbf{Q}_{fg}(b) - \mathbf{Q}_{fg}(a)) \mathcal{P}_g, \quad (\text{A.12})$$

$$\begin{aligned} \mathbf{Q}_{fg}(t) = & \left( \sum_{i=0}^{p_f} \left( (-1)^i \Delta_{i0, f}^\top \mathbf{N}_{n-i+1, p-i+1, f}^\top(t) \mathbf{N}_{n+i, p+i, g}(t) \Xi_{i0}(t) \right) \right. \\ & \left. + (-1)^{(p_f+1)} \Delta_{p_f 0, f}^\top \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \mathbf{R}(t) \Xi_{(p_f+1)0, g} \right). \end{aligned} \quad (\text{A.13})$$



# Appendix B

## Planning Algorithm Properties

Aside from the algorithm presented in Section 3.3, different algorithms can also be constructed (e.g., by including more temporal points in the minimization of (5.31)). These planning algorithms can be written in the form.

$$\mathcal{P}_{i,k} = \mathbf{A}_i \mathcal{P}_{i,k-1} + \mathbf{B}_i \mathcal{P}_{i-1,k} + \mathbf{E}_i. \quad (\text{B.1})$$

However, to achieve certain desired behavior the dynamics corresponding to these algorithms should satisfy a number of criteria, which are briefly discussed next, as found in [137].

**Lemma B.1.** *To ensure continuity in states  $s_i(t)$ ,  $v_i(t)$  and  $a_i(t)$  the following requirements should hold for all vehicles:*

- $\begin{bmatrix} I_3 & 0 \end{bmatrix} A_i = \Upsilon(t_p)$ ,
- $\begin{bmatrix} I_3 & 0 \end{bmatrix} B_i = 0$ ,
- $\begin{bmatrix} I_3 & 0 \end{bmatrix} E_i = 0$ .

*Proof.* Assuming an update every  $t_p$ , such that  $t_{i,k} = t_{i,k} + t_p$ , then continuity in  $s_i(t)$ ,  $v_i(t)$  and  $a_i(t)$  requires that  $s_{i,k}(kt_p) = s_{i,k-1}(kt_p)$ , corresponding to (5.16), which leads to

$$\begin{bmatrix} I_3 & 0 \end{bmatrix} A_i \mathcal{P}_{i,k-1} + \begin{bmatrix} I_3 & 0 \end{bmatrix} B_i \mathcal{P}_{i-1,k} + \begin{bmatrix} I_3 & 0 \end{bmatrix} E_i = \Upsilon(t_p) \mathcal{P}_{i,k-1}, \quad (\text{B.2})$$

which should hold for arbitrary  $\mathcal{P}_{i,k-1}$  and  $\mathcal{P}_{i-1,k}$  such that

$$\begin{bmatrix} I_3 & 0 \end{bmatrix} B_i = 0, \quad \begin{bmatrix} I_3 & 0 \end{bmatrix} E_i, \quad \begin{bmatrix} I_3 & 0 \end{bmatrix} A_i = \Upsilon(t_p). \quad (\text{B.3})$$

□



**Lemma B.2.** *For follower vehicles to be capable of steady state vehicle following at velocity  $\bar{v}$ , with an inter-vehicle distance of  $d_i = c_i + L_i + h\bar{v}$ , requires that*

- $(I - A_i - B_i)\mathbb{1} = 0$ ,
- $E_i = -(L_i + c_i)B_i\mathbb{1}$ ,
- $(I - A_i - B_i)\Xi_{10|U_i^{(1)}}\mathbb{1} = (-t_p A_i + B_i h)\mathbb{1}$ ,

with  $\mathbb{1} = [1 \ \dots \ 1]^\top$  and where  $\Xi_{10|U_i^{(1)}} \in \mathbb{R}^{n+1 \times n}$  represents the matrix to integrate the velocity control points  $\mathcal{P}_i^{(1)}$  to obtain the offsetted position control points  $\mathcal{P}_{i,k}^{(1)} - P_{0,i,k} = \Xi_{10|U_i^{(1)}} \mathcal{P}_i^{(1)}$ .

*Proof.* With spacing error (3.6) the following dynamics can be written

$$\begin{aligned} \begin{bmatrix} e_{i,k}(t) \\ \dot{s}_{i,k} \\ \ddot{s}_{i,k} \end{bmatrix} &= \begin{bmatrix} \mathbf{N}_{n,p}(t - t_k) \\ 0 \\ 0 \end{bmatrix} (\mathcal{P}_{i-1,k} - \mathcal{P}_{i,k}) \\ &+ \begin{bmatrix} -h\mathbf{N}_{n-1,p-1}(t - t_k) \\ \mathbf{N}_{n-1,p-1}(t - t_k) \\ \mathbf{N}_{n-2,p-2}(t - t_k)\Delta_{21} \end{bmatrix} \Delta_{10}\mathcal{P}_{i,k} - \begin{bmatrix} L_i + c_i \\ 0 \\ 0 \end{bmatrix}. \end{aligned} \quad (\text{B.4})$$

For  $e_i = 0$  and  $v_i = \bar{v}$  to be an equilibrium

$$\Delta_{10}\mathcal{P}_{i,k} = \bar{v}\mathbb{1}, \quad (\mathcal{P}_{i-1,k} - \mathcal{P}_{i,k}) = (h\bar{v} + L_i + c_i)\mathbb{1}, \forall i, k, \quad (\text{B.5})$$

$\mathbf{N}_{n,p}(t)\mathbb{1} = \mathbf{N}_{n-1,p-1}(t)\mathbb{1} = 1$ . Then note that

$$\mathcal{P}_{i,k} = P_{0,i,k}\mathbb{1} + \Xi_{10|U_i^{(1)}}\Delta_{10}\mathcal{P}_{i,k}, \quad (\text{B.6})$$

$$\mathcal{P}_{i-1,k} = \left( P_{0,i,k}\mathbb{1} + \Xi_{10|U_i^{(1)}}\Delta_{10}\mathcal{P}_{i,k} \right) + (h\bar{v} + L_i + c_i)\mathbb{1}, \quad (\text{B.7})$$

where (B.7) is found by combining (B.5) and (B.6). Then substituting (B.5) and (B.7) in (B.1) results in

$$\begin{aligned} P_{0,i,k}\mathbb{1} + \Xi_{10|U_i^{(1)}}\Delta_{10}\mathcal{P}_{i,k} &= A_i(P_{0,i,k-1}\mathbb{1} + \Xi_{10|U_i^{(1)}}\Delta_{10}\mathcal{P}_{i,k-1}) \\ &+ B_i \left( (P_{0,i,k}\mathbb{1} + \Xi_{10|U_i^{(1)}}\Delta_{10}\mathcal{P}_{i,k}) + (h\bar{v} + L_i + c_i)\mathbb{1} \right) + E_i. \end{aligned} \quad (\text{B.8})$$

