

Dipl.-Ing. Astrid Rupp, BSc

Trajectory Planning and Formation Control for Automated Driving on Highways

Doctoral Thesis

to achieve the university degree

Doktorin der Technischen Wissenschaften

submitted to

Graz University of Technology

Supervisor:

Univ.-Prof. Dipl.-Ing. Dr.techn. Martin Horn

Institute of Automation and Control

Faculty of Electrical and Information Engineering

Graz, May 2018

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, _____
Date

Signature

Eidesstattliche Erklärung¹

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am _____
Datum

Unterschrift

¹ Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

Abstract

This thesis deals with automated driving scenarios on highways and is divided into three parts: First, a trajectory planning algorithm for automated driving on highways is presented. A polynomial approach is used to generate a set of trajectories, which is then evaluated by a cost function considering safety and comfort. In order to achieve collision-free driving, other traffic participants are predicted. Simulation results using MATLAB/Simulink and IPG CarMaker show that the performance of this planning level is well suited for basic maneuvers on highways.

However, in dense traffic scenarios, the desired distance to other vehicles is violated by human drivers, and merging scenarios with small inter-vehicle distances are difficult to handle for autonomous cars. Hence, the second part focuses on robust velocity control of a string of vehicles with small inter-vehicle distances. The effect on other traffic participants is investigated via string stability analysis. A platooning algorithm is presented that is capable of reducing large initial spacing errors in order to maintain a small constant distance eventually. This algorithm is then extended to a merging scenario in lane reduction scenarios. Simulation studies with MATLAB/Simulink and SUMO demonstrate that this novel merging assist algorithm is capable of efficient and safe maneuvers in platooning and lane reduction scenarios.

In the third part, the proposed algorithms have been implemented on small-scale vehicles in order to investigate real-time capability and performance. For this purpose, a low-cost testbed that has been built up is presented, which allows fast and safe testing of automated driving functions. In comparison to common approaches in literature, the presented algorithms are fast and easy to tune in the experiments.

Kurzfassung

Diese Arbeit befasst sich mit automatisiertem Fahren auf Autobahnen und ist in drei Teile unterteilt: Zunächst wird eine Trajektorienplanung für klassische Szenarien auf Autobahnen vorgestellt. Eine Trajektorienmenge wird basierend auf Polynomen generiert, die dann in Bezug auf Sicherheit und Komfort des Fahrers evaluiert wird. Um Kollisionen zu vermeiden, werden die anderen Verkehrsteilnehmer prädiziert. Simulationen mit MATLAB/Simulink und IPG CarMaker zeigen, dass sich diese Trajektorienplanung sehr gut für Autobahnfahrten eignet.

Bei hohem Verkehrsaufkommen werden die Sicherheitsabstände von menschlichen Fahrern allerdings oft unterschritten und das Einordnen auf eine bestimmte Fahrspur kann zu Problemen führen. Daher wird im zweiten Teil der Arbeit eine Geschwindigkeitsregelung vorgestellt, die auch mit kleinen Sicherheitsabständen und vielen Verkehrsteilnehmern funktioniert, wobei der Effekt auf andere Verkehrsteilnehmer durch die sogenannte String-Stabilität untersucht wird. Ein neuer Algorithmus für das Fahren in einem Platoon wird vorgestellt, der auch für große Anfangsabweichungen ein string-stabiles Verhalten garantiert, und schlussendlich zu einem geringen konstanten Abstand zum Vorderfahrzeug führt. Dieser Algorithmus wird so erweitert, dass das Einordnen bei Fahrspurreduktionen effektiv gelöst werden kann, was mit MATLAB/Simulink und SUMO in Simulationen demonstriert wird.

Die Algorithmen wurden auf einem eigens entwickelten Prüfstand getestet, welcher der Fokus des dritten Teils der Arbeit ist. Modellfahrzeuge werden verwendet, um sicheres und kostengünstiges Testen von unterschiedlichsten Algorithmen zu ermöglichen. Diese Tests zeigen, dass die vorgestellten Algorithmen gute Ergebnisse erzielen, wenig Rechenaufwand benötigen und ihre Parameter leicht einzustellen sind.

Danksagung

Finanzielle Unterstützung

Diese Arbeit entstand im Rahmen des Projektes "TECAHAD" am VIRTUAL VEHICLE Research Center in Graz, Österreich. Ich bedanke mich für die Förderung im Rahmen des COMET K2 - Competence Centers for Excellent Technologies Programms des Österreichischen Bundesministeriums für Verkehr, Innovation und Technologie (bmvit), des Österreichischen Bundesministeriums für Wissenschaft, Forschung und Wirtschaft (bmwfw), der Österreichischen Forschungsförderungsgesellschaft mbH (FFG), des Landes Steiermark sowie der Steirischen Wirtschaftsförderung (SFG). Ebenfalls danke ich den unterstützenden Firmen und Projektpartnern AVL LIST GmbH und MAGNA STEYR Engineering AG & Co KG.

Ich möchte mich für die langjährige Unterstützung bei Projektleiter Dr. Michael Stolz und den Projektmitarbeitern Pamela Innerwinkler und Georg Nestlinger bedanken.

Fachliche Unterstützung

Am Anfang war die Herausforderung.

Daher geht mein erster Dank an Herrn Em.Univ.-Prof. Nicolaos Dourdoumas, dessen Begeisterung für die Regelungstechnik ansteckend ist. An dieser Stelle möchte ich auch Herrn Ao.Univ.-Prof. Anton Hofer nennen, der durch seine Erfahrungen sowohl in der Regelungstechnik als auch in den Bergen brilliert.

Mein größter Dank gilt Herrn Univ.-Prof. Martin Horn, der mich durch seine großartige Lehre zu der Entscheidung brachte, nach dem Masterstudium noch mehr von der Regelungstechnik wissen zu wollen. Vielen Dank für die langjährige Betreuung und Unterstützung mit dem richtigen Maß an Flexibilität, Unterstützung und Möglichkeiten, eigene Kompetenzen weiterzuentwickeln - und mich den Herausforderungen zu stellen.

Außerdem möchte ich mich bei Ass.Prof. Markus Reichhartinger bedanken, der mich auf die Reise ins Land der Sliding Mode Regler für Multi-Agenten-Systeme geschickt hat, deren Ergebnisse sich auch in dieser Arbeit wiederfinden. Ein großer Dank gilt auch Ass.Prof. Martin Steinberger, der mich durch hartnäckige Diskussionen zu den größten Erfolgen führte, und der mir bei der Betreuung von Studierenden zur Seite steht.

Ich möchte mich außerdem bei den Kollegen am Institut für Regelungs- und Automatisierungstechnik bedanken: bei Markus Tranninger, für äußerst beruhigende Gespräche über auf beiden Seiten von Bäumen begrenzte Wege, und für die Hilfe beim Aufbau

der Testumgebung; außerdem bei Raffael Wallner, der bei dieser Testumgebung sehr große Verbesserungen durchgeführt hat; bei Dr. Richard Seeber, für seine Bereitschaft, sein gewaltiges Wissen mit seinen Kollegen und Kolleginnen zu teilen (zum Teil über "daily lectures" auf monatlicher Basis); bei allen anderen Kollegen, für unzählige Kaffeepausen und das entspannte, freundliche Arbeitsklima.

Emotionale Unterstützung

Diese Arbeit wäre wohl nie ohne die emotionale Unterstützung der wichtigsten Personen in meinem Umfeld entstanden. Was als fachliche Herausforderung begann, wurde bald zur Prüfung der emotionalen Ausdauer und Flexibilität. Ich möchte mich vor allem bei Teresa und Karin bedanken, die die Arbeit als Frau in der Technik nicht so einsam wirken lassen. An dieser Stelle möchte ich auch das Karriereprogramm der Universitäten Graz nennen: einerseits möchte ich mich bei Martin Horn für das Einverständnis zur Teilnahme bedanken; andererseits bei allen Personen, die am Karriereprogramm mitwirken - die Unterstützung in dieser Weiterbildung war Gold wert, und der Austausch mit den Teilnehmerinnen war sehr wichtig für mich.

Meinen Bergkameraden, Wanderwütigen und kurzweiligen Hüttenmitbewohnern gebührt auch ein großer Dank. Dass das Bergsteigen sehr einem Dissertationsvorhaben gleicht, habe ich erst mit der Zeit bewusst wahrgenommen; die Erlebnisse, Erfahrungen und Lektionen haben mir zunächst den Ausgleich, später auch das Durchhaltevermögen und die Geduld verschafft.

Des weiteren geht ein grüner Dank an meinen Bruder Karli, dessen Rat mir immer sehr wichtig war, und an meine Familie, die mich mein Leben lang meinen eigenen Träumen nachgehen ließ - auch, wenn diese vielleicht nicht immer sehr einleuchtend erschienen (Stichwort: Aisha).

Schlussendlich geht mein herzlichster Dank an Matthias, der mir immer zur Seite steht und der Einheiten mit einer Hingabe korrigiert, wie es nur ein Physiker kann; und Maddox, der meine Arbeit nicht gefressen hat und der mich einiges über Lernen lernen lässt.

Am Ende war die Dissertation - und doch viel, viel mehr.

Contents

Abstract	v
Kurzfassung	vii
I Automated Driving on Highways	1
1 Introduction	3
1.1 Literature Review on Automated Driving	4
1.1.1 Advanced Driver Assistance Systems	4
1.1.2 Motion Planning	5
1.1.3 Tracking Controllers	6
1.2 Contribution of the Thesis	6
2 Trajectory Planning for Automated Highway Driving	9
2.1 Motion Planning	9
2.2 Trajectory Planning	11
2.2.1 Transformation to Road Coordinates	11
2.2.2 Trajectory Set Generation	13
2.2.3 Prediction of Obstacles	26
2.2.4 Evaluation	28
2.2.5 Reference Computation	39
2.3 Trajectory Tracking	39
2.3.1 Lateral Control	40
2.3.2 Longitudinal Control	41
2.3.3 Choice of Sampling Times	41
3 Simulation Studies and Results	45
3.1 MATLAB/Simulink	45
3.2 IPG CarMaker	51
3.2.1 Motorway Assist	52
3.2.2 Motorway Assist Plus	53
3.2.3 Motorway Chauffeur	54
3.3 SUMO	55

II	Cooperative Automated Driving	59
4	Literature Review on Cooperative Automated Driving	61
4.1	Consensus Control	62
4.2	Formation Control of Robots	62
4.3	Platooning	63
4.4	Cooperative Merging	65
5	String Stability with Linear Controllers	69
5.1	Constant Distance Spacing	71
5.2	Constant Time-Headway Spacing	74
5.3	String Stability	76
5.3.1	Constant Distance Spacing	79
5.3.2	Constant Time-Headway Spacing	80
5.3.3	Non-Autonomous Control	82
5.4	Consensus Formulation	83
6	String Stability in the Context of Sliding Mode Control	91
6.1	Problem Statement	93
6.1.1	Summary of Design Goals	95
6.2	Constant Distance Sliding Variable	95
6.2.1	Constant Distance Sliding Variable with Relative Degree Two	95
6.2.2	Constant Distance Sliding Variable with Relative Degree One	98
6.3	Constant Time-Headway Sliding Variable	106
6.4	Adaptive Time-Headway Sliding Variable	120
6.4.1	Adaptive Time-Headway Formation Controller - ATFC	121
6.4.2	Simulation Results	126
7	Formation Control in Highway Driving	131
7.1	Topology Selection	133
7.2	Reaching Phase	137
7.2.1	Reaching Phase with Constant Distance Spacing	137
7.2.2	Extension of the ATFC for Negative Initial Spacing Errors	150
7.3	Robustness with Respect to Curvature	157
7.4	Actuator Dynamics	163
7.4.1	Constant Distance Spacing	163
7.4.2	Constant Time-Headway Spacing	164
7.4.3	Adaptive Time-Headway Spacing - ATFC	165
III	Application to Small-Scale Vehicles	173
8	Testbed for Automated Driving using Small-Scale Vehicles	175
8.1	Hardware	176
8.1.1	Trucks	176
8.1.2	Cars	176
8.1.3	Sensors	177
8.1.4	Actuation	178

8.1.5	Projection of Virtual Components	178
8.2	Position Tracking	178
8.3	Software	180
8.3.1	Real-Time Operating System	181
8.3.2	MATLAB/Simulink	181
8.4	Conducting an Experiment	184
9	Experimental Results	185
9.1	Lane Keeping with Constant Velocity	185
9.2	Platooning	186
9.3	Lane Changing	191
9.4	Alternative Reference Generation Techniques	193
9.4.1	Fifth order polynomials for Intersection Maneuvers	193
9.4.2	Model Predictive Control	193
9.5	Trajectory Planning Level	195
9.6	Adaptive Time-Headway Formation Controller	199
10	Conclusion and Outlook	205
A	Consensus Reaching with Sliding Mode Control	209
B	Linear Time-Varying Systems	213
	Bibliography	215

List of Figures

1.1	SAE Levels	3
2.1	Architecture of the proposed Motorway Chauffeur	9
2.2	Planning level of the proposed Motorway Chauffeur	10
2.3	Trajectory planning level in the closed-loop system	12
2.4	Different coordinate systems	12
2.5	Polynomials for lateral deviation	15
2.6	Polynomials for lateral velocity	16
2.7	Polynomials for lateral acceleration	16
2.8	Example for a polynomial to one end point that exceeds the nominal final time (circle).	18
2.9	Generated polynomials for longitudinal velocity	20
2.10	Generated polynomials for longitudinal acceleration	20
2.11	Set of generated trajectories to different end points for one cycle	22
2.12	Set of generated trajectories to different end points with two cycles	22
2.13	Overtaking example with two planning cycles	23
2.14	Two-cycle and three-cycle planning	24
2.15	Planning with a high sample rate	25
2.16	Prediction of an obstacle using a prediction observer.	27
2.17	Ellipses with different axes	29
2.18	Example for the different levels used as costs for obstacle avoidance	30
2.19	Switching functions for the ellipses	31
2.20	Switching function for different parameters β	31
2.21	Ellipses in different situations on the road	32
2.22	Cost component for the desired lane	34
2.23	Cost component for the lane center	35
2.24	Cost component for the desired velocity	36
2.25	Longitudinal acceleration for two different velocity profiles with the same endpoint	37
2.26	Example with one slow obstacle in front and two generated trajectories for the ego vehicle	39
2.27	Costs of the two trajectories	40
2.28	Cost components of the two trajectories from Figure 2.26 at $t = 2s$	40
2.29	Steering Torque using different sampling times for planning and tracking level.	42
2.30	Lateral Offset extrapolated to deal with different sampling times.	42
2.31	Trajectory tracking	43
2.32	Approaches to deal with different sampling times	43

3.1	Results of an overtaking maneuver	46
3.2	Overtaking maneuver	47
3.3	Collision avoidance with three-cycle planning	48
3.4	Merging maneuver without adaptation of the safety distance	49
3.5	Merging maneuver with adapted ellipses	49
3.6	Numerical error of the lateral deviation	50
3.7	Numerical error of the lateral velocity	50
3.8	Overtaking scenario simulated with IPG CarMaker	51
3.9	Schematic of the closed-loop system for simulations with MATLAB and IPG CarMaker	52
3.10	Trajectory planning replacing ACC and LKA functionalities	53
3.11	Additional ellipses are used for a lane change recommendation	54
3.12	Tracking performance of the vehicle in a lane change maneuver using the MWC	54
3.13	Lateral acceleration of the vehicle in a lane change maneuver	55
3.14	Coupling between MATLAB and SUMO	56
3.15	Generated trajectories in MATLAB at t_1	56
3.16	MWC tests in SUMO at t_1	57
3.17	Generated trajectories in MATLAB at t_2	57
3.18	MWC tests in SUMO at t_2	57
3.19	Generated trajectories in MATLAB at t_3	57
3.20	MWC tests in SUMO at t_3	57
3.21	MWC tests in SUMO after t_3	58
3.22	Generated trajectories in MATLAB after t_4	58
3.23	MWC tests in SUMO at t_4	58
3.24	MWC tests in SUMO between t_4 and t_5	58
3.25	Generated trajectories in MATLAB at t_5	58
3.26	MWC tests in SUMO at t_5	58
5.1	Example for Stop-and-Go traffic	70
5.2	Results using a PD controller and a constant distance spacing	72
5.3	Results using a PD controller with different parameters and a constant distance spacing	72
5.4	Results using a PD controller and a constant distance spacing with a different frequency of the leader's acceleration	73
5.5	Closed-loop system with the constant distance policy of one agent i	73
5.6	Magnitude of the closed-loop transfer function with the constant distance policy and different controller parameters	74
5.7	Results using a PD controller with a constant time-headway policy	75
5.8	Closed loop system of one agent with constant time-headway spacing	76
5.9	Magnitude of the closed-loop transfer function for different time-headways	77
5.10	The zero of the transfer function with constant distance spacing is to the right of the rightmost pole.	80
5.11	Results using a PD controller with feed-forward and a constant distance spacing	84
5.12	Results using a PD controller with feed-forward and a constant time-headway spacing	85

List of Figures

5.13	Results using the consensus formulation	87
5.14	Results using the consensus formulation with large initial spacing errors	88
5.15	Results with imperfect communication	90
6.1	Results using the twisting algorithm with constant distance spacing	97
6.2	Phase plane of the sliding variables using the twisting algorithm	97
6.3	Results using the SOC with constant distance spacing and feed-forward	99
6.4	Phase plane of the sliding variables using the SOC with monotonic convergences	99
6.5	Results using the SOC with constant distance sliding surface of relative degree one and feed-forward	101
6.6	Phase plane of the sliding variable using the SOC with feed-forward	101
6.7	Results using the FOSMC with constant distance sliding surface of relative degree one	103
6.8	Phase planes using the FOSMC with constant distance sliding surface of relative degree one	103
6.9	Results using the FOSMC with constant distance sliding surface of relative degree one with feed-forward	104
6.10	Phase plane using the FOSMC with constant distance sliding surface of relative degree one with feed-forward	104
6.11	Results using the FOSMC with the constant distance sliding surface of relative degree one and delayed feed-forward	105
6.12	Phase plane of the errors using the FOSMC with the constant distance sliding surface of relative degree one and delayed feed-forward	105
6.13	Phase plane of the errors when using a constant time-headway spacing	106
6.14	Results using the constant time-headway spacing in sliding	106
6.15	Results using the FOSMC with constant time-headway spacing	108
6.16	Phase planes using the FOSMC with constant time-headway spacing	108
6.17	Results using the STA with constant time-headway spacing	109
6.18	Phase planes using the STA with constant time-headway spacing	109
6.19	Results using the STA with constant time-headway spacing with a different time-headway	110
6.20	Phase plane of the errors using the STA with constant time-headway spacing with a different time-headway	111
6.21	Results using the SOC with constant time-headway spacing and small initial spacing errors	112
6.22	Phase plane of the sliding variables using the SOC with constant time-headway spacing and small initial spacing errors	112
6.23	Results using the SOC with constant time-headway spacing and large initial spacing errors	114
6.24	Phase plane of the sliding variables using the SOC with constant time-headway spacing and large initial spacing errors	114
6.25	Results using the SOC with constant time-headway spacing and monotonic convergence	115
6.26	Phase plane of the sliding variables using the SOC with constant time-headway spacing and monotonic convergence	115

6.27	Results using the SOC with constant time-headway spacing and monotonic convergence with actuator limits	116
6.28	Phase plane of the sliding variables using the SOC with constant time-headway spacing and monotonic convergence with actuator limits	116
6.29	Results using the SOC with constant time-headway spacing and monotonic convergence with actuator limits and anti-windup measure	117
6.30	Phase plane of the sliding variables with SOC and anti-windup measure	117
6.31	Results using the SOC with constant time-headway spacing with communication	118
6.32	Phase plane of the sliding variables using the SOC with constant time-headway spacing with communication	118
6.33	Example with different sliding surfaces using a FOSMC	119
6.34	Positions using the ATFC with a time-varying reference acceleration	127
6.35	Position errors using the ATFC	127
6.36	Phase plane of the errors using the ATFC	128
6.37	Time-headways of the ATFC	128
6.38	Bounded velocity errors using the ATFC	129
6.39	Filtered control inputs using the ATFC	129
7.1	Architecture of the merging assist on highways	131
7.2	Structure for longitudinal control of vehicle i	132
7.3	Example for topology selection	133
7.4	Example for a changing predecessor with an active predecessor selection algorithm	134
7.5	Range limit for predecessor selection	134
7.6	Phase Plane of the errors for predecessor selection	138
7.7	Phase Plane with different trajectories using the constant distance sliding surface if the ATFC is not applicable	139
7.8	Example for a merging scenario	140
7.9	Results of the reaching phase with constant distance spacing in Example 1	141
7.10	Phase plane of the errors in Example 1	141
7.11	Time-headways in Example 1	142
7.12	Results of the constant distance spacing with waiting phase in Example 2	143
7.13	Phase plane of the errors in Example 2	143
7.14	Time-headways in Example 2	144
7.15	Results of the proposed cooperative merging assist in Example 3	145
7.16	Phase plane of the errors in Example 3	145
7.17	Time-headways in Example 3	146
7.18	Sliding variable of the blue vehicle	147
7.19	SUMO results of the proposed cooperative merging assist in Example 4	148
7.20	Results of Example 5 using the constant distance sliding surface for negative position errors	149
7.21	Phase plane of the position errors using the extended ATFC	151
7.22	Phase plane of the position errors with different trajectories	152
7.23	State diagram of the proposed cooperative merging assist	152
7.24	Results of the platoon merging scenario using the proposed cooperative merging control with the ATFC extension	154

List of Figures

7.25	Results using the proposed cooperative merging control for merging with random initial errors and the ATFC extension	156
7.26	Effect of curvature on way-lengths	157
7.27	Clothoids for curvature on highways	158
7.28	Road coordinate system with corresponding velocities	159
7.29	Example for rejection of disturbance during lane change	161
7.30	Results of disturbance compensation during a lane change scenario	162
7.31	Schematic of the closed-loop system using the constant distance spacing with actuator dynamics for vehicle-following	164
7.32	Schematic of the auxiliary system for the sliding variable using the FOSMC with constant distance spacing and actuator dynamics in a platoon	164
7.33	Schematic of the closed-loop system using the constant time-headway spacing for one agent in a platoon	165
7.34	Results using the ATFC with a FOSMC in presence of actuator dynamics	167
7.35	Phase plane of the errors using the ATFC with a FOSMC in presence of actuator dynamics	168
7.36	Phase plane of the errors using the ATFC with a SOC in presence of actuator dynamics	168
7.37	Results using the ATFC with a SOC in presence of actuator dynamics	169
7.38	Results using the FOSMC in presence of slow actuators	170
7.39	Results using the FOSMC in presence of slow actuator dynamics	171
8.1	Setup of the small-scale testbed for control engineering purposes	176
8.2	Model trucks used in the testbed	177
8.3	Small-scale car used in the testbed	177
8.4	Calibration of the planar coordinate system	179
8.5	Field of view of the webcams	180
8.6	Road coordinates defined for testing in the foyer of the institute	181
8.7	Simulink block diagram for ADAS implementation	182
9.1	Results of the lane keeping experiments	186
9.2	Results of the lane keeping experiments using a small-scale car	186
9.3	Results of the platooning experiments using C2C communication and constant distance spacing	188
9.4	Inter-vehicle spacings of the platooning experiments without C2C communication	188
9.5	Results of the platooning tests with a FOSMC and the constant distance spacing	189
9.6	Results of the platooning tests with a FOSMC and the constant time-headway spacing	190
9.7	Results of the platooning tests with the SOC and a constant time-headway spacing	191
9.8	Model car performing lane changes with different velocities	192
9.9	Results of the lane changing tests	192
9.10	Results of the different reference generation and tracking experiments using fifth order polynomials as function of waylength	193

9.11	Results of an overtaking maneuver with model predictive reference computation	194
9.12	Results using the model predictive reference computation with fast rear obstacle	194
9.13	Results of the trajectory planning level in a collision avoidance scenario with two obstacles at standstill	195
9.14	Results of the trajectory planing level in a collision avoidance scenario . . .	195
9.15	Results of the trajectory planning level using the small-scale vehicles in overtaking maneuvers	196
9.16	Results of the trajectory planing level in an overtaking scenario	196
9.17	Cost components of the trajectory planning level	198
9.18	Results of one agent approaching the leader using the ATFC with a FOSMC	200
9.19	Computed velocity and control input in the experiment of the ATFC with a FOSMC	200
9.20	Computed velocity and accelerations of the ATFC using the SOC	201
9.21	Results of the ATFC using the SOC	202
A.1	Results using the twisting controller for leader-tracking consensus problems	211
A.2	Phase plane using the twisting controller for leader-tracking consensus problems	211
B.1	Example for LTI and LTV systems with different convergence	214

List of Abbreviations

ACC	Adaptive cruise control
AD	Automated driving
ADAS	Advanced driver assistance system
ABS	Anti-lock braking system
AHS	Automated Highway Systems
ATFC	Adaptive time-headway formation controller
AW	Anti-windup
BBB	BeagleBone Black
CACC	Cooperative adaptive cruise control
C2C	Car-to-Car
C2x	Car-to-everything/Car-to-Infrastructure
ESC	Electronic stability control
FF	Feed-forward
FOSMC	First order sliding mode controller
HMI	Human machine interface
LCA	Lane change assist
LDWS	Lane departure warning system
LKA	Lane keeping assist
MWA	Motorway assist
MWA+	Motorway assist plus
MWC	Motorway chauffeur
LTI	Linear time-invariant
LTV	Linear time-variant
MPC	Model predictive control
PI	Proportional-integral
PD	Proportional-derivative
RRT	Rapidly-exploring random tree
SMC	Sliding mode control
SOC	Suboptimal controller
STA	Super-twisting algorithm
TJA	Traffic jam assist
TP	Trajectory planning
V2V	Vehicle-to-Vehicle
V2x	Vehicle-to-everything/Vehicle-to-Infrastructure

Part I.

Automated Driving on Highways

1. Introduction

Autonomous driving has attracted interest of researchers for many decades, as discussed, e.g., in [141], [142]. The benefits are doubtlessly appealing: safer and more efficient driving, with the “Vision Zero” that no fatal accidents occur. Over the last years, automobile industry has invested very much in the development of autonomous vehicles. However, autonomous vehicles need a lot of additional sensors compared to the automated driving systems that are currently used in cars, e.g., Adaptive Cruise Control (ACC) to control the velocity of the car automatically, or Lane Keeping Assist (LKA) to support the driver in staying on the current lane. With this in mind, a step-by-step integration of advanced driver assistance systems (ADAS) is proposed, increasing the automation gradually. In Figure 1.1, the widely known SAE Levels [18] are displayed: In the current vehicles in the market, level 2 ADAS such as ACC or LKA are available. The driver has to monitor the environment, i.e., has to be available at all times, and bears responsibility for collisions.

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

Copyright © 2014 SAE International. The summary table may be freely copied and distributed provided SAE International and J3016 are acknowledged as the source and must be reproduced AS-IS.

Figure 1.1.: The different SAE Levels and their definition according to [18]

1. Introduction

By introducing conditional automation in level 3 and high or full automation in higher levels, the driver can be taken out of the loop for specific tasks. In a first step in level 3, the driver is allowed to be occupied by other tasks while the ADAS is responsible for standard highway driving. At first glance, this step seems to be very easy, since longitudinal and lateral control are already implemented in vehicles that are equipped with ADAS. However, the system has to make decisions for the first time, which can be simple for drivers, but very hard for machines. Thus, this step from level 2 to level 3 ADAS has been the focus of the COMET K2-project "TECAHAD" at the VIRTUAL VEHICLE Research Center in Graz, which gave rise to this work.

The first part of the thesis describes the motion planning level, i.e., the computation of a reference path and a corresponding velocity, of an automated vehicle driving individually on highways. This motion planning level corresponds to making choices based on the current situation of the vehicle. The second part of this thesis is dedicated to cooperative driving in order to make automated driving more efficient with respect to traffic flow. The third part describes a testbed at the Institute of Automation and Control, Graz University of Technology, which has been used to test the developed algorithms on small-scale vehicles.

1.1. Literature Review on Automated Driving

As already mentioned, automated and autonomous driving have been investigated for many decades. In a first approach, researchers had the vision that cars will drive along magnetic markers [160], which has later been dropped due to high infrastructural requirements. In the 1990s, the main focus has been placed on automated highway systems (AHS), e.g., [81], [116], where longitudinal and/or lateral control for automated driving on highways has been discussed, with the goal to decrease traffic congestions and increase safety. However, only functionalities that aid the driver indirectly by stabilizing the vehicle, such as the anti-lock braking system (ABS) or electronic stability control (ESC), have been mass-produced, see, e.g., [19], [161], [173]. In the early 2000s, the first functionality that can be switched on/off by the driver on demand for longitudinal guidance, namely "Adaptive Cruise Control (ACC)", has been implemented in vehicles in series, e.g. [106], [155]. Since then, different additional "comfort functionalities" for longitudinal and lateral guidance have been added and will be briefly discussed subsequently.