Then substituting  $\Delta_{10}\mathcal{P}_{i,k} = \Delta_{10}\mathcal{P}_{i-1,k} = \bar{v}\mathbb{1} = \bar{v}\mathbb{1}$  and collecting terms with  $\bar{v}$  yields

$$\begin{aligned} \left( (I - A_i - B_i)\Xi_{10|U_i^{(1)}} - B_i h \right) \bar{v}\mathbb{1} &= A_i(P_{0,i,k-1} - P_{0,i,k})\mathbb{1} \\ &- (I - B_i - A_i)\mathbb{1}P_{0,i,k} + B_i(L_i + c_i)\mathbb{1} + E_i, \end{aligned} \quad (\text{B.9})$$

which should also hold at standstill when  $\bar{v} = 0$  and  $P_{0,i,k-1} = P_{0,i,k}$ , and for arbitrary  $P_{0,i,k}$  such that

$$(I - B_i - A_i) \mathbb{1} = 0 \quad (\text{B.10})$$

$$0 = B_i(L_i + c_i)\mathbb{1} + E_i. \quad (\text{B.11})$$

Then finally using  $P_{0,i,k-1} - P_{0,i,k} = -t_p \bar{v}$  and substituting (B.10) and (B.11) in (B.9) to find

$$\left( (I - A_i - B_i) \Xi_{10|\mathbf{U}_i^{(1)}} - B_i h \right) \mathbb{1} \bar{v} = A_i \mathbb{1} (-t_p \bar{v}) \quad (\text{B.12})$$

$$(I - A_i - B_i) \Xi_{10|\mathbf{U}_i^{(1)}} \mathbb{1} = (-t_p A_i + B_i h) \mathbb{1}. \quad (\text{B.13})$$

□



# Bibliography

- [1] SAE On-Road Automated Vehicle Standards, “Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems,” *SAE International*, 2018. DOI: 10.4271/J3016\_201806.
- [2] T. van der Sande and H. Nijmeijer, “From Cooperative to Autonomous Vehicles,” in *Sensing and Control for Autonomous Vehicles*, ser. Lecture Notes in Control and Information Sciences, Dordrecht: Springer, 2017, pp. 435–452, ISBN: 978-3-319-55372-6.
- [3] J. Ploeg, “Analysis and design of controllers for cooperative and automated driving,” Ph.D dissertation, Eindhoven University of Technology, Eindhoven, 2014, ISBN: 978-94-6259-104-2. DOI: 10.6100/IR772224.
- [4] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, “A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles,” *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016. DOI: 10.1109/TIV.2016.2578706.
- [5] L. Han, H. Yashiro, H. T. N. Nejad, Q. H. Do, and S. Mita, “Bézier curve based path planning for autonomous vehicle in urban environment,” in *2010 IEEE Intelligent Vehicles Symposium*, Jun. 2010, pp. 1036–1042. DOI: 10.1109/IVS.2010.5548085.
- [6] J. w Choi, R. Curry, and G. Elkaim, “Path Planning Based on Bezier Curve for Autonomous Ground Vehicles,” in *Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*, Oct. 2008, pp. 158–166. DOI: 10.1109/WCECS.2008.27.
- [7] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, and P. Mahoney, “Stanley: The robot that won the DARPA Grand Challenge,” *J. Field Robotics*, vol. 23, pp. 661–692, Jan. 2006. DOI: 10.1002/rob.20147.

- [8] M. Likhachev and D. Ferguson, "Planning Long Dynamically Feasible Maneuvers for Autonomous Vehicles," *The Int'l Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, Aug. 1, 2009. DOI: 10.1177/0278364909340445.
- [9] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments," *Journal of Field Robotics*, vol. 25, pp. 939–960, 11–12 Nov. 17, 2008. DOI: 10.1002/rob.20265.
- [10] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marciel, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, "Junior: The Stanford Entry in the Urban Challenge," in *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, ser. Springer Tracts in Advanced Robotics, M. Buehler, K. Iagnemma, and S. Singh, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 91–123, ISBN: 978-3-642-03991-1.
- [11] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments," *The Int'l Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, Apr. 1, 2010. DOI: 10.1177/0278364909359210.
- [12] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A Review of Motion Planning Techniques for Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016. DOI: 10.1109/TITS.2015.2498841.
- [13] U. Schwesinger, M. Rufli, P. Furgale, and R. Siegwart, "A sampling-based partial motion planning framework for system-compliant navigation along a reference path," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2013, pp. 391–396. DOI: 10.1109/IVS.2013.6629500.
- [14] X. Li, Z. Sun, A. Kurt, and Q. Zhu, "A sampling-based local trajectory planner for autonomous driving along a reference path," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, Jun. 2014, pp. 376–381. DOI: 10.1109/IVS.2014.6856397.
- [15] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, Nov. 2015. DOI: 10.1016/j.trc.2015.09.011.
- [16] X. Li, Z. Sun, Z. He, Q. Zhu, and D. Liu, "A practical trajectory planning framework for autonomous ground vehicles driving in urban environments," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2015, pp. 1160–1166. DOI: 10.1109/IVS.2015.7225840.
- [17] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenét Frame," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 987–993. DOI: 10.1109/ROBOT.2010.5509799.

- [18] M. Werling, S. Kammel, J. Ziegler, and L. Gröll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *The Int'l Journal of Robotics Research*, vol. 31, no. 3, pp. 346–359, Mar. 2012. DOI: 10.1177/0278364911423042.
- [19] J. Ziegler and C. Stiller, "Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2009, pp. 1879–1884. DOI: 10.1109/IRoS.2009.5354448.
- [20] M. McNaughton, C. Urmson, J. M. Dolan, and J. W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 4889–4895. DOI: 10.1109/ICRA.2011.5980223.
- [21] M. Mcnaughton, "Parallel Algorithms for Real-time Motion Planning," Ph.D dissertation, Carnegie Mellon University, The Robotics Institute, Pittsburgh, PA, USA, 2011.
- [22] E. Wachter, A. Schmeitz, F. Bruzelius, and M. Alirezaei, "Path Control in Limits of Vehicle Handling: A Sensitivity Analysis," in *Advances in Dynamics of Vehicles on Roads and Tracks*, M. Klomp, F. Bruzelius, J. Nielsen, and A. Hillemyr, Eds., ser. Lecture Notes in Mechanical Engineering, Cham: Springer International Publishing, 2020, pp. 1089–1095, ISBN: 978-3-030-38077-9. DOI: 10.1007/978-3-030-38077-9\_126.
- [23] K. Kritayakirana and J. C. Gerdes, "Autonomous vehicle control at the limits of handling," *International Journal of Vehicle Autonomous Systems*, vol. 10, no. 4, pp. 271–296, Jan. 2012. DOI: 10.1504/IJVAS.2012.051270.
- [24] K. Kritayakirana and J. C. Gerdes, "Using the centre of percussion to design a steering controller for an autonomous race car," *Vehicle System Dynamics*, vol. 50, pp. 33–51, sup1 Jan. 1, 2012. DOI: 10.1080/00423114.2012.672842.
- [25] J. Y. Goh and J. C. Gerdes, "Simultaneous stabilization and tracking of basic automobile drifting trajectories," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2016, pp. 597–602. DOI: 10.1109/IVS.2016.7535448.
- [26] M. Werling and D. Liccardo, "Automatic collision avoidance using model-predictive online optimization," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, Dec. 2012, pp. 6309–6314. DOI: 10.1109/CDC.2012.6426612.
- [27] J. Ploeg, B. T. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *14th International IEEE Conference on Intelligent Transportation Systems*, IEEE, 2011, pp. 260–265. DOI: 10.1109/ITSC.2011.6082981.
- [28] D. Swaroop and J. K. Hedrick, "String stability of interconnected systems," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 349–357, Mar. 1996. DOI: 10.1109/9.486636.
- [29] D. Swaroop, "String Stability Of Interconnected Systems: An Application To Platooning In Automated Highway Systems," Ph.D dissertation, University of California at Berkeley, Berkeley, 1994.