1.1.1. Advanced Driver Assistance Systems

The ACC functionality has been investigated in the early 1990s, see e.g., [53], [84], [166], and has successfully been implemented on vehicles in the last decade, e.g., [48], [106], [155], [168]. Communication between the cars facilitates a cooperative ACC (CACC), [121], [128], [169], which is expected to increase traffic flow and reduce collisions.

At the same time, platooning for automated highway systems (AHS) has emerged [81], [116]. In a platoon, several vehicles are aligned in a string and the velocity of each vehicle is controlled such that a small inter-vehicle distance is safely maintained to the vehicle in front; thus the step from longitudinal control of one vehicle to controlling multiple vehicles

has been made as, e.g., in [133]. Communication between the vehicles is mandatory for heavy-duty vehicle platooning since fuel efficiency can only be achieved if the inter-vehicles distances are small. This can only be achieved if some information, e.g. the leading vehicle's velocity or acceleration, is sent to all other vehicles, [72], [162], [164]. With the platooning application, the so-called "string stability" has become an important property and has been discussed, e.g., in [112], [129], [157], [174]. String stability is necessary to avoid collisions in a string of vehicles, i.e., error amplification from one vehicle to another has to be avoided.

Lateral control has been investigated in order to develop a Lane Keeping Assist (LKA) [80]. In a first step, however, only a warning system has been developed, known as lane departure warning system (LDWS), [70], [173], which has also been used in platooning applications [152]. Nowadays automated assistance system that control the vehicle directly are developed as presented, e.g., in [114], [170].

A combination of longitudinal and lateral guidance has been implemented in the Traffic Jam Assist (TJA), see, e.g. [20], where a vehicle tracks its predecessor at low velocities, in some implementations also in absence of lane markings. In the last years, the Motorway Chauffeur (MWC) as described in [113], also called "Highway Pilot" by some companies, has been developed by the automotive industry. The MWC is capable of adapting the velocities and changing lanes according to the current traffic situation autonomously on highways.

Fully automated or autonomous driving and cooperative driving are currently discussed extensively, starting from the first DARPA Challenge [163]. With the following DARPA Grand Cooperative Driving Challenges, many prototypes have been built and tested, e.g., [37], [73], [132], [178]. In contrast to the level 2 systems ACC and LKA, higher level ADAS require a motion planning system. Many different approaches exist, which are summarized in the following section.

1.1.2. Motion Planning

In [74], different motion planning algorithms from the perspective of autonomous unmanned aerial vehicle guidance have been discussed. These planning methods depend on the environment, where one can distinguish between two different parts as described, e.g., in [95], [167]. The first one is related to the structured environment, e.g., planning on roads with fixed lanewidth, detected lane markings, and a known number of lanes. The second one emerged from the field of mobile robotics and is applied in the unstructured environment, which is, e.g., offroad or in parking lots, where no lanes are defined. The main difference is that in unstructured environments, the path to the destination has to be found, avoiding obstacles and possibly considering constraints of the vehicle. Sampling-based methods such as the A^* -algorithm or the Rapidly-Exploring Random Tree (RRT) have been used to find paths from a starting point to a destination. However, on highways, the path cannot vary arbitrarily and these path finding methods are computationally too expensive. Moreover, other traffic participants have to be taken into account, and the unpredictability over long time horizons render these approaches unsuitable for highway driving. In [62] and [63], different planning approaches are proposed for these two parts, therein distinguished as off-road or on-road driving.

1. Introduction

In this work, the focus is on highway driving, i.e., driving in structured environments. Thus, the focus is placed on local on-road planning, which has been neatly summarized in [89]. Therein, the path planning methods are divided into incremental search techniques such as RRT or lattice planners, and local search techniques that define the endpoints and compute certain geometric curves. These local optimization-based techniques can be divided into polynomial planning as in [83], [107], or model predictive control (MPC) as, e.g., in [60], [115]. For details, the interested reader is referred to [89], [148], and references therein.

In this thesis, local search techniques with fifth order polynomials are used similar to [83] due to their computational efficiency. This planning approach is described in detail in Chapter 2.

1.1.3. Tracking Controllers

In autonomous driving scenarios, the references for both lateral and longitudinal controllers are computed by the planning level and have to be tracked by a low level controller. Appropriate controllers have to be used in order to guarantee that the vehicle tracks the references. The lateral and longitudinal dynamics are coupled and non-holonomic constraints are present, e.g., the vehicle cannot drive side-ways, which makes the controller design difficult. On highways, the vehicle typically does not drive at its limits and hence, the separation of controller design for longitudinal and lateral tracking is reasonable. A detailed survey on control schemes for these longitudinal and lateral assistance systems can be found in [145].

Note that the separation of planning and tracking level can lead to undesired behavior, e.g., external disturbances such as wind cannot be included in the planning level, but the tracking performance will be influenced by this disturbance. Hence, there exists literature on how to combine these two components so that planning stability can be guaranteed using MPC as discussed, e.g., in [159]. However, due to the computational complexity of MPC, this work focuses on methods that can be implemented fast and in real-time.

1.2. Contribution of the Thesis

The main contribution of the first part of this thesis is the implementation of a real-time capable trajectory planning algorithm for simulation with MATLAB/Simulink¹ and different other tools, such as IPG CarMaker² and SUMO³. A polynomial approach has been used for both lateral and longitudinal motion. An extended cost function is discussed that takes into account the surrounding vehicles similar to human drivers. Intuitive parameter tuning allows for “sporty” or comfortable maneuvers, which can increase the acceptance of passengers. In Chapter 2, this planning level with focus on trajectory generation for

¹ <http://de.mathworks.com/products/simulink.html>, accessed: 2018-04-25

² <http://ipg-automotive.com/products-services/simulation-software/carmaker/>, accessed: 2018-04-25

³ <http://sumo.dlr.de/index.html>, accessed: 2018-04-25

automated highway driving is presented. Different scenarios have been tested and show that this trajectory planning algorithm is capable of handling various scenarios. Although the planning level is designed for comfortable scenarios such as overtaking, collision avoidance with large lateral acceleration can also be performed by the proposed method. The simulation results are discussed in Chapter 3.

The second part of the thesis focuses on longitudinal control in cooperative scenarios. The goal for a single vehicle is to safely merge to the desired lane or yield to preceding vehicles on the adjacent lane, where the gap to merge has to be found depending on the current traffic situation. In order to analyze safety and efficiency of a merging maneuver, a large number of vehicles has to be considered, and the effects of one vehicle on the others has to be analyzed. These effects of the traffic flow are investigated by means of the so-called string stability, where the amplification of errors and accelerations along a string of vehicles has to be avoided, which is the focus of the second part of this thesis. In Chapter 4, a literature review on cooperative driving is presented. Platooning is investigated in detail using linear controllers in Chapter 5, or sliding mode controllers in Chapter 6, and a smart choice of the inter-vehicle spacing for safe and efficient driving is proposed. In Chapter 7, a robust decentralized merging assist is discussed for lane reduction scenarios.

A testbed using small-scale vehicles, which has been built up at the Institute of Automation and Control in order to test different algorithms for automated vehicles, is presented in the third part. In Chapter 8, the experimental setup is explained in detail, and the results of different planning and tracking algorithms are given in Chapter 9. Finally, conclusions are drawn in Chapter 10, and future work is discussed.

2. Trajectory Planning for Automated Highway Driving

The motion planning level of a highly automated ADAS such as Motorway Chauffeur consists of a global path computation and a local trajectory planning algorithm. Locally, paths and velocities are computed to avoid collisions, track a desired velocity and respect given constraints, e.g., maximum accelerations with minimum jerk, thus generating “comfortable” driving maneuvers. The architecture used for this work is described in the following section.

2.1. Motion Planning

The software architecture of a highly automated ADAS used in this work from [113] is shown in Figure 2.1. The focus in this work is the motion planning of the Motorway Chauffeur, which is also called planning level. The operational constraints include conditions when the MWC can be switched on and off, e.g., activation is only possible if the vehicle is

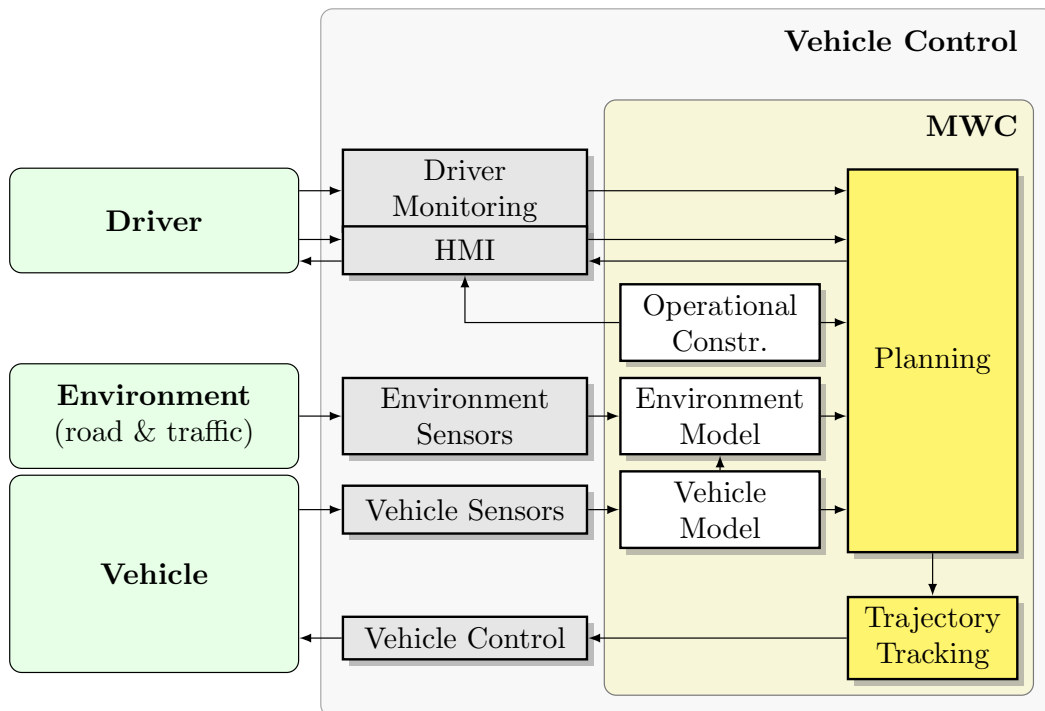


Figure 2.1.: Architecture of the proposed Motorway Chauffeur

2. Trajectory Planning for Automated Highway Driving

on a highway. Moreover, constraints arising from traffic signs, e.g., maximum velocity, or vehicle constraints are included in this block. The main “intelligence” of the MWC is the planning level, where global and local decisions are made, while other traffic participants are predicted and constraints are satisfied. For this purpose, the environment model has to be accurate, i.e., road and traffic are reasonably represented by so-called “objects” in the environment model. In this work, it is thus assumed that the vehicle sensors are capable of detecting all relevant objects and that the sensor fusion results in accurate objects, such as traffic participants, road map information, traffic regulations etc., placing the focus of this work on control engineering tasks.

The interaction of the MWC with the driver and vice versa in a highly automated system is limited: the driver can switch the system on or off, can set a desired velocity and a desired destination by using the Human Machine Interface (HMI). All other control tasks on the highway are handled by the MWC and the driver does not need to monitor the system until a take-over request by the system is triggered. Note that driver monitoring is only necessary at this take-over request, i.e., the system can be switched off if the driver has completely taken over control of the vehicle. Otherwise, it must be detected that no take-over was possible and emergency maneuvers to stop the vehicle safely need to be performed.

From a control engineer’s point of view, the yellow highlighted parts in Figure 2.1 are of special interest: first, reference trajectories have to be planned, then tracked by low-level controllers. The planning level can be further divided into three levels as shown in Figure 2.2, which is similar to the architecture used for the autonomous vehicle Bertha in [159]. First, the mission planning level is responsible for global route computation from

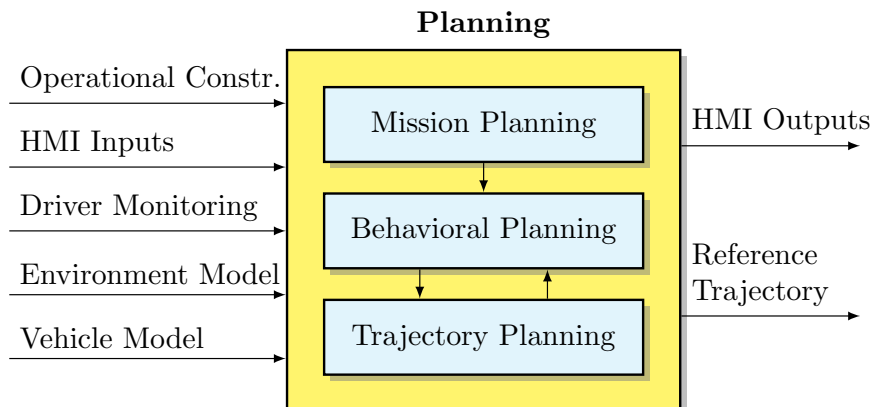


Figure 2.2.: Planning level of the proposed Motorway Chauffeur

the current point to a destination specified by the driver, and is thus also called navigation level. The second level is the behavioral planning (BP) level, also called decision unit, which is implemented as a state machine that switches different functions on or off and has to intercept errors. The trajectory planning (TP) is the third level, which computes a local reference path and velocity while taking into account other traffic participants.

As described above, the inputs of the planning level consist of the environmental data, driver inputs, operational constraints and vehicle information. The operational constraints are settings that are mainly monitored by the behavioral planning level, such as activation

criteria for a certain ADAS, maximum values for road information, maximum velocities, etc. However, limits for the states such as maximum velocities or accelerations have to be incorporated also by the trajectory planning level, and permanent interaction between the behavioral planning and trajectory planning level is necessary. The driver can activate and deactivate an ADAS and set desired values by HMI inputs, such as desired destination and velocities. This information is typically processed in the behavioral planning level, which interacts with the driver via an HMI output. Moreover, driver monitoring and take-over requests via HMI outputs are handled by the state machine of the behavioral planning level. In order to be able to react to the current traffic situation, actual states of the objects and vehicle states have to be inputs to the planning level, e.g., the actual position of vehicles on the road, or the actual velocity of the controlled car.

The mission planning and behavioral planning levels are necessary to compute additional information for the trajectory planning level, e.g., which lane has to be taken at highway intersections. Then, the main reference generation is performed within the trajectory planning level, which is described subsequently.