- [30] J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Lp String Stability of Cascaded Systems: Application to Vehicle Platooning," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 2, pp. 786–793, Mar. 2014. DOI: 10.1109/TCST.2013.2258346.
- [31] A. A. Alam, A. Gattami, and K. H. Johansson, "An experimental study on the fuel reduction potential of heavy duty vehicle platooning," in *13th International IEEE Conference on Intelligent Transportation Systems*, Sep. 2010, pp. 306–311. DOI: 10.1109/ITSC.2010.5625054.
- [32] A. Bayuwindra, L. Aakre, J. Ploeg, and H. Nijmeijer, "Combined lateral and longitudinal CACC for a unicycle-type platoon," in *IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2016, pp. 527–532. DOI: 10.1109/IVS.2016.7535437.
- [33] E. Lefeber, J. Ploeg, and H. Nijmeijer, "A Spatial Approach to Control of Platooning Vehicles: Separating Path-Following from Tracking," *IFAC-PapersOnLine*, 20th IFAC World Congress, vol. 50, no. 1, pp. 15 000–15 005, Jul. 1, 2017. DOI: 10.1016/j.ifacol.2017.08.2568.
- [34] A. I. M. Medina, N. van de Wouw, and H. Nijmeijer, "Automation of a T-intersection Using Virtual Platoons of Cooperative Autonomous Vehicles," in *IEEE 18th International Conference on Intelligent Transportation Systems*, Sep. 2015, pp. 1696–1701. DOI: 10.1109/ITSC.2015.275.
- [35] A. I. Morales Medina, N. van de Wouw, and H. Nijmeijer, "Cooperative Intersection Control Based on Virtual Platooning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1727–1740, Jun. 2018. DOI: 10.1109/TITS.2017.2735628.
- [36] C. Bergenheim, S. Shladover, and E. Coelingh, "Overview of platooning systems," in *19th ITS World Congress Proceedings*, 2012, p. 8.
- [37] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, "Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 695–708, Jul. 2000. DOI: 10.1109/87.852914.
- [38] E. van Nunen, M. R. J. A. E. Kwakernaat, J. Ploeg, and B. D. Netten, "Cooperative Competition for Future Mobility," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1018–1025, Sep. 2012. DOI: 10.1109/TITS.2012.2200475.
- [39] A. Geiger, M. Lauer, F. Moosmann, B. Ranft, H. Rapp, C. Stiller, and J. Ziegler, "Team AnnieWAY's Entry to the 2011 Grand Cooperative Driving Challenge," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1008–1017, Sep. 2012. DOI: 10.1109/TITS.2012.2189882.
- [40] J. Ploeg, E. Semsar-Kazerooni, A. I. Morales Medina, J. F. C. M. de Jongh, J. van de Sluis, A. Voronov, C. Englund, R. J. Bril, H. Salunkhe, I. Arrúe, A. Ruano, L. García-Sol, E. van Nunen, and N. van de Wouw, "Cooperative Automated Maneuvering at the 2016 Grand Cooperative Driving Challenge," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1213–1226, Apr. 2018. DOI: 10.1109/TITS.2017.2765669.

- [41] M. Aramrattana, J. Detournay, C. Englund, V. Frimodig, O. U. Jansson, T. Larsson, W. Mostowski, V. D. Rodríguez, T. Rosenstatter, and G. Shahanoor, "Team Halmstad Approach to Cooperative Driving in the Grand Cooperative Driving Challenge 2016," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1248–1261, Apr. 2018. DOI: 10.1109/TITS.2017.2752359.
- [42] S. Öncü, N. van de Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Tradeoffs between control and network specifications," in *14th International IEEE Conference on Intelligent Transportation Systems*, IEEE, 2011, pp. 2051–2056. DOI: 10.1109/ITSC.2011.6082894.
- [43] J. C. Zegers, E. Semsar-Kazerooni, J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Consensus Control for Vehicular Platooning With Velocity Constraints," *IEEE Transactions on Control Systems Technology*, 2018. DOI: 10.1109/TCST.2017.2720141.
- [44] J. Lunze, "Design of the Communication Structure of Cooperative Adaptive Cruise Controllers," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2019. DOI: 10.1109/TITS.2019.2941080.
- [45] Y. Zheng, S. Eben Li, J. Wang, D. Cao, and K. Li, "Stability and Scalability of Homogeneous Vehicular Platoon: Study on the Influence of Information Flow Topologies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 14–26, Jan. 2016. DOI: 10.1109/TITS.2015.2402153.
- [46] Y. Zheng, S. E. Li, K. Li, and L.-Y. Wang, "Stability Margin Improvement of Vehicular Platoon Considering Undirected Topology and Asymmetric Control," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1253–1265, Jul. 2016. DOI: 10.1109/TCST.2015.2483564.
- [47] X. Qian, A. d. L. Fortelle, and F. Moutarde, "A hierarchical Model Predictive Control framework for on-road formation control of autonomous vehicles," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2016, pp. 376–381. DOI: 10.1109/IVS.2016.7535413.
- [48] J. I. Ge and G. Orosz, "Dynamics of connected vehicle systems with delayed acceleration feedback," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 46–64, Sep. 1, 2014. DOI: 10.1016/j.trc.2014.04.014.
- [49] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful Degradation of Cooperative Adaptive Cruise Control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 488–497, Feb. 2015. DOI: 10.1109/TITS.2014.2349498.
- [50] E. van Nunen, J. Verhaegh, E. Silvas, E. Semsar-Kazerooni, and N. van de Wouw, "Robust model predictive cooperative adaptive cruise control subject to V2V impairments," in *IEEE 20th International Conference on Intelligent Transportation Systems*, Oct. 2017, pp. 1–8. DOI: 10.1109/ITSC.2017.8317758.
- [51] E. van Nunen, J. Reinders, E. Semsar-Kazerooni, and N. van de Wouw, "String Stable Model Predictive Cooperative Adaptive Cruise Control for Heterogeneous Platoons," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 186–196, Jun. 2019. DOI: 10.1109/TIV.2019.2904418.



- [52] “I-Cave – Integrated cooperative automated vehicles.” (2016), [Online]. Available: <https://i-cave.nl/> (visited on 09/21/2020).
- [53] F. Remmen, I. Cara, E. de Gelder, and D. Willemsen, “Cut-in Scenario Prediction for Automated Vehicles,” in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Sep. 2018, pp. 1–7. DOI: 10.1109/ICVES.2018.8519594.
- [54] C. Lu, M. J. G. van de Molengraft, and G. Dubbelman, “Monocular Semantic Occupancy Grid Mapping With Convolutional Variational Encoder–Decoder Networks,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 445–452, Apr. 2019. DOI: 10.1109/LRA.2019.2891028.
- [55] C. Lu and G. Dubbelman, “Learning to complete partial observations from unpaired prior knowledge,” *Pattern Recognition*, vol. 107, p. 107426, Nov. 1, 2020. DOI: 10.1016/j.patcog.2020.107426.
- [56] R. van Hoek, J. Ploeg, and H. Nijmeijer, “Gap Closing for Cooperative Driving in Automated Vehicles using B-splines for Trajectory Planning,” in *The 31st IEEE Intelligent Vehicles Symposium (IV2020)*, Jan. 2020. DOI: 10.1109/IV47402.2020.9304732.
- [57] W. Scholte, “Control for cooperative merging maneuvers into platoons at highway on-ramps,” PhD. Dissertation, Eindhoven University of Technology, Department of Mechanical Engineering, Eindhoven, 2022 (in preparation).
- [58] W. Schinkel, “Extended automated driving features: An experimental approach,” PhD. Dissertation, Eindhoven University of Technology, Department of Mechanical Engineering, Eindhoven, 2021 (in preparation).
- [59] B. Alves Beirigo, “Dynamic Fleet Management for Autonomous Vehicles: Learning- and optimization-based strategies,” PhD. Dissertation, Delft University of Technology, 2021.
- [60] F. Lampel, R. F. Tigrek, A. Alvarado, and F. M. J. Willems, “A Performance Enhancement Technique for a Joint FMCW RadCom System,” in *2019 16th European Radar Conference (EuRAD)*, Oct. 2019, pp. 169–172.
- [61] D. Dey, “External communication for self-driving cars: Designing for encounters between automated vehicles and pedestrians,” PhD. Dissertation, Eindhoven University of Technology, Department of Industrial Design, Dec. 18, 2020.
- [62] F. Walker, “To trust or not to trust?: Assessment and calibration of driver trust in automated vehicles,” PhD. Dissertation, University of Twente, Feb. 3, 2021.
- [63] A. Boelhouwer, “Exploring, developing and evaluating in-car HMI to support appropriate use of automated cars,” PhD. Dissertation, University of Twente, Jan. 20, 2021.
- [64] A. C. Serban, “Designing Safety Critical Software Systems to Manage Inherent Uncertainty,” in *2019 IEEE International Conference on Software Architecture Companion (ICSA-C)*, Mar. 2019, pp. 246–249. DOI: 10.1109/ICSA-C.2019.00051.

- [65] F. N. Hoogeboom, "Safety of automated vehicles: design, implementation, and analysis," PhD. Dissertation, Eindhoven University of Technology, Department of Mechanical Engineering, Eindhoven, Oct. 27, 2020, ISBN: 978-90-386-5141-5.
- [66] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A Review of Motion Planning for Highway Autonomous Driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, May 2020. DOI: 10.1109/TITS.2019.2913998.
- [67] Z. Huang, D. Chu, C. Wu, and Y. He, "Path Planning and Cooperative Control for Automated Vehicle Platoon Using Hybrid Automata," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 959–974, Mar. 2019. DOI: 10.1109/TITS.2018.2841967.
- [68] S. Gao, A. Lim, and D. Bevly, "An empirical study of DSRC V2V performance in truck platooning scenarios," *Digital Communications and Networks, Next Generation Wireless Communication Technologies*, vol. 2, no. 4, pp. 233–244, Nov. 1, 2016. DOI: 10.1016/j.dcan.2016.10.003.
- [69] E. Dijkstra W, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959. DOI: 10.1007/BF01386390.
- [70] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, Jul. 1968. DOI: 10.1109/TSSC.1968.300136.
- [71] P. E. Hart, N. J. Nilsson, and B. Raphael, "Correction to 'a formal basis for the heuristic determination of minimum cost paths'," *ACM SIGART Bulletin*, vol. 37, pp. 28–29, 1972. DOI: 10.1145/1056777.1056779.
- [72] A. Stentz, "Optimal and Efficient Path Planning for Partially Known Environments," in *Intelligent Unmanned Ground Vehicles: Autonomous Navigation Research at Carnegie Mellon*, ser. The Springer International Series in Engineering and Computer Science, M. H. Hebert, C. Thorpe, and A. Stentz, Eds., Boston, MA: Springer US, 1997, pp. 203–220, ISBN: 978-1-4615-6325-9. DOI: 10.1007/978-1-4615-6325-9\_11.
- [73] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path Planning with Modified a Star Algorithm for a Mobile Robot," *Procedia Engineering, Modelling of Mechanical and Mechatronic Systems*, vol. 96, pp. 59–69, Jan. 1, 2014. DOI: 10.1016/j.proeng.2014.12.098.
- [74] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, Aug. 1996. DOI: 10.1109/70.508439.
- [75] S. M. Lavalle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," Iowa State University, Department of Computer Science, Ames, 1998.
- [76] S. M. LaValle and J. J. Kuffner, "Randomized Kinodynamic Planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, May 1, 2001. DOI: 10.1177/02783640122067453.