2.2. Trajectory Planning

The trajectory planning level computes the transition of the vehicle's states at the current time to desired final states, which is determined by incorporation of all inputs to the TP level such as driver's intention, other traffic participants' information and constraints. Typically, the output of the trajectory planning level is a reference path with a reference velocity profile for the controlled vehicle, which will be called ego vehicle furtheron.

In order to compute these references, the trajectory planning in this work can once again be divided into four parts as shown in Figure 2.3: First, a coordinate transformation from local sensor data to road coordinates is performed, which allows to plan references for a straight road. Second, the generation of the trajectory set uses different endpoints to compute possible paths that do not consider any traffic participants. In addition, the prediction of the obstacle trajectories is performed, incorporating the trajectories of the ego vehicle. Third, all trajectories are evaluated and the best with respect to safety, efficiency and comfort is chosen. In a final step, the best trajectory, which has been sampled sparsely in the evaluation procedure, has to be transformed to a high-resolution reference path and velocity on the actual road for the controllers, thus a reference computation is performed at the end of the planning level. The output of the TP is a reference vector $\mathbf{r}(t)$ for the tracking controller, which consists of a longitudinal and a lateral part. These four steps are explained in this section in detail, where special attention is paid to the evaluation, since it constitutes a large part of the decision making process of the ADAS and aims at imitating human drivers. The error computation and controller shown in Figure 2.3 will be explained in Section 2.3.

2.2.1. Transformation to Road Coordinates

The trajectories are computed for straight roads. Since this is not necessarily true on highways, road coordinates are used to transform the current road states to states corresponding

2. Trajectory Planning for Automated Highway Driving

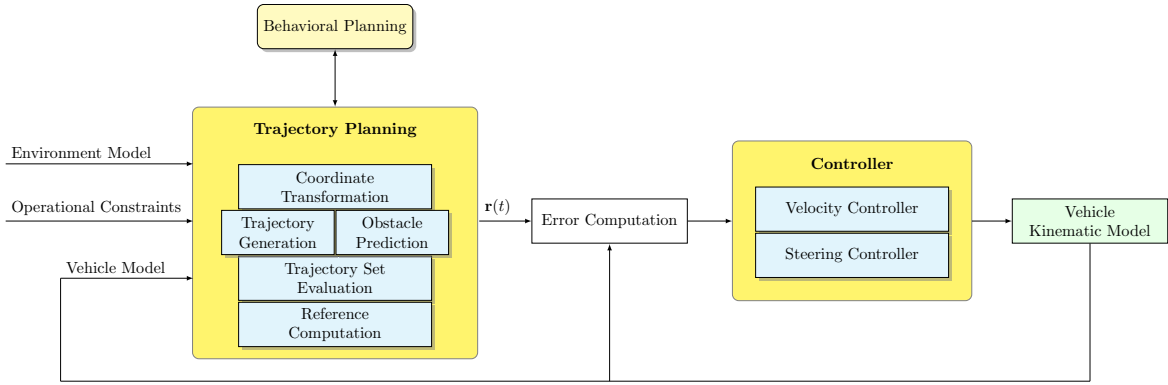


Figure 2.3.: The trajectory planning level in the closed-loop system. Interaction with the higher-level behavioral planning is necessary to obtain the desired values of the driver. The references are computed based on the actual states of the environment and the vehicle, and are tracked by the vehicle using appropriate controllers.

to a straight road with zero curvature.

In order to facilitate trajectory planning, the vehicles are assumed to be point-mass models in the road coordinate system (S, L) that moves along the road as depicted in Figure 2.4. Note that the road coordinate system differs from a global coordinate system (x, y) since it does not consider the orientation or curvature of the road. The point-mass models can be described by double-integrators, both for longitudinal and lateral motion, i.e., by the states s, \dot{s} and the input \ddot{s} for longitudinal and states l, \dot{l} and input \ddot{l} for lateral dynamics. However, the sensor data delivers distances to obstacles in the vehicle's local coordinate system $\Delta x, \Delta y$ and the orientation of the car influences these results. In order to simplify the trajectory generation of the TP, all vehicles are transformed to local road coordinates (s, l) , where the longitudinal part is relative to the ego vehicle's coordinates $(0, l_{ego})$, and the lateral component is assigned with respect to the lateral deviation on the road as shown in Figure 2.4. Note that this idea corresponds to the Frenét Frame approach as described, e.g., in [172].

For the implementation of this transformation, high-precision knowledge of the road coordinates is necessary, otherwise the assignment of other vehicles to longitudinal relative

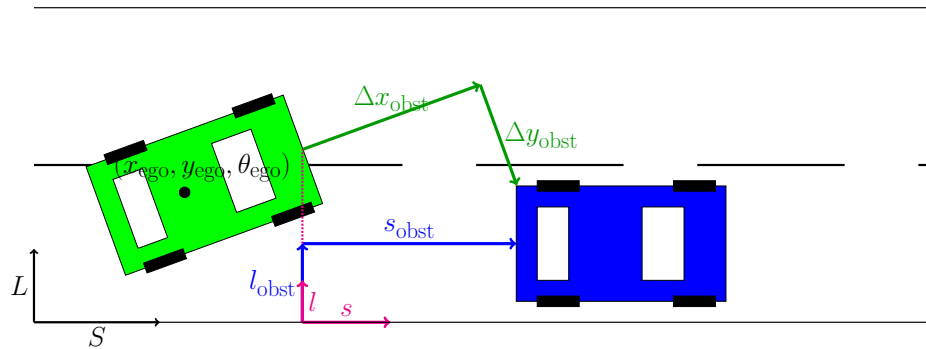


Figure 2.4.: Different coordinate systems have to be considered: the planning is executed in (s, l) space, while the sensors of the green ego vehicle deliver $(\Delta x, \Delta y)$ distances to the blue obstacle.

distances with respect to road coordinates is not accurate. In this work, the following road information in the surroundings of the ego vehicle are assumed to be provided by the environment model with appropriate accuracy:

- way-length along the road s_{road} , e.g., from a global start position
- positions in global coordinates $(x_{\text{road}}, y_{\text{road}})$,
- global orientation θ_{road} ,
- global curvature κ_{road} .

Note that the curvature κ_{road} can be used for the computation of the road coordinates $(x_{\text{road}}, y_{\text{road}}, \theta_{\text{road}})$, or the road coordinates are given by a map, i.e., matrices that store the coordinates in certain distances ds along the route. The road information in the ego vehicle's local coordinate system can either be available from sensor measurement, or can be computed from the current information $x_{\text{ego}}, y_{\text{ego}}, \theta_{\text{ego}}, s_{\text{ego}}$ of the ego vehicle: the local road coordinates $(x_{\text{road,local}}, y_{\text{road,local}})$ with respect to the ego vehicle are computed by

$$\begin{bmatrix} x_{\text{road,local}} \\ y_{\text{road,local}} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{\text{ego}}) & \sin(\theta_{\text{ego}}) \\ -\sin(\theta_{\text{ego}}) & \cos(\theta_{\text{ego}}) \end{bmatrix} \begin{bmatrix} x_{\text{road}} - x_{\text{ego}} \\ y_{\text{road}} - y_{\text{ego}} \end{bmatrix}, \quad (2.1)$$

and the local way-length $s_{\text{road,local}}$ and orientation $\theta_{\text{road,local}}$ are given by

$$\begin{aligned} s_{\text{road,local}} &= s_{\text{road}} - s_{\text{ego}}, \\ \theta_{\text{road,local}} &= \theta_{\text{road}} - \theta_{\text{ego}}. \end{aligned} \quad (2.2)$$

The local measurement of an object i by the ego vehicle represented by the local distances $(\Delta x_{\text{obst},i}, \Delta y_{\text{obst},i})$ is then assigned to the closest local road parameter $x_{\text{road,local}}, y_{\text{road,local}}$ and simple vector computations together with the way-length information $s_{\text{road,local}}$ are then used to obtain $(s_{\text{obst},i}, l_{\text{obst},i})$ of the obstacle i .

In this work, 200 m in front and 100 m rear of the vehicle based on the sensor ranges are chosen for road description in the environment model, where it is assumed that the course of the road with the parameters listed above is known with an accuracy of 0.1 m. Since the ego vehicle represents the origin of the local coordinate system, its s -coordinate is assumed to be zero, i.e., $s_{\text{ego}} = 0$.

2.2.2. Trajectory Set Generation

After the coordinate transformation, the trajectories can be generated for straight roads. There exist various possibilities to generate candidate trajectories, e.g., polynomials based on way-length, where lateral and longitudinal dynamics are combined, or polynomials based on time, where it is possible to separate lateral deviation from velocities during the generation process. The main advantage of the second approach is the low computational complexity, since the polynomial coefficients can be computed given the initial and end conditions and no optimization is necessary. The disadvantage that the longitudinal and lateral dynamics are completely decoupled, which is in reality not possible, is alleviated by the consideration on highways only: since the velocity is more important than the longitudinal position and the vehicle typically drives in a comfortable range of dynamics, i.e., no driving at the limits is performed, this disadvantage is acceptable. Hence, in this section the polynomial generation for longitudinal and lateral motion is described separately, as is also discussed in [172] for high speeds.

2. Trajectory Planning for Automated Highway Driving

2.2.2.1. Polynomial Generation for the Lateral Offset

The lateral trajectory set generation is based on [83] and briefly described in this section. The following information from the higher-level behavioral planning about the environment is necessary for lateral polynomial generation, including road parameters and constraints:

- number of available lanes n_{lane} and lanewidth w_{lane}
- desired lane (in case of a highway junction) and thus the desired lateral deviation on the road l_{des}
- ego-vehicle information: actual lateral position in road coordinates $l = l_{\text{ego}}$

Based on this information, a set of smooth trajectories from the initial states $l(t_0)$ of the point-mass model to different end points $l(t_f)$ on the available lanes is computed. Since the lateral jerk j_{lat} , which is the third derivative of the position l , is the most important indicator for comfortable maneuvers, the set of trajectories are generated such that the cost functional

$$J = \int_{t_0}^{t_f} \frac{1}{2} j_{\text{lat}}(t)^2 dt \quad (2.3)$$

for some fixed final time t_f is minimized. The optimization problem for one trajectory with $t_0 = 0$ without loss of generality can be stated as

$$\begin{aligned} \min_{j_{\text{lat}}(t)} & \int_0^{t_f} \frac{1}{2} j_{\text{lat}}(t)^2 dt, \\ \text{s.t.} & \quad \dot{l}(t) = v_{\text{lat}}(t), \\ & \quad \dot{v}_{\text{lat}}(t) = a_{\text{lat}}(t), \\ & \quad \dot{a}_{\text{lat}}(t) = j_{\text{lat}}(t), \\ \text{with} & \quad l(0) = l_0, \quad v_{\text{lat}}(0) = v_{\text{lat},0}, \quad a_{\text{lat}}(0) = a_{\text{lat},0}, \\ & \quad l(t_f) = l_f, \quad v_{\text{lat}}(t_f) = v_{\text{lat},f}, \quad a_{\text{lat}}(t_f) = a_{\text{lat},f}. \end{aligned} \quad (2.4)$$

The Hamiltonian H is then defined by

$$H = \frac{1}{2} j_{\text{lat}}^2 + \lambda_1 v_{\text{lat}} + \lambda_2 a_{\text{lat}} + \lambda_3 j_{\text{lat}}, \quad (2.5)$$

and the following equations hold

$$\begin{aligned} \dot{\lambda}_1 &= -\frac{\partial H}{\partial l} = 0, & \Rightarrow \lambda_1 &= C_1, \\ \dot{\lambda}_2 &= -\frac{\partial H}{\partial v_{\text{lat}}} = -\lambda_1, & \Rightarrow \lambda_2 &= -C_1 t + C_2, \\ \dot{\lambda}_3 &= -\frac{\partial H}{\partial a_{\text{lat}}} = -\lambda_2, & \Rightarrow \lambda_3 &= \frac{1}{2} C_1 t^2 - C_2 t + C_3, \\ 0 &= \frac{\partial H}{\partial j_{\text{lat}}} = j_{\text{lat}} + \lambda_3, & \Rightarrow j_{\text{lat}} &= -\lambda_3. \end{aligned} \quad (2.6)$$

Then the lateral states can be expressed as

$$\begin{aligned}
 j_{\text{lat}}(t) &= -\frac{1}{2}C_1t^2 + C_2t - C_3, \\
 a_{\text{lat}}(t) &= -\frac{1}{6}C_1t^3 + \frac{1}{2}C_2t^2 - C_3t + C_4, \\
 v_{\text{lat}}(t) &= -\frac{1}{24}C_1t^4 + \frac{1}{6}C_2t^3 - \frac{1}{2}C_3t^2 + C_4t + C_5, \\
 l(t) &= -\frac{1}{120}C_1t^5 + \frac{1}{24}C_2t^4 - \frac{1}{6}C_3t^3 + \frac{1}{2}C_4t^2 + C_5t + C_6.
 \end{aligned} \tag{2.7}$$

The coefficients can then be computed for a fixed final time t_f by using the initial conditions and final conditions of $l, v_{\text{lat}}, a_{\text{lat}}$ in (2.4), which yields after some mathematical manipulations

$$\begin{aligned}
 C_6 &= l_0, \quad C_5 = v_{\text{lat},0}, \quad C_4 = a_{\text{lat},0}, \\
 C_3 &= \frac{3(3a_{\text{lat},0} - a_{\text{lat},f})t_f^2 + 12(3v_{\text{lat},0} + 2v_{\text{lat},f})t_f + 60(l_0 - l_f)}{t_f^3}, \\
 C_2 &= \frac{12(3a_{\text{lat},0} - 2a_{\text{lat},f})t_f^2 + 24(8v_{\text{lat},0} + 7v_{\text{lat},f})t_f + 360(l_0 - l_f)}{t_f^4}, \\
 C_1 &= \frac{60(a_{\text{lat},0} - a_{\text{lat},f})t_f^2 + 360(v_{\text{lat},0} + v_{\text{lat},f})t_f + 720(l_0 - l_f)}{t_f^5}.
 \end{aligned} \tag{2.8}$$

The final values $v_{\text{lat},f}$ and $a_{\text{lat},f}$ are always chosen to be zero in order to generate smooth transitions between maneuvers, whereas l_f is chosen such that every lane center is included. Additional end points in the lane can be considered, as shown in Fig. 2.5 for equally spaced end points. In Figures 2.6 and 2.7, the corresponding velocities and accelerations are shown.