- [77] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, "LQR-RRT\*: Optimal sampling-based motion planning with automatically derived extension heuristics," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 2537–2542. DOI: 10.1109/ICRA.2012.6225177.
- [78] K. Macek, M. Becker, and R. Siegwart, "Motion Planning for Car-Like Vehicles in Dynamic Urban Scenarios," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2006, pp. 4375–4380. DOI: 10.1109/IR0S.2006.282013.
- [79] Y. Kuwata, G. A. Fiore, J. Teo, E. Frazzoli, and J. P. How, "Motion planning for urban driving using RRT," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2008, pp. 1681–1686. DOI: 10.1109/IR0S.2008.4651075.
- [80] J. h Jeon, S. Karaman, and E. Frazzoli, "Anytime computation of time-optimal off-road vehicle maneuvers using the RRT\*," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, Dec. 2011, pp. 3276–3282. DOI: 10.1109/CDC.2011.6161521.
- [81] L. Garrote, C. Premebida, M. Silva, and U. Nunes, "An RRT-based navigation approach for mobile robots and automated vehicles," in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, Jul. 2014, pp. 326–331. DOI: 10.1109/INDIN.2014.6945533.
- [82] L. Worms, "Semi-continuous curvature path planning for an automated parking system," MSc. Thesis, Eindhoven University of Technology, Department of Mechanical Engineering, Eindhoven, Apr. 19, 2019.
- [83] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The Int'l Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, Jun. 1, 2011. DOI: 10.1177/0278364911406761.
- [84] D. J. Webb and J. van den Berg, "Kinodynamic RRT\*: Asymptotically optimal motion planning for robots with linear dynamics," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 5054–5061. DOI: 10.1109/ICRA.2013.6631299.
- [85] Y. Li, Z. Littlefield, and K. E. Bekris, "Sparse Methods for Efficient Asymptotically Optimal Kinodynamic Planning," in *Algorithmic Foundations of Robotics XI*, ser. Springer Tracts in Advanced Robotics, Springer, Cham, 2015, pp. 263–282, ISBN: 978-3-319-16594-3 978-3-319-16595-0.
- [86] D. Althoff, M. Buss, A. Lawitzky, M. Werling, and D. Wollherr, "On-line Trajectory Generation for Safe and Optimal Vehicle Motion Planning," *Autonomous Mobile Systems 2012*, Informatik Aktuell, pp. 99–107, 2012. DOI: 10.1007/978-3-642-32217-4\_11.
- [87] X. Li, Z. Sun, D. Cao, D. Liu, and H. He, "Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles," *Mechanical Systems and Signal Processing*, Signal Processing and Control Challenges for Smart Vehicles, vol. 87, pp. 118–137, Mar. 15, 2017. DOI: 10.1016/j.ymssp.2015.10.021.

- [88] H. B. Pacejka and I. Besselink, *Tire and Vehicle Dynamics*, 3. ed, ser. Engineering Automotive Engineering. Amsterdam: Elsevier/Butterworth-Heinemann, 2012, 632 pp., ISBN: 978-0-08-097016-5.
- [89] K. J. Kyriakopoulos and G. N. Saridis, "Minimum jerk path generation," in *1988 IEEE International Conference on Robotics and Automation Proceedings*, Apr. 1988, 364–369 vol.1. DOI: 10.1109/R0BOT.1988.12075.
- [90] J. Ziegler and C. Stiller, "Fast collision checking for intelligent vehicle motion planning," in *2010 IEEE Intelligent Vehicles Symposium*, Jun. 2010, pp. 518–522. DOI: 10.1109/IVS.2010.5547976.
- [91] S. Gottschalk, M. C. Lin, and D. Manocha, "OBBTree: A hierarchical structure for rapid interference detection," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '96*, Not Known: ACM Press, 1996, pp. 171–180, ISBN: 978-0-89791-746-9. DOI: 10.1145/237170.237244.
- [92] J. Huynh, *Separating Axis Theorem for Oriented Bounding Boxes*, Sep. 15, 2009. [Online]. Available: <http://www.jkh.me/tutorials.html> (visited on 07/11/2020).
- [93] J. Wei, J. Dolan, and B. Litkouhi, "A prediction- and cost function-based algorithm for robust autonomous freeway driving," in *2010 IEEE Intelligent Vehicles Symposium*, San Diego, CA, USA, 21-24-2010, ISBN: 978-1-4244-7866-8. DOI: 10.1109/IVS.2010.5547988.
- [94] E. Snapper, H. Hellendoorn, E. Semsar-Kazerooni, and M. Alirezaei, "Unified Path Planning and Control for Autonomous Vehicles using Artificial Potential Fields and Model Predictive Control," in *Advanced Vehicle Control: Proceedings of the 14th International Symposium on Advanced Vehicle Control (AVEC'18)*, May 2018.
- [95] Z. Huang, D. Chu, C. Wu, and Y. He, "Path Planning and Cooperative Control for Automated Vehicle Platoon Using Hybrid Automata," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 959–974, Mar. 2019. DOI: 10.1109/TITS.2018.2841967.
- [96] T. Sauer, "Splines in Industrial Applications," Lecture Notes, Lecture Notes, Gießen, Feb. 28, 2007. [Online]. Available: <https://www.fim.uni-passau.de/fileadmin/dokumente/fakultaeten/fim/lehrstuhl/sauer/geyer/SplinesinIndustry.pdf> (visited on 02/24/2021).
- [97] J. w Choi, R. E. Curry, and G. H. Elkaim, "Curvature-continuous trajectory generation with corridor constraint for autonomous ground vehicles," in *49th IEEE Conference on Decision and Control (CDC)*, Dec. 2010, pp. 7166–7171. DOI: 10.1109/CDC.2010.5718154.
- [98] K. Renny Simba, N. Uchiyama, and S. Sano, "Real-time smooth trajectory generation for nonholonomic mobile robots using Bézier curves," *Robotics and Computer-Integrated Manufacturing*, vol. 41, pp. 31–42, Oct. 1, 2016. DOI: 10.1016/j.rcim.2016.02.002.
- [99] S. Taketoshi, U. Riku, and M. Takashi, "Automatic Two-Lane Path Generation for Autonomous Vehicles using Quartic B-spline Curves," *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2018. DOI: 10.1109/TIV.2018.2874532.