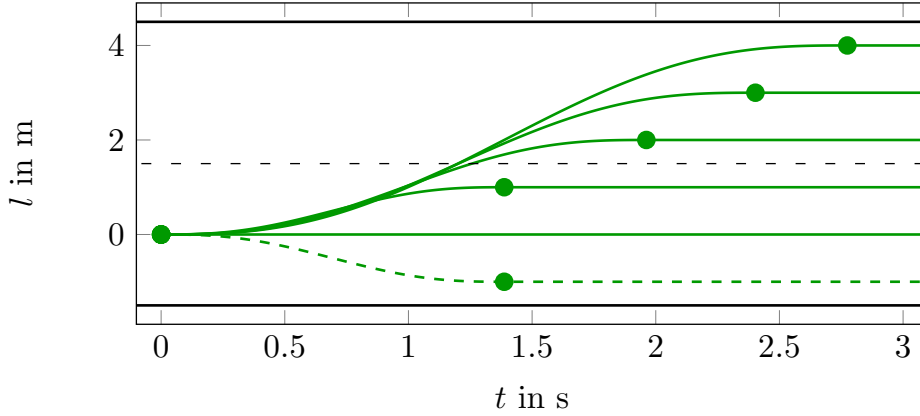


Figure 2.5.: Generated polynomials for the lateral deviation l to different end points l_f . The circles depict the nominal final time to reach the end points.

Note that the maximum acceleration has not been considered so far, and handling these constraints can be difficult. Instead of incorporating this limit into the optimization problem, the final time t_f as only remaining free parameter is adjusted in order to guarantee that

2. Trajectory Planning for Automated Highway Driving

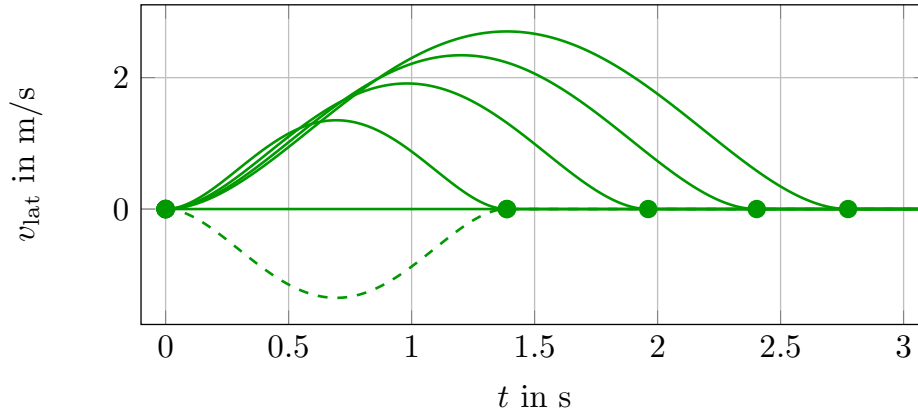


Figure 2.6.: Generated polynomials for the lateral velocity v_{lat} . The circles depict the nominal final time to reach the end points.

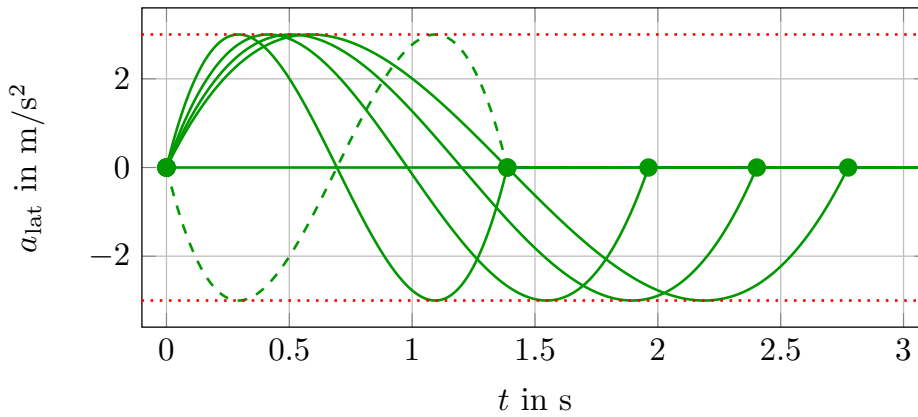


Figure 2.7.: Generated polynomials with predefined maximum acceleration a_{lat} in lateral direction. The circles depict the nominal final time to reach the end points, and the red dotted lines indicate the maximum acceleration that is defined in the operational constraints. Note that this maximum acceleration is not the physical limit of the vehicle, but a maximum acceleration that is defined for comfortable maneuvers.

the lateral acceleration stays below a certain maximum value $|a_{\text{lat}}(t)| < a_{\text{lat,max}}$ for all t and $a_{\text{lat,max}} > 0$. Hence, the same lateral acceleration is used for all trajectories, but the time to reach a certain lateral deviation varies. This procedure is reasonable, since a small lateral change within the own lane should be performed faster than, e.g., two successive lane changes. Then, the generated polynomials have different lengths as shown in Fig. 2.5. Note that after reaching the final time t_f of each trajectory as depicted by the circles, the lateral deviation is held constant and thus the derivatives are zero. Note that if the lateral deviation is held constant, the end time coincides with the initial time.

In order to compute the final time of the trajectories, a bisection method is used. For this purpose, the resulting maximum acceleration of each trajectory $a_{b,\text{max}}$ has to be computed given a fixed final time, and if this maximum acceleration exceeds the defined value $a_{\text{lat,max}}$, the final time has to be increased. This acceleration extremum of the trajectory $a_{b,\text{max}}$ for a

given final time can be found by setting

$$\dot{a}_{\text{lat}}(t_{\text{max}}) = 0, \quad (2.9)$$

where t_{max} is the time when the extremum occurs, which can be computed using (2.7) as

$$t_{\text{max}} = \frac{C_2}{C_1} + \sqrt{\left(\frac{C_2}{C_1}\right)^2 - \frac{2C_3}{C_1}}. \quad (2.10)$$

The maximum acceleration of the trajectory then is given by

$$a_{b,\text{max}} = -\frac{1}{6}C_1 t_{\text{max}}^3 + \frac{1}{2}C_2 t_{\text{max}}^2 - C_3 t_{\text{max}} + C_4, \quad (2.11)$$

which has to coincide with the predefined value $a_{\text{lat,max}}$. The nominal final time, i.e., the time to reach the stationary end point, is computed by using the initial and end conditions $v_{\text{lat},0} = a_{\text{lat},0} = v_{\text{lat},f} = a_{\text{lat},f} = 0$ and substituting (2.8), (2.10) with (2.11) fulfilling $a_{b,\text{max}} = a_{\text{lat,max}}$ yields

$$t_{f,\text{nom}} = \sqrt{\frac{10}{\sqrt{3}} \frac{1}{a_{\text{lat,max}}} |l_f - l_0|}. \quad (2.12)$$

Hence, the nominal time to reach the end point given a maximum acceleration can be computed.

Remark 1. *The computation of the final time via the bisection method (with starting points $t_{\text{low}} = t_{f,\text{nom}}$ and $t_{\text{high}} = 2t_{f,\text{nom}}$) is the only part of the trajectory planning level that has a variable computation time; however, by appropriate procedures, e.g., defining tolerances or a maximum number of bisection steps, a maximum computation time can be specified.*

Note that it is important to evaluate the polynomial not later than this final time, as shown in Figure 2.8, which is a standard fifth order polynomial that exceeds the nominal final time (depicted by the circle). For details on the generation of these lateral polynomials, the interested reader is referred to [83].

Finally, different endpoints are used according to predefined endpoints,

$$\mathbf{l}_f = [l_{1,f} \quad l_{2,f} \quad \dots \quad l_{n_l,f}]^T, \quad (2.13)$$

where n_l is the total number of generated trajectories in lateral direction to be chosen. It is reasonable to use the lane centers, i.e., n_{lane} end points that depend on the lanewidth w_{lane} . Additionally, two or more points can be used that are within the same lane, but slightly off in order to maintain safe distance to vehicles that drive closely (e.g., trucks that overlap the lane markings).

Note that different longitudinal velocity polynomials $v_s(t)$ have to be generated in order to be able to react to the environment. This generation is often called “velocity profile” and is discussed in the next section.

2. Trajectory Planning for Automated Highway Driving

2.2.2.2. Velocity Profile Generation

Similar to the procedure for lateral behavior (2.7), a set of different polynomials is generated for the longitudinal dynamics. In contrast to the lateral dynamics, the longitudinal velocity is more important on highways than the position, thus the focus of this section is on velocity polynomials: Since there are no conditions on the longitudinal end position, i.e., s_f can typically be chosen freely, the polynomials are reduced to fourth order instead of fifth order.

Remark 2. *However, if constraints on the end position exist, e.g., if the behavioral planning level requires to stop at a certain distance $s_f = s_{\text{stop}}$, it is again recommended to use fifth order polynomials as described in the previous section.*

For the general highway driving task, the velocity has to be generated so that the jerk is minimized,

$$J = \int_{t_0}^{t_f} \frac{1}{2} j_s(t)^2 dt, \quad (2.14)$$

for a initial time t_0 and a given final time t_f , where j_s is the longitudinal jerk. Again, this minimization corresponds to an optimization of ride comfort, and the optimization problem with $t_0 = 0$ is stated as

$$\begin{aligned} & \min_{j_s(t)} \int_0^{t_f} \frac{1}{2} j_s(t)^2 dt, \\ & \text{s.t.} \quad \dot{v}_s(t) = a_s(t), \\ & \quad \quad \dot{a}_s(t) = j_s(t), \\ & \text{with} \quad v_s(0) = v_{s,0}, \quad a_s(0) = a_{s,0}, \\ & \quad \quad v_s(t_f) = v_{s,f}, \quad a_s(t_f) = a_{s,f}. \end{aligned} \quad (2.15)$$

The Hamiltonian H is then given by

$$H = \frac{1}{2} j_s^2 + \lambda_{s,2} a_s + \lambda_{s,3} j_s, \quad (2.16)$$

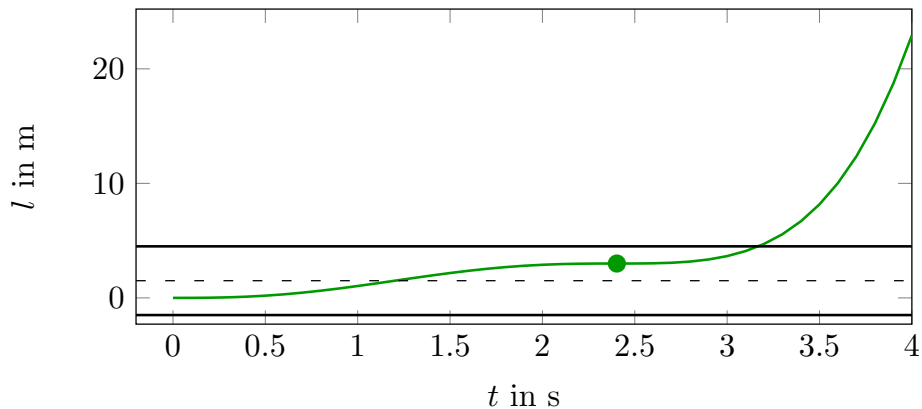


Figure 2.8.: Example for a polynomial to one end point that exceeds the nominal final time (circle).

and the following equations hold:

$$\begin{aligned}
\dot{\lambda}_{s,2} &= -\frac{\partial H}{\partial v_s} = 0, & \Rightarrow \lambda_{s,2} &= C_{s,1}, \\
\dot{\lambda}_{s,3} &= -\frac{\partial H}{\partial a_s} = -\lambda_{s,2}, & \Rightarrow \lambda_{s,3} &= -C_{s,1}t + C_{s,2}, \\
0 &= \frac{\partial H}{\partial j_s} = j_s + \lambda_{s,3}, & \Rightarrow j_s &= -\lambda_{s,3}.
\end{aligned} \tag{2.17}$$

The longitudinal states can then be expressed as

$$\begin{aligned}
j_s(t) &= C_{s,1}t - C_{s,2}, \\
a_s(t) &= \frac{1}{2}C_{s,1}t^2 - C_{s,2}t + C_{s,3}, \\
v_s(t) &= \frac{1}{6}C_{s,1}t^3 - \frac{1}{2}C_{s,2}t^2 + C_{s,3}t + C_{s,4}, \\
s(t) &= \frac{1}{24}C_{s,1}t^4 - \frac{1}{6}C_{s,2}t^3 + \frac{1}{2}C_{s,3}t^2 + C_{s,4}t + C_{s,5},
\end{aligned} \tag{2.18}$$

with the coefficients defined by the initial and end conditions,

$$\begin{aligned}
C_{s,5} &= s_0, \quad C_{s,4} = v_{s,0}, \quad C_{s,3} = a_{s,0}, \\
C_{s,2} &= \frac{(4a_{s,0} + 2a_{s,f})t_f + 6(v_{s,0} - v_{s,f})}{t_f^2}, \\
C_{s,1} &= \frac{6(a_{s,0} + a_{s,f})t_f + 12(v_{s,0} - v_{s,f})}{t_f^3}.
\end{aligned} \tag{2.19}$$

Note, however, that the polynomial of the velocity $v_s(t)$ is used and the coefficient $C_{s,5}$ does not need to be defined. Similar to the lateral approach, the final time of the polynomials can be found by a bisection method, computing the maximum acceleration by setting

$$\dot{a}_s(t_{\max}) = 0, \tag{2.20}$$

which allows to compute the time at which the extremum occurs,

$$t_{\max} = \frac{C_{s,2}}{C_{s,1}}. \tag{2.21}$$

Then, the maximum acceleration using (2.18), (2.21) yields

$$a_{s,\max} = \frac{1}{2}C_{s,1}t_{\max}^2 - C_{s,2}t_{\max} + C_{s,3}, \tag{2.22}$$

and for $a_{s,0} = a_{s,f} = 0$ with (2.19) one arrives at

$$t_{f,\text{nom}} = \frac{-3(v_{s,0} - v_{s,f})}{2a_{s,\max}}. \tag{2.23}$$

Then, for a given longitudinal acceleration, the nominal time to reach the desired velocity is found. Again, a bisection method is applied to find the nominal time so that the longitudinal

2. Trajectory Planning for Automated Highway Driving

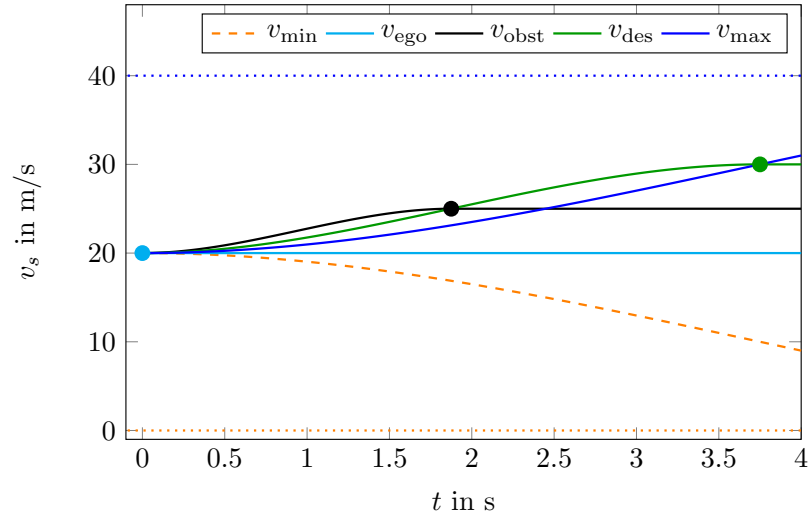


Figure 2.9.: Generated polynomials for the longitudinal velocity from initial to end condition with one considered obstacle. Note that in the given time horizon of 4 seconds, not all end points can be reached.

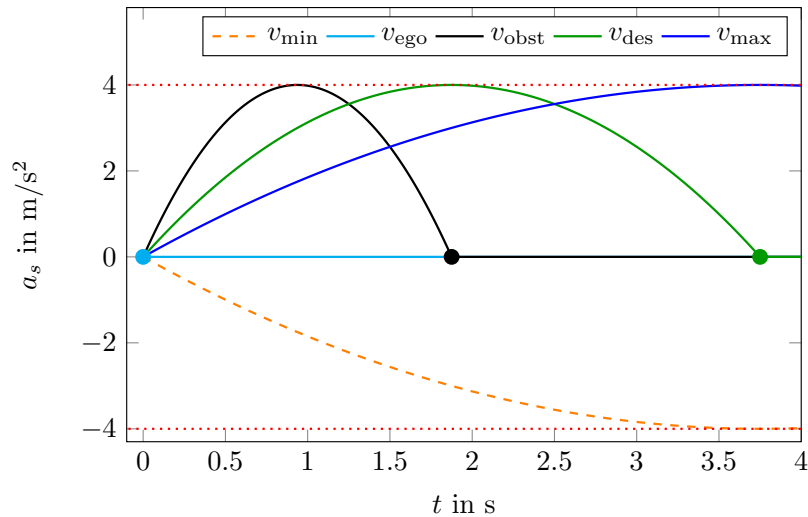


Figure 2.10.: Generated polynomial for longitudinal acceleration with predefined maximum acceleration and one considered obstacle velocity.

acceleration of the polynomial $a_s(t_{\max})$ does not exceed the given “comfortable” acceleration $a_{s,\max}$.

The generated polynomials (2.18) are shown in Figures 2.9, 2.10 for different end points for the velocities $v_{s,f}$. The choice of end conditions for the velocities $v_{s,f}$ is not trivial. A simple idea is given by equally spaced velocities in the range of v_{\min} to v_{\max} . This choice, however, does not allow to follow a vehicle smoothly and oscillations in the velocities and distances with respect to other vehicles may arise. Additionally, one can add the velocities of other participants v_{obst} as end condition in order to avoid these oscillations, assuming these velocities are constant. The desired velocity v_{des} is also added to the end points, and in order to allow constant velocities, the current velocity of the ego vehicle is added as well,

resulting in a total number of n_v polynomials with

$$\mathbf{v}_f = [v_{\min} \quad v_{\max} \quad v_{\text{des}} \quad v_{\text{ego}} \quad \mathbf{v}_{\text{obst}}^T]^T . \quad (2.24)$$

All detected obstacles in the surroundings of the ego vehicle can be taken into account. However, in order to reduce computational effort, only relevant obstacle velocities should be considered, e.g., the velocity of the vehicle in front or other target vehicles.

Remark 3. *In order to simplify a merging maneuver, where the ego vehicle merges between two vehicles on the adjacent lane, one can switch to fifth order polynomials with $s_f = (s_{\text{obst,front}} - s_{\text{obst,rear}})/2$. However, by appropriate choice of the trajectory evaluation, one can also use fourth order polynomials for these maneuvers by choosing $v_{\text{des}} = v_{\text{obst,front}}$ and an appropriate safety distance in the cost evaluation in Section 2.2.4. In both cases, one has to know which vehicles have to be considered on the adjacent lane. However, the same velocity is typically maintained by all vehicles on the lane of interest in dense traffic. Hence, fourth order polynomials may be a better choice than fifth order polynomials.*

2.2.2.3. Set Generation

In order to evaluate the safety of the different trajectories, the generated polynomials in lateral and longitudinal direction have to be combined and then investigated in spatial coordinates (s, l) . For each generated polynomial in l according to (2.13), all velocity profiles v_s with the end points in (2.24) are used. The overall number of generated trajectories in this set is given by $n_1 = n_l \cdot n_v$. The trajectories in spatial coordinates (s, l) then read as

$$\begin{aligned} s(t) &= \int_0^t \left(\frac{1}{6}C_{s,1}\tau^3 - \frac{1}{2}C_{s,2}\tau^2 + C_{s,3}\tau + C_{s,4} \right) d\tau , \\ l(t) &= -\frac{1}{120}C_1t^5 + \frac{1}{24}C_2t^4 - \frac{1}{6}C_3t^3 + \frac{1}{2}C_4t^2 + C_5t + C_6 , \end{aligned} \quad (2.25)$$

with the coefficients given in (2.19), (2.8), as shown in Figure 2.11 for a prediction horizon $t_{\text{pred},1} = 3$ s. Note that the end points of the lateral polynomial (2.13) differ from the endpoints of the trajectories due to different velocities.

In order to consider whole maneuvers in the prediction, e.g. overtaking, a second trajectory set is generated. This set is computed from each endpoint of the first set while maintaining a constant velocity, i.e., $\mathbf{v}_f = v_{s,f}$ of the corresponding velocity end point of the first set, see Figure 2.12. With only one velocity profile, the number of trajectories in the second generation cycle is thus $n_2 = n_l$.

The maneuvers can thus be planned over a prediction horizon of $t_{\text{pred}} = t_{\text{pred},1} + t_{\text{pred},2}$, where the first set is predicted until $t_{\text{pred},1}$, and $t_{\text{pred},2}$ is the time horizon of the second cycle. The prediction horizon of the second maneuver has been chosen as $t_{\text{pred},2} = 1.5$ s. Note that the longer the prediction horizon is chosen, the larger the prediction error will be. Short prediction horizons, however, do not allow to predict collisions. In simulation studies, a reasonable time horizon of $t_{\text{pred}} = 4.5$ s has been found for velocities up to 40 m/s. For larger values, e.g. $t_{\text{pred}} = 6$ s, the vehicle is very conservative and for too small time horizons, e.g., $t_{\text{pred}} = 1.5$ s, a collision cannot be predicted in time and the vehicle will drive

2. Trajectory Planning for Automated Highway Driving

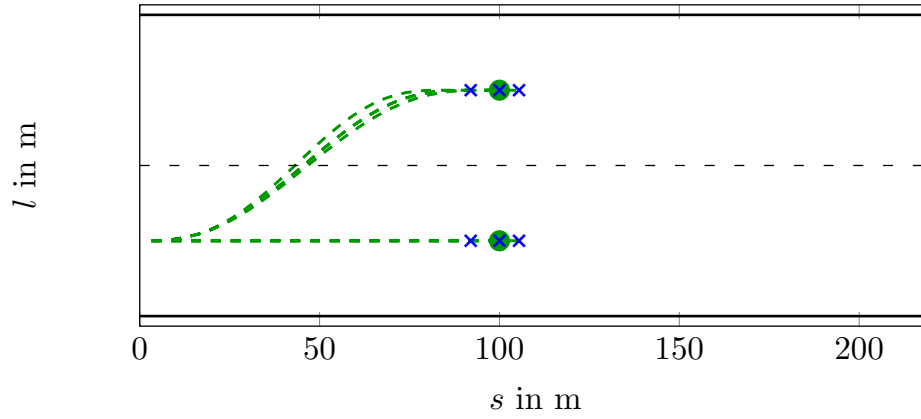


Figure 2.11.: Set of generated trajectories to different lateral end points (green circles) combined with three velocity end points (blue crosses) for one planning cycle

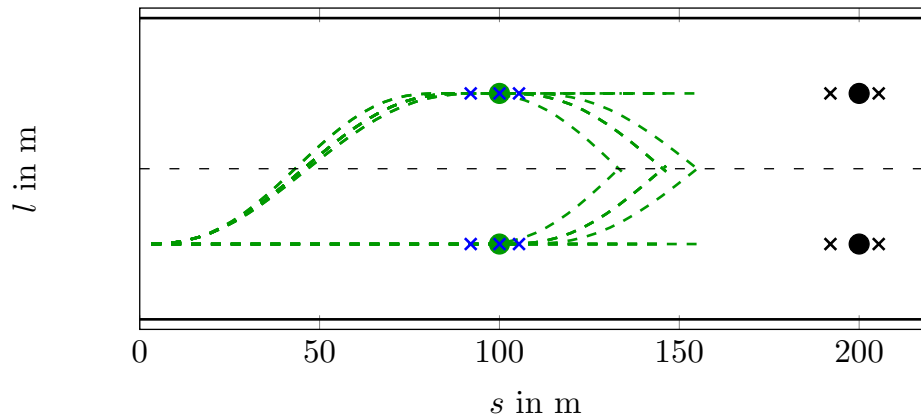


Figure 2.12.: Set of generated trajectories to different end points with two cycles. In the first cycle, two lateral endpoints are combined with three velocities (end points depicted by blue crosses). Two lateral end points in the lane centers and a constant velocity have been considered in the second cycle, where $t_{\text{pred},1} = 3\text{ s}$ and $t_{\text{pred},2} = 1.5\text{ s}$. Combining the first and the second cycle yields the black end points.

too aggressive maneuvers. The total set consists of $n_1 + n_2 = n_l^2 \cdot n_v$ trajectories and is shown in Figure 2.12.

This second cycle allows to predict the ego vehicle's maneuver in a more realistic way and improves the performance of the ego vehicle: for example in Figure 2.13, when a fast vehicle on the left lane (blue) overtakes the ego vehicle (green) and a slow obstacle (red) in front, this two-cycle generation allows to predict the overtaking maneuver of the ego vehicle. Then, the ego vehicle can either pass the slow obstacle before the fast vehicle if the velocity difference to the blue vehicle is small (cyan trajectory of the blue vehicle); or the ego vehicle will not be able to overtake in time and waits for the fast vehicle to pass (blue dotted line). Note that with only one cycle, predicting the overtaking maneuver is not possible and hence, the ego vehicle will in many cases wait for the fast vehicle to pass. This conservative behavior certainly increases the safety, but the acceptance of the driver will be low if the vehicle will not pass a slow obstacle.

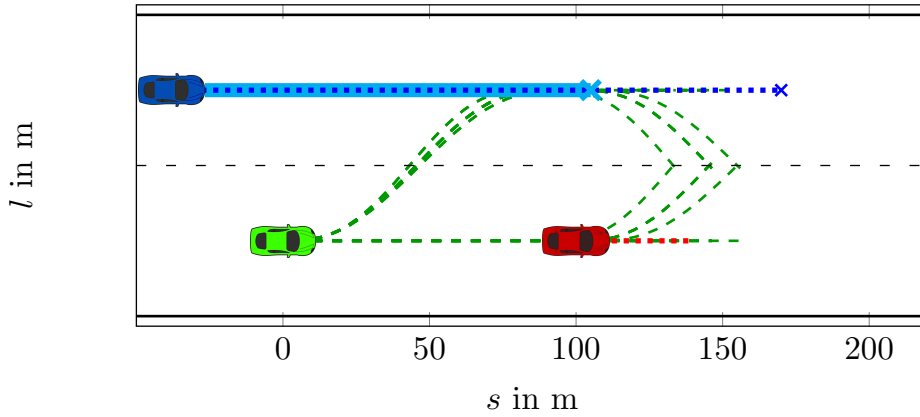


Figure 2.13.: The set of generated trajectories to different end points with two cycles yields collision-free trajectories of the green vehicle that pass the red vehicle and avoid the blue vehicle.

In case of emergencies, it is reasonable to use more cycles in order to achieve more possible trajectories. For example, three sets can be computed iteratively, where each set has a time horizon of $t_{\text{pred},r} = 1.5\text{s}$ and uses all velocity endpoints, thus resulting in a set of $(n_l \cdot n_v)^3$ trajectories. In Figure 2.14, the generated sets for the two-cycle and the three-cycle planning are shown. The number of trajectories is much larger for the three-cycle case, but the possibility to find a collision-free path also increases. However, in standard highway maneuvers, the three-cycle generation is not necessary and hence, the two-cycle planning is used. If the two-cycle approach does not yield collision-free trajectories, a flag can be sent to the behavioral planning that triggers either an emergency system or the three-cycle planning.

Instead of generating more trajectories, it is reasonable to plan with a small sampling time T_{TP} in order to cover more possible trajectories, see Figure 2.15 for two and three consecutive planning time steps. Note that with a high sample rate, it is also possible to cancel maneuvers, e.g., stop a lane change, and the planning level is reactive to changes in the environment.

In order to evaluate the overall costs of the trajectories, the obstacle have to be predicted as in Figure 2.13, which will be described subsequently.