- [100] Q. J. Ge, L. Srinivasan, and J. Rastegar, "Low-harmonic rational Bézier curves for trajectory generation of high-speed machinery," *Computer Aided Geometric Design*, vol. 14, no. 3, pp. 251–271, Apr. 1, 1997. DOI: 10.1016/S0167-8396(96)00032-5.
- [101] K. G. Jolly, R. Sreerama Kumar, and R. Vijayakumar, "A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits," *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 23–33, Jan. 31, 2009. DOI: 10.1016/j.robot.2008.03.009.
- [102] W. Van Loock, G. Pipeleers, and J. Swevers, "B-spline parameterized optimal motion trajectories for robotic systems with guaranteed constraint satisfaction," *Mechanical Sciences*, vol. 6, no. 2, pp. 163–171, Sep. 1, 2015. DOI: 10.5194/ms-6-163-2015.
- [103] T. Mercy, R. V. Parys, and G. Pipeleers, "Spline-Based Motion Planning for Autonomous Guided Vehicles in a Dynamic Environment," *IEEE Transactions on Control Systems Technology*, vol. PP, no. 99, pp. 1–8, 2017. DOI: 10.1109/TCST.2017.2739706.
- [104] R. Van Parys and G. Pipeleers, "Spline-Based Motion Planning in an Obstructed 3D Environment," *IFAC-PapersOnLine*, 20th IFAC World Congress, vol. 50, no. 1, pp. 8668–8673, Jul. 1, 2017. DOI: 10.1016/j.ifacol.2017.08.1525.
- [105] P. Bézier, "Essai de definition numerique des courbes et des surfaces experimentales: contribution a l'etude des proprietes des courbes et des surfaces parametriques polynomiales a coefficients vectoriels," Universite Pierre et Marie Curie, Paris, 1977.
- [106] P. de Casteljaeu, "Courbes et surfaces à pôles," *André Citroën, Automobiles SA, Paris*, vol. 66, 1963.
- [107] L. Piegl and W. Tiller, *The NURBS Book*, 2nd ed., ser. Monographs in Visual Communication. Berlin, Heidelberg: Springer-Verlag, 1997, ISBN: 978-3-642-97385-7.
- [108] J. R. Manning, "Continuity conditions for spline curves," *Comput J*, vol. 17, no. 2, pp. 181–186, Jan. 1, 1974. DOI: 10.1093/comjnl/17.2.181.
- [109] B. A. Barsky and T. D. DeRose, "Geometric continuity of parametric curves: Three equivalent characterizations," *IEEE Computer Graphics and Applications*, vol. 9, no. 6, pp. 60–69, Nov. 1989. DOI: 10.1109/38.41470.
- [110] B. A. Barsky, *Computer Graphics and Geometric Modeling Using Beta-Splines*, ser. Computer Science Workbench. Berlin Heidelberg: Springer-Verlag, 1988, ISBN: 978-3-642-72294-3. DOI: 10.1007/978-3-642-72292-9.
- [111] T. N. T. Goodman, "Properties of Beta-splines," *Journal of Approximation Theory*, vol. 44, no. 2, pp. 132–153, Jun. 1, 1985. DOI: 10.1016/0021-9045(85)90076-0.
- [112] C. de Boor, "On calculating with B-splines," *Journal of Approximation theory*, vol. 6, no. 1, pp. 50–62, Sep. 28, 1972. DOI: 10.1016/0021-9045(72)90080-9.
- [113] A. H. Vermeulen, R. H. Bartels, and G. R. Heppler, "Integrating Products of B-Splines," *SIAM J. Sci. and Stat. Comput.*, vol. 13, no. 4, pp. 1025–1038, Jul. 1, 1992. DOI: 10.1137/0913060.

- [114] C. de Boor, *A Practical Guide to Splines*, Revised Edition, ser. Applied Mathematical Sciences. New York, NY, USA: Springer-Verlag, 2001, vol. 27, ISBN: 0-387-95366-3.
- [115] W. Nelson, "Continuous-curvature paths for autonomous vehicles," in *1989 International Conference on Robotics and Automation Proceedings*, May 1989, 1260–1264 vol.3. DOI: 10.1109/ROBOT.1989.100153.
- [116] X. Chen, Y. Zhou, Z. Shu, and H. Su, "Improved Algebraic Algorithm on Point projection for Beziercurves," in *Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2007)*, Aug. 2007, pp. 158–163. DOI: 10.1109/IMSCCS.2007.17.
- [117] R. van Hoek, *Bézier Toolbox*, version 1.0.2, MATLAB Central File Exchange, May 14, 2020. [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/69302-bezier-toolbox>, MATLABCentralFileExchange. Retrieved October 6, 2020.
- [118] E. Velenis and P. Tsiotras, "Optimal velocity profile generation for given acceleration limits: Theoretical analysis," in *Proceedings of the 2005, American Control Conference, 2005.*, Jun. 2005, 1478–1483 vol. 2. DOI: 10.1109/ACC.2005.1470174.
- [119] E. Velenis and P. Tsiotras, "Optimal Velocity Profile Generation for Given Acceleration Limits; The Half-Car Model Case," in *Proceedings of the IEEE International Symposium on Industrial Electronics, 2005. ISIE 2005.*, vol. 1, Jun. 2005, pp. 361–366. DOI: 10.1109/ISIE.2005.1528937.
- [120] T. Völkl, M. Muehlmeier, and H. Winner, "Extended Steady State Lap Time Simulation for Analyzing Transient Vehicle Behavior," *SAE Int. J. Passeng. Cars - Mech. Syst.*, vol. 6, no. 1, pp. 283–292, Apr. 8, 2013. DOI: 10.4271/2013-01-0806.
- [121] S. Van Koutrik, "Optimal Control for Race Car Minimum Time Maneuvering," MSc. Thesis, Delft University of Technology, Delft, 2015.
- [122] R. van Hoek, J. Ploeg, and H. Nijmeijer, "Cooperative Driving in Automated Vehicles using B-splines for Trajectory Planning," *IEEE Transactions on Intelligent Vehicles (Accepted)*, 2021. DOI: 10.1109/TIV.2021.3072679.
- [123] D. Swaroop and J. K. Hedrick, "Constant Spacing Strategies for Platooning in Automated Highway Systems," *J. Dyn. Sys., Meas., Control*, vol. 121, no. 3, pp. 462–470, Sep. 1, 1999. DOI: 10.1115/1.2802497.
- [124] G. J. L. Naus, R. P. A. Vugts, J. Ploeg, M. J. G. van de Molengraft, and M. Steinbuch, "String-Stable CACC Design and Experimental Validation: A Frequency-Domain Approach," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 9, pp. 4268–4279, Nov. 2010. DOI: 10.1109/TVT.2010.2076320.
- [125] J. Ploeg, D. P. Shukla, N. van de Wouw, and H. Nijmeijer, "Controller Synthesis for String Stability of Vehicle Platoons," *IEEE Trans. Intell. Transport. Syst.*, vol. 15, no. 2, pp. 854–865, Apr. 2014. DOI: 10.1109/TITS.2013.2291493.
- [126] P. Ioannou and C. Chien, "Autonomous intelligent cruise control," *IEEE Transactions on Vehicular Technology*, vol. 42, no. 4, pp. 657–672, Nov. 1993. DOI: 10.1109/25.260745.