2. Trajectory Planning for Automated Highway Driving

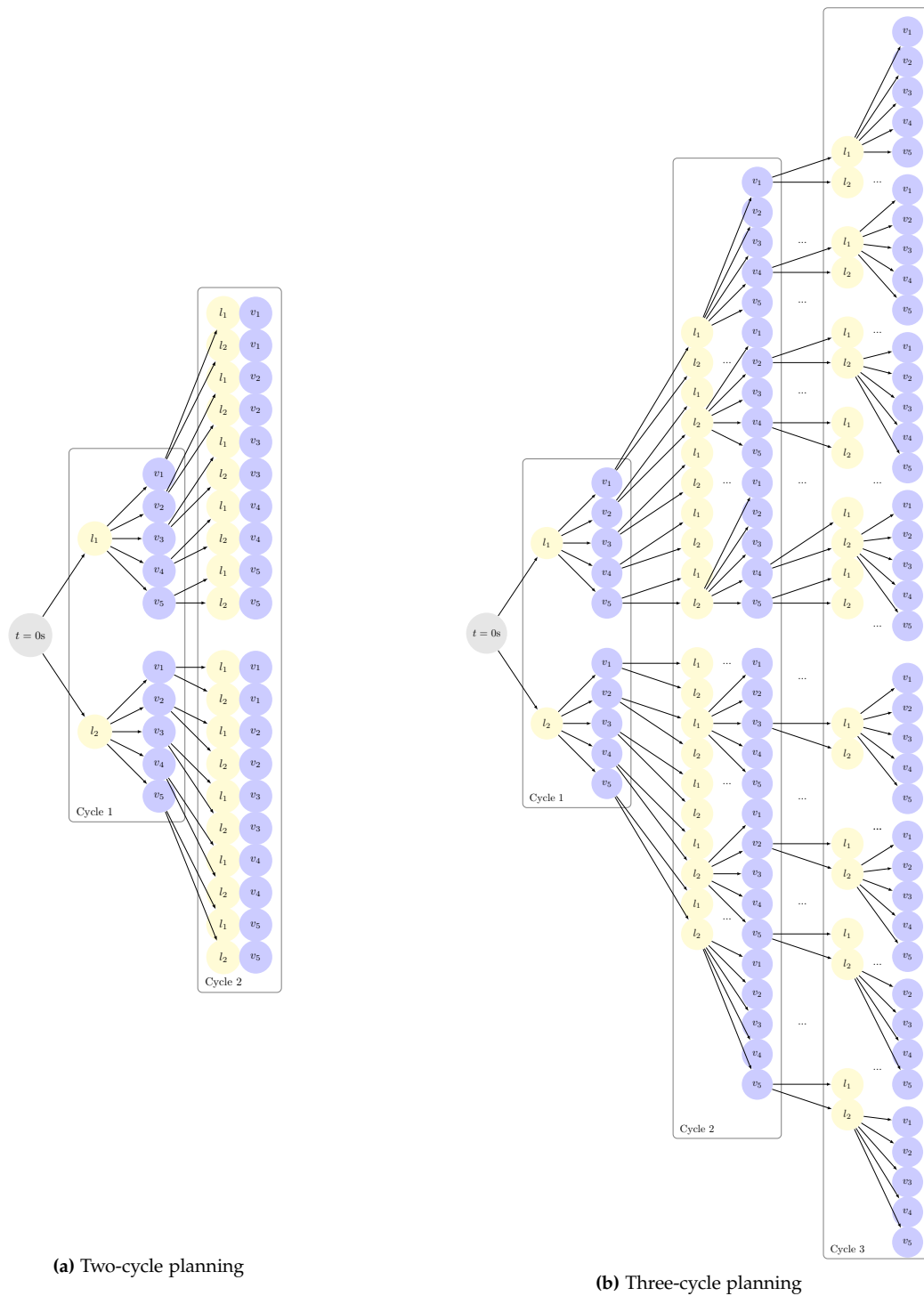
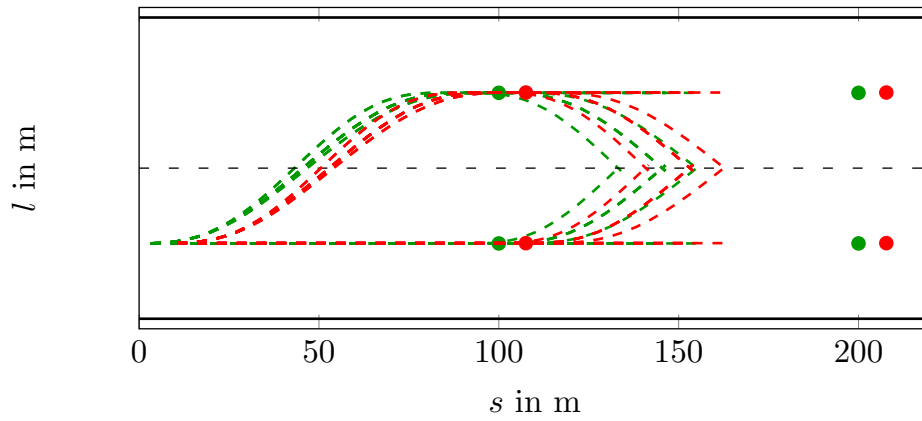
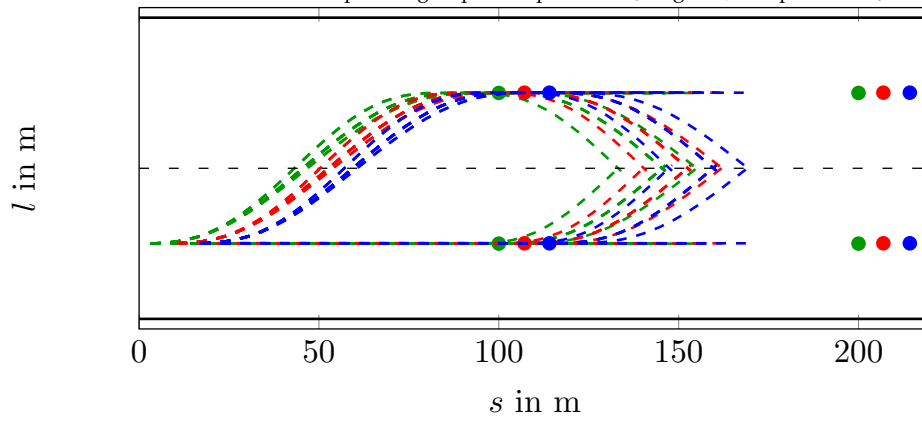


Figure 2.14.: Different set generation methods can be used for different scenarios. The two-cycle planning approach should be used for standard highway driving. Three cycles generate more possible trajectories and can be executed if the standard planning does not yield collision-free trajectories.



(a) Two consecutive planning steps: end points at t_0 in green, end points at $t_0 + T_{TP}$ in red.



(b) Three consecutive planning steps: end points at t_0 in green, end points at $t_0 + T_{TP}$ in red, end points at $t_0 + 2T_{TP}$ in blue.

Figure 2.15.: Planning with a high sample rate is necessary to generate various possible trajectories and thus react to changes in the environment.

2. Trajectory Planning for Automated Highway Driving

2.2.3. Prediction of Obstacles

The trajectory planning approach can also be used for prediction of obstacle trajectories: first a set of possible trajectories in (s, l) coordinates is generated and then, this set is evaluated based on a cost function, as described in Section 2.2.4. The best trajectory is then the prediction of the obstacle used by the ego vehicle.

In order to speed up computation, it is assumed that the velocity of each obstacle is constant during one prediction. Moreover, there is only one end point l_f per lane, i.e. a total number of n_{lane} trajectories for each obstacle. The lateral polynomials are then computed as in (2.7).

In the evaluation of the possible trajectories, other obstacles and the choice of lane are considered, but other cost components are not incorporated: the comfort and efficiency of other obstacles is not important. Instead, overtaking maneuvers of obstacles need to be considered in order to avoid collisions with other traffic participants. Otherwise, the obstacles should drive on the right lane, which is considered by the lane component in the cost function. For details, see Section 2.2.4.

Remark 4. *The constant velocity of the obstacle is not restrictive if the obstacle prediction is performed at a high rate. For example, if the planning level is sampled with time $T_{\text{TP}} = 0.1$ s, then the change in the velocity of an obstacle can be detected very fast and can thus be incorporated in the next step of the prediction. In simulation studies, the prediction with constant velocities and high sample rate yields better results than incorporating the acceleration of the obstacle in the prediction of the velocity profile. Moreover, the acceleration of the obstacle might be difficult to measure. Hence, a constant velocity profile for obstacles is sufficient.*

Remark 5. *Note that the trajectories (2.7), (2.18) are continuous functions in time. However, since the planning level is implemented on an embedded unit, a certain sampling time T_{TP} is used and computations can only be performed at these time instances. Hence, the following discussions are given in discrete time, with the time instances*

$$t \in [T_{\text{TP}}, t_{\text{pred}}] \rightarrow k \in \{0, 1, \dots, N_{\text{pred}}\}, \quad (2.26)$$

with $N_{\text{pred}} = (t_{\text{pred}} - T_{\text{TP}}) / T_s$ and the sampling time of the predicted values T_s . Then, a continuous signal $f(t)$ is sampled by $f_k = f(T_{\text{TP}} + kT_s)$. Note that T_s can be chosen different from the planning sample time. In order to not mix up indices of obstacles and time, the discrete-time notation used is $f(k)$ subsequently.

Note that basic maneuvers such as lane changes and overtaking intentions can be predicted by this approach. However, basic maneuvers in the obstacle prediction might be insufficient: if the other vehicle is behaving in a completely different way, e.g., driving on the center line between two lanes for a long time period, this cannot be predicted if only the lane centers are used as lateral end points, see Figure 2.16. In order to deal with unpredictable drivers, a prediction observer has been implemented as described in the following subsection.

2.2.3.1. Prediction Observer

The error between the predicted and the actual states is computed as

$$e(k) = w_{s,p}|s_p(k) - s_{\text{obst}}(k)| + w_{l,p}|l_p(k) - l_{\text{obst}}(k)|, \quad (2.27)$$

where $w_{s,p} \geq 0$, $w_{l,p} > 0$ are parameters to be defined, and $s_p(k), l_p(k)$ are the predicted states for the current time instance k , computed at the last planning step $k - 1$, and $s_{\text{obst}}(k), l_{\text{obst}}(k)$ are the actual states of the obstacle.

The prediction error E_p for one obstacle is then computed based on previous prediction errors (2.27) as

$$E_p = \sum_{i=0}^{N_e-1} e(k - N_e + i), \quad (2.28)$$

where N_e is the number of considered errors and k is the current time instance. If this prediction error is larger than a predefined tolerance, i.e., $E_p > P_{\text{tol}}$, the current lateral state $l_{\text{obst}}(k)$ is held constant over the prediction horizon instead of generating polynomials to different lateral end points. This allows to consider the case that a driver stays in-between two lanes as shown in Figure 2.16b. Without this measure, the ego vehicle predicts that the obstacle changes lane to the right-hand lane. Thus the TP predicts that the ego vehicle can pass the obstacle, and the ego vehicle accelerates to the desired velocity. However, since the other vehicle does not change lane, this acceleration is not safe and the ego vehicle has to brake to be able to stop behind the obstacle. With the prediction observer, this situation can be improved since the ego vehicle does not predict that the obstacle moves to the right, if the other vehicle has been on the center line for a certain time.

As soon as the obstacle can be predicted correctly, i.e., driving as predicted by the polynomials, the error sinks below another threshold $E_p < P_{\text{tol},2}$, and the standard prediction can be used.

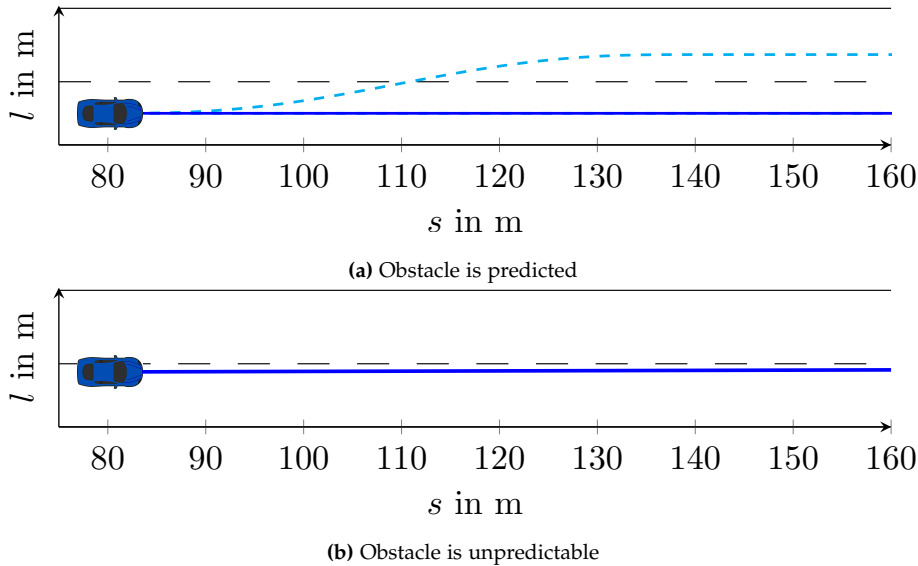


Figure 2.16.: Prediction of an obstacle using a prediction observer.

2. Trajectory Planning for Automated Highway Driving

The output of this obstacle prediction are the samples of the predicted obstacle trajectories $\{s_{\text{obst}}(k), l_{\text{obst}}(k), v_{\text{obst}}(k)\}$ for $k = \{0, 1, \dots, N_{\text{pred}}\}$, where the velocity is constant $v_{\text{obst}}(k) = v_{\text{obst}}(0)$. Note that the same prediction horizon t_{pred} and prediction sampling time T_s as for the ego trajectory set has to be used. These samples are then evaluated in order to assess the safety of the ego vehicle trajectories, as described in the following section.

2.2.4. Evaluation

In order to evaluate the generated ego vehicle trajectory set, a cost function with several components is used, as published in [144]. As already mentioned, the trajectories are sampled at certain time instances and these samples are used in the cost function, where the best trajectory is chosen based on the lowest cost. Note that with this procedure, the computational costs are fixed and no on-line optimization is necessary. In this work, equally spaced samples as in Remark 5 have been used with a sampling time T_s . In the following considerations, the discrete samples of the trajectories are used and the different components of the cost function are introduced.

2.2.4.1. Obstacles

Collision avoidance is the most important component of the cost function. In order to maintain a safe distance to all other traffic participants, the obstacle cost depends on the distance in longitudinal and lateral direction to all other vehicles. The proximity to obstacles is represented by an ellipse as described in [83]. A collision is said to occur as soon as the ellipse is entered, i.e., if a certain safety distance is no longer maintained. Then, the costs for this trajectory is set to a very large value or infinity. Note that uncertainties in the position or prediction of obstacles can be considered by increasing the safety distances.

In order to compute the ellipses for obstacle i , the safety distance parameters have to be defined, which are the axes of the ellipses $s_{\text{safe},i}$ and $l_{\text{safe},i}$. Since the longitudinal safety distance to the vehicle in front should depend on the velocity of the ego vehicle $v_s(k)$, it can be defined as

$$s_{\text{safe},i}(k) = D_{\text{min}} + v_s(k)t_h, \quad (2.29)$$

where $D_{\text{min}} \geq 0$ is the minimum distance in standstill, e.g., $D_{\text{min}} = 5$ m, and t_h is the so-called time-headway in seconds. The additional velocity-dependent distance $v_s(k)t_h$ is necessary in order to be able to react fast enough to changes of the vehicle in front and is also used for ACC or platooning scenarios as discussed in Part II of this thesis.