- [127] R. van Hoek, J. Ploeg, and H. Nijmeijer, "Motion Planning for Automated Connected Vehicles," in *Proceedings of the 14th International Symposium on Advanced Vehicle Control (AVEC'18)*, May 2018.
- [128] R. Goldenthal and M. Bercovier, "Spline Curve Approximation and Design by Optimal Control Over the Knots," in *Geometric Modelling*, S. Hahmann, G. Brunnnett, G. Farin, and R. Goldman, Eds., Vienna: Springer, 2004, pp. 53–64, ISBN: 978-3-7091-0587-0.
- [129] S. E. Li, Y. Zheng, K. Li, Y. Wu, J. K. Hedrick, F. Gao, and H. Zhang, "Dynamical Modeling and Distributed Control of Connected and Automated Vehicles: Challenges and Opportunities," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 46–58, 2017. DOI: 10.1109/MITS.2017.2709781.
- [130] H. Xing, J. Ploeg, and H. Nijmeijer, "Compensation of Communication Delays in a Cooperative ACC System," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1177–1189, Feb. 2020. DOI: 10.1109/TVT.2019.2960114.
- [131] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative Adaptive Cruise Control in Real Traffic Situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, Feb. 2014. DOI: 10.1109/TITS.2013.2278494.
- [132] W. Scholte J., P. Zegelaar W.A., and H. Nijmeijer, "Gap Opening Controller Design to Accommodate Merges in Cooperative Autonomous Platoons," in *Proceedings of the 21th IFAC World Congress*, Dec. 2019.
- [133] E. Semsar-Kazerooni, J. Verhaegh, J. Ploeg, and M. Alirezaei, "Cooperative adaptive cruise control: An artificial potential field approach," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2016, pp. 361–367. DOI: 10.1109/IVS.2016.7535411.
- [134] E. Semsar-Kazerooni, K. Elferink, J. Ploeg, and H. Nijmeijer, "Multi-objective platoon maneuvering using artificial potential fields," *IFAC-PapersOnLine*, 20th IFAC World Congress, vol. 50, no. 1, pp. 15 006–15 011, Jul. 1, 2017. DOI: 10.1016/j.ifacol.2017.08.2570.
- [135] W. L. Jansen, "Lateral Path-Following Control of Automated Vehicle Platoons.," MSc. Thesis, Delft University of Technology, Delft, 2016.
- [136] A. Bayuwindra, "Look-ahead tracking controllers for integrated longitudinal and lateral control of vehicle platoons," PhD. Dissertation, Eindhoven University of Technology, Department of Mechanical Engineering, Eindhoven, Oct. 30, 2019, ISBN: 978-90-386-4866-8.
- [137] E. Lefeber, R. van Hoek, J. Ploeg, and H. Nijmeijer, "String Stability for cascaded systems with trajectory planning," *In preparation*, 2021.
- [138] R. Bellman, "The theory of dynamic programming," *Bull. Amer. Math. Soc.*, vol. 60, no. 6, pp. 503–515, Nov. 1954. [Online]. Available: <https://projecteuclid.org/euclid.bams/1183519147> (visited on 08/27/2019).
- [139] C.-T. Chen, *Linear System Theory and Design*, 3rd. New York Oxford: Oxford University Press, 1999, ISBN: 0-19-511777-8.

- [140] H. Xing, "Delay-aware analysis and design of cooperative adaptive cruise controllers," PhD. Dissertation, Eindhoven University of Technology, Department of Mechanical Engineering, Eindhoven, Dec. 18, 2019, ISBN: 978-90-386-4906-1.
- [141] S. Baaij, "Development and validation of a multibody model of a Renault Twizy," MSc. Thesis, Eindhoven University of Technology, Department of Mechanical Engineering, Eindhoven, May 2019.
- [142] A. Severinson, "Rendits Vehicle-to-Anything Router Whitepaper," Apr. 2017. [Online]. Available: <http://rendits.com/> (visited on 03/14/2020).
- [143] E. T. S. Institute, *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, 2011. [Online]. Available: [https://www.etsi.org/deliver/etsi\\_ts/102600\\_102699/10263702/01\\_02\\_01\\_60/ts\\_10263702v010201p.pdf](https://www.etsi.org/deliver/etsi_ts/102600_102699/10263702/01_02_01_60/ts_10263702v010201p.pdf) (visited on 01/27/2021).
- [144] European Telecommunications Standards Institute, *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service*, 2014. [Online]. Available: [https://www.etsi.org/deliver/etsi\\_en/302600\\_302699/30263703/01\\_02\\_01\\_30/en\\_30263703v010201v.pdf](https://www.etsi.org/deliver/etsi_en/302600_302699/30263703/01_02_01_30/en_30263703v010201v.pdf) (visited on 01/27/2021).
- [145] Bosch Motorsport, *Acceleration Sensor MM5.10*, 2020. [Online]. Available: <https://www.bosch-motorsport.com/content/downloads/Raceparts/en-GB/51546379119226251.html#&Tabs=51582091/> (visited on 01/26/2021).
- [146] u-blox, *EVK-M8T UserGuide*, 2018. [Online]. Available: [https://www.u-blox.com/sites/default/files/products/documents/EVK-M8T\\_UserGuide\\_%28UBX-14041540%29.pdf](https://www.u-blox.com/sites/default/files/products/documents/EVK-M8T_UserGuide_%28UBX-14041540%29.pdf) (visited on 01/26/2021).
- [147] Bosch Automotive Aftermarket, *Bosch sensors*, 2014. [Online]. Available: <http://www.scribd.com/document/294268552/49776-Sensoren> (visited on 02/24/2021).
- [148] Bosch Mobility Solutions, *Product data sheet mid range radar sensor MRR 2*, 2015. [Online]. Available: [https://www.bosch-mobility-solutions.com/media/global/products-and-services/passenger-cars-and-light-commercial-vehicles/driver-assistance-systems/predictive-emergency-braking-system/mid-range-radar-sensor-\(mrr\)/product-data-sheet-mid-range-radar-sensor-\(mrr\)-2.pdf](https://www.bosch-mobility-solutions.com/media/global/products-and-services/passenger-cars-and-light-commercial-vehicles/driver-assistance-systems/predictive-emergency-braking-system/mid-range-radar-sensor-(mrr)/product-data-sheet-mid-range-radar-sensor-(mrr)-2.pdf) (visited on 01/26/2021).
- [149] J. van de Sluis, O. Baijer, L. Chen, H. H. Bengtsson, L. Garcia-Sol, and P. Balaguer, *DEL\_i-GAME\_D3.1 Wireless communication basic specification document*, 2015. [Online]. Available: <https://cordis.europa.eu/docs/projects/cnect/5/612035/080/deliverables/001-D32.pdf> (visited on 02/24/2021).
- [150] K. Hendriks, "Collision avoidance in cooperative trajectory planning via time scaling," MSc. Thesis, Eindhoven University of Technology, Department of Mechanical Engineering, Eindhoven, May 18, 2020.
- [151] R. van Hoek, J. Ploeg, and H. Nijmeijer, "Design and experimental evaluation of trajectory planning for cooperative and automated vehicles," *In preparation*, 2021.



- [152] R. van Hoek, M. Alirezaei, A. Schmeitz, and H. Nijmeijer, "Vehicle State Estimation Using a State Dependent Riccati Equation," *IFAC-PapersOnLine*, 20th IFAC World Congress, vol. 50, no. 1, pp. 3388–3393, Jul. 1, 2017. DOI: 10.1016/j.ifacol.2017.08.590.

# Dankwoord

Met een stevige portie doorzettingsvermogen is het dan na ruim vier jaar eindelijk zo ver en mag ik binnenkort mijn proefschrift verdedigen. Met het doorlopen van een promotietraject laat je als onderzoeker zien dat je in staat bent zelfstandig onderzoek te doen. Dit betekent echter niet dat je het alleen doet. Graag wil ik de mensen die betrokken waren bij dit proces bedanken.

Op de eerste plaats wil ik mijn promotoren Henk en Jeroen bedanken voor jullie onmisbare begeleiding. Henk, ik wil je hartelijk bedanken voor de mogelijkheid om binnen de Dynamics & Control groep mijn promotie onderzoek te mogen doen. Je input is zowel op wetenschappelijk als persoonlijk vlak zeer belangrijk geweest. Jeroen, ook jouw bijdrage is gewaardeerd. Ik ben je ontzettend dankbaar voor het zeer zorgvuldig voorzien van commentaar van publicaties en mijn proefschrift. Daarnaast zijn je bemoedigende woorden erg gewaardeerd, en deze hebben me mede geholpen om vol te houden bij tegenslagen.

I would like to thank professor Peter Zegelaar, professor Bayu Jayawardhana, Roland Toth and Gabor Orosz for taking part in my PhD committee and providing me with constructive feedback on my thesis.

Daarnaast wil ik een aantal personen bedanken die inhoudelijk betrokken zijn geweest bij mijn promotie. Erjen, bedankt voor je input en de discussies die we hebben gevoerd. Ik waardeer het geduld en de tijd die je nam om me wiskundige concepten uit te leggen. Mede daardoor is mijn enthousiasme voor het meer fundamentele stuk van het onderzoek aangewakkerd. Daarnaast mijn i-CAVE collega's, Wouter, Wouter en Tom, bedankt voor jullie samenwerking en assistentie bij de experimenten. Wietze, Erwin en Gerard, bedankt voor jullie werk omtrent de Twizy's. Karsten, Siemen en Daniel bedankt voor jullie bijdrage in de vorm van jullie afstudeerproject en stage.

In het bijzonder wil ik 'collega' Frans bedanken. Naast onze inhoudelijke samenwerking (en je onmisbare werk aan de Twizies), heb je een bijzondere rol gehad in mijn promotietraject. Ik kon bij jou altijd terecht voor hulp en overleg. Daarnaast hebben we ook de nodige ontspanning gehad samen, vaak met een sportieve ondertoon. In de basis gevormd door de wekelijkse bodypump, gevolgd door (topatleten als we zijn) een voedzame consumptie in de

sportkantine. Dit vormde de voorbereiding op de jaarlijkse meerkamp, waar we bloedfanatiek waren. Ook heb ik ontzettend genoten van onze wintersport vakantie waar de uitdrukkingen ‘Gang is alles’ en ‘Vaak ben je te bang’ vaak gehoord werden tijdens de *eurocarves* en het *pole-planting* en natuurlijk iedere dag afstellen.

I would also like to thank my office mates, Haitao, Bayu, Alejandro and Andy, as well as our next door neighbours, Isaac and Veronika. I’ve enjoyed the international flavour you all have given my Phd. I enjoyed the small talk we’ve had in the office and during our BBQ’s, hot-pot and daily lunches and sport activities. Daarnaast wil ik alle (oud)-collegas van D&C en CST bedanken voor de prettige werksfeer, koffiepauzes en LAN-parties. Geertje, bedankt voor de ondersteuning gedurende de promotie, je interesse, en het organiseren van de jaarlijkse uitjes.

Daarnaast wil ik mijn familie en vrienden bedanken voor de steun en de nodige afleiding. Papa en Mama, bedankt voor jullie onvoorwaardelijke steun, zowel gedurende mijn promotietraject als de rest van mijn leven. De basis voor mijn interesse in techniek werd al vroeg gelegd, waarin jullie mij altijd van harte hebben aangemoedigd.

Ten slotte, de persoon zonder wie mijn promotie niet mogelijk geweest zou zijn. Tanja, bedankt voor de liefde die je me hebt gegeven, en de steun die je me bood tijdens de pieken en dalen van het de afgelopen periode. Je hebt me verschillende keren weer terug in het zadel geholpen, waardoor we nu deze promotie met succes af kunnen sluiten en ons kunnen richten op het volgende avontuur.

Robbin van Hoek  
Eindhoven, Maart 2021

# Curriculum Vitae

Robbin van Hoek was born on June 26, 1991, in Gilze, the Netherlands. After finishing his secondary education in 2009 at Theresia Lyceum in Tilburg, the Netherlands, he studied Mechanical Engineering at the Eindhoven University of Technology (TU/e), the Netherlands, where he received his Bachelor of Science degree in 2012 (with great appreciation). In 2016, he received his Master of Science degree in Mechanical Engineering (Cum Laude). As part of his Masters degree, he carried out an internship in 2015 at Audi Sport in Neuburg an der Donau, Germany, where he worked on aerodynamic design targets for the Audi R18 e-tron quattro LMP1 hybrid race car. His graduation internship was carried out at the Integrated vehicle safety group of TNO in Helmond, the Netherlands, where he worked on vehicle state estimation for ADAS systems.



In November 2016, he started his Ph.D. research in the Dynamics and Control group at the department of Mechanical Engineering at TU/e under supervision of Henk Nijmeijer and Jeroen Ploeg. His work was part of the Integrated Cooperative Automated VEHicles (i-CAVE) project. The main results of his research are printed in this dissertation.

## List of Publications

Robbin van Hoek has the following publications:

### Journal articles

- R. van Hoek, J. Ploeg, and H. Nijmeijer, "Design and experimental evaluation of trajectory planning for cooperative and automated vehicles," *In preparation*, 2021
- E. Lefeber, R. van Hoek, J. Ploeg, *et al.*, "String Stability for cascaded systems with trajectory planning," *In preparation*, 2021
- R. van Hoek, J. Ploeg, and H. Nijmeijer, "Cooperative Driving in Automated Vehicles using B-splines for Trajectory Planning," *IEEE Transactions on Intelligent Vehicles (Accepted)*, 2021. DOI: 10.1109/TIV.2021.3072679

### Proceedings and Congress Contributions

- R. van Hoek, J. Ploeg, and H. Nijmeijer, "Gap Closing for Cooperative Driving in Automated Vehicles using B-splines for Trajectory Planning," in *The 31st IEEE Intelligent Vehicles Symposium (IV2020)*, Jan. 2020. DOI: 10.1109/IV47402.2020.9304732
- R. van Hoek, J. Ploeg, and H. Nijmeijer, "Motion Planning for Automated Connected Vehicles," in *Proceedings of the 14th International Symposium on Advanced Vehicle Control (AVEC'18)*, May 2018
- R. van Hoek, M. Alirezaei, A. Schmeitz, *et al.*, "Vehicle State Estimation Using a State Dependent Riccati Equation," *IFAC-PapersOnLine*, 20th IFAC World Congress, vol. 50, no. 1, pp. 3388–3393, Jul. 1, 2017. DOI: 10.1016/j.ifacol.2017.08.590