In addition, the velocity difference of the vehicles can be taken into account, where the corresponding distance is given by

$$s_{\text{vel},i}(k) = c_v(v_{\text{obst},i}(k) - v_s(k)), \quad (2.30)$$

where $c_v \geq 0$ is a parameter that can be chosen. Note that the safety distance to rear vehicles, e.g., after the overtaking maneuver, should depend on the velocity of the rear

vehicle and hence, the following safety distances with respect to front / rear vehicles are defined:

$$\begin{aligned} s_{\text{rear},i}(k) &= \max(s_{\text{safe},i}(k) + s_{\text{vel},i}(k), D_{\text{min}}) , \\ s_{\text{front},i}(k) &= \max(s_{\text{safe},i}(k) - s_{\text{vel},i}(k), s_{\text{safe},i}(k)) . \end{aligned} \quad (2.31)$$

In lateral direction l , the safety distance to obstacle i for all time steps is given by

$$l_{\text{safe},i} = \frac{W_{\text{ego}} + W_{\text{obst},i}}{2} + D_{l,\text{min}} , \quad (2.32)$$

with a parameter $D_{l,\text{min}} > 0$ and the width of the ego vehicle W_{ego} and the width of the obstacle's vehicle $W_{\text{obst},i}$.

Since overtaking on the right lane is not permitted in Europe, two additional ellipses are generated for the lateral position. For this purpose, another safety distance in l is defined by

$$l_{r,i} = n_{\text{lane}} w_{\text{lane}} , \quad (2.33)$$

with the number of lanes n_{lane} and lanewidth w_{lane} . Thus, the ellipse of a slow vehicle on the left lane is extended to the right so that passing on the right lane is not possible. In Figure 2.17, the different axes definitions (2.29), (2.31), (2.32), (2.33) are shown.

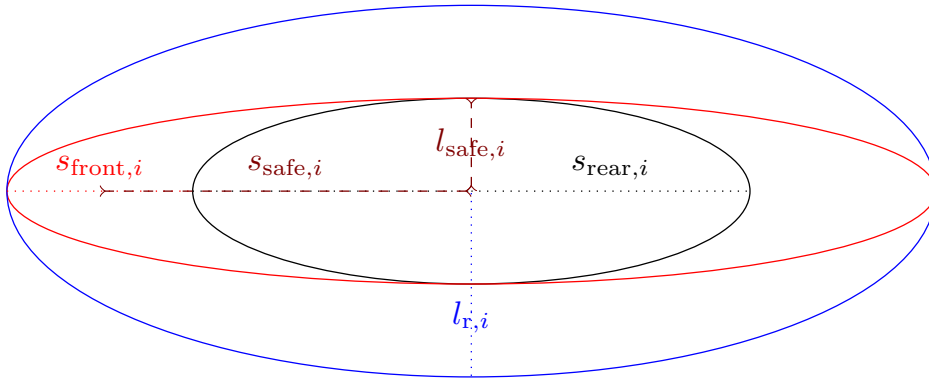


Figure 2.17.: Ellipses with different axes in longitudinal and lateral direction to ensure safety distances.

This approach leads to four different ellipsoids that can be written as

$$\begin{aligned} g_{1,i}(k) &= f(s_{\text{front},i}(k), l_{\text{safe},i}) , \\ g_{2,i}(k) &= f(s_{\text{rear},i}(k), l_{\text{safe},i}) , \\ g_{3,i}(k) &= f(s_{\text{front},i}(k), l_{r,i}) , \\ g_{4,i}(k) &= f(s_{\text{rear},i}(k), l_{r,i}) . \end{aligned} \quad (2.34)$$

The level G of the general coordinates (s_g, l_g) with respect to an ellipsoid f in the origin with axes (σ_s, σ_l) is computed by

$$G(s_g, l_g, f(\sigma_s, \sigma_l)) = \frac{s_g^2}{\sigma_s^2} + \frac{l_g^2}{\sigma_l^2} . \quad (2.35)$$

2. Trajectory Planning for Automated Highway Driving

Then, for $G = 1$, the coordinates (s_g, l_g) are on the ellipse, for $G > 1$ outside the ellipse, otherwise inside.

In order to have decaying costs outside the ellipse and a certain level when intersecting the ellipse with increasing costs close to the obstacle, the levels are scaled by

$$\hat{G}(s_g, l_g, f(\sigma_s, \sigma_l)) = \frac{1}{G(s_g, l_g, f(\sigma_s, \sigma_l))}. \quad (2.36)$$

In Figure 2.18, an example for a variation of the coordinate s_g is shown, where the levels (2.35) and (2.36) are plotted for a constant $l_g = 0$ and given axes. Note that in [83], the square root of G is used in (2.36), as shown by the red line. However, similar costs can be achieved with (2.36) with the advantage that the costs increase faster close to the obstacle.

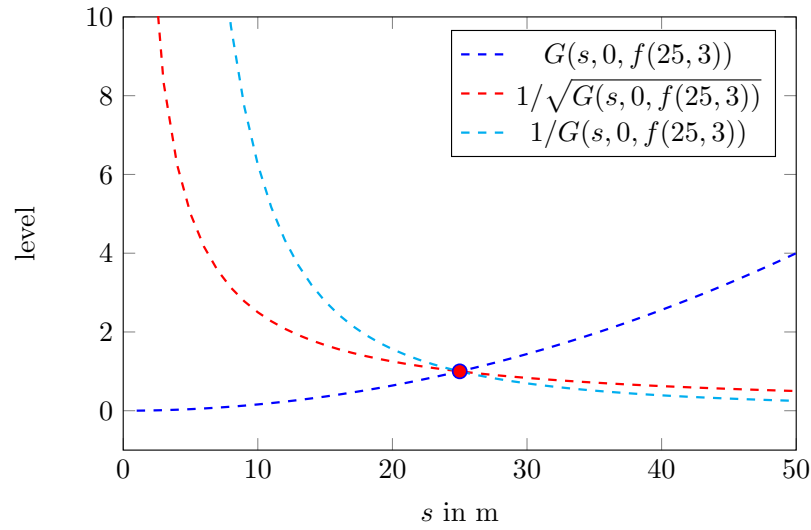


Figure 2.18.: Example for the different levels used as costs for obstacle avoidance with axes $\sigma_s = 25$, $\sigma_l = 3$. The circle indicates the intersection of the coordinates with the ellipse.

In order to compute the proximity to the obstacles for the cost component, the distances between ego vehicle and obstacle are computed by

$$\begin{aligned} \Delta s_i(k) &= s_{\text{ego}}(k) - s_{\text{obst},i}(k), \\ \Delta l_i(k) &= l_{\text{ego}}(k) - l_{\text{obst},i}(k). \end{aligned} \quad (2.37)$$

Then, the costs for obstacle i and for all ellipses $e = 1, 2, 3, 4$ in (2.34) are obtained using (2.36) with

$$\hat{g}_{e,i} = \hat{G}(\Delta s_i(k), \Delta l_i(k), g_{e,i}(k)). \quad (2.38)$$

The ellipse e is hence entered if $\hat{g}_{e,i} \geq 1$ holds.

Note, however, that the four different ellipses cannot be active at the same time; hence, a switching function is used in order to activate the appropriate axis depending on the longitudinal and lateral position with respect to the obstacle. In [83], only longitudinal

switching is performed; In this work, however, two switching functions are used. A general switching function as function of x can be defined by

$$P(x) = \underbrace{\frac{1}{1 + \exp(-\beta x)}}_{S_1(x)} f_1 + \underbrace{\frac{1}{1 + \exp(\beta x)}}_{S_2(x)} f_2, \quad (2.39)$$

with β a parameter that is chosen. An example for a one-dimensional switching function (2.39) is shown in Figure 2.19, with $f_1 = 1$ and $f_2 = 2$ and a constant parameter $\beta = 3$. The parameter β is the “slope” of the switching function (2.39) and the effect of different parameters β is shown in Figure 2.20.

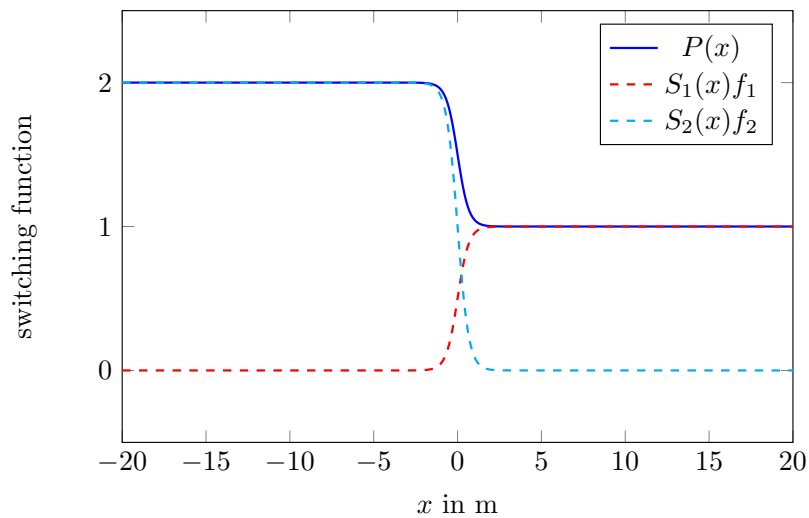


Figure 2.19.: Switching function with S_1 and S_2 from (2.39) and $f_1 = 1$, $f_2 = 2$.

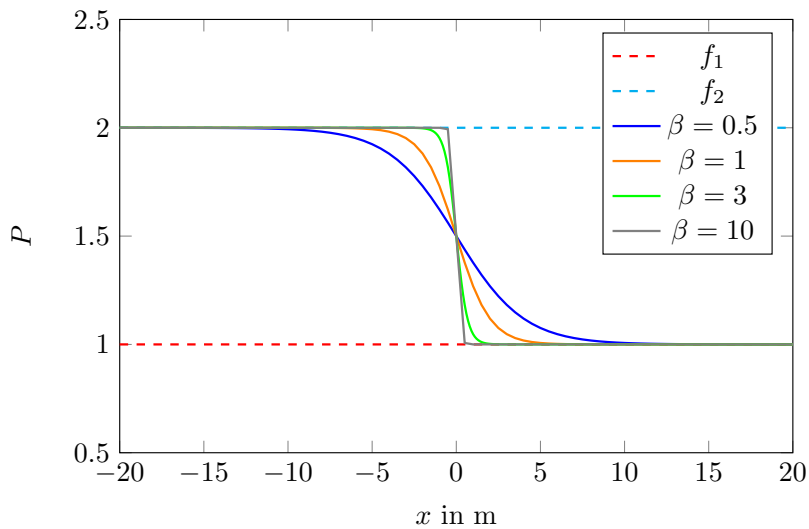


Figure 2.20.: Switching function for different parameters β .

2. Trajectory Planning for Automated Highway Driving

Let the lateral switching function with the ellipse levels (2.38) be defined by

$$\begin{aligned}\hat{q}_{f,i}(\Delta s_i(k), \Delta l_i(k)) &= \frac{1}{1 + \exp(-\beta \Delta l_i(k))} \hat{g}_{1,i} + \frac{1}{1 + \exp(\beta \Delta l_i(k))} \hat{g}_{3,i}, \\ \hat{q}_{r,i}(\Delta s_i(k), \Delta l_i(k)) &= \frac{1}{1 + \exp(-\beta \Delta l_i(k))} \hat{g}_{2,i} + \frac{1}{1 + \exp(\beta \Delta l_i(k))} \hat{g}_{4,i},\end{aligned}\quad (2.40)$$

then the overall switching function yields

$$\begin{aligned}p_i(\Delta s_i(k), \Delta l_i(k)) &= \frac{1}{1 + \exp(-\beta \Delta s_i(k))} \hat{q}_{f,i}(\Delta s_i(k), \Delta l_i(k)) \\ &+ \frac{1}{1 + \exp(\beta \Delta s_i(k))} \hat{q}_{r,i}(\Delta s_i(k), \Delta l_i(k)).\end{aligned}\quad (2.41)$$

Fast vehicles on the left lane, however, are allowed to overtake; thus, only $\hat{q}_{f,i} = \hat{g}_1$ and $\hat{q}_{r,i} = \hat{g}_2$ are used if the relative velocity $\Delta v_i = v_{\text{obst},i} - v_{\text{ego}}$ is positive. Otherwise, the fast vehicle might “push” the ego vehicle forward. The results using this overall switching function are depicted in Figure 2.21. The ego vehicle keeps a velocity-dependent safety distance that also depends on the relative velocity and the relative distance of the obstacle (a). Passing a slow vehicle on the right is penalized (b), but a faster vehicle on the right lane will not influence the ego vehicle’s trajectory (c and d).

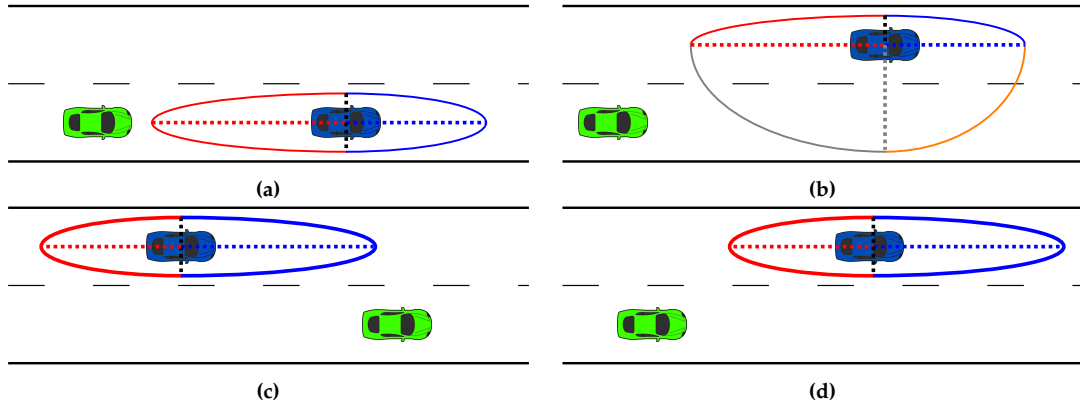


Figure 2.21.: Different Ellipses are active depending on the relative velocity between the ego vehicle (green) and the blue obstacle. a) slow vehicle in front on the right lane. b) slow vehicle in front on the left lane, passing is not allowed. c) and d) faster vehicle is passing the ego vehicle.

The ellipse levels of all obstacles are computed over the entire prediction horizon $k = \{0, 1, \dots, N_{\text{pred}}\}$. If an ellipse of obstacle i is entered at time index $k_{\text{collision},i}$, this index is stored and used to weight the component. Moreover, the difference in velocities at this time instance of safety distance violation $\Delta v_{\text{collision},i}$ is stored.

Then, the final cost for proximity to obstacle i is the maximum value of the ellipse level (2.41) over the entire prediction horizon, i.e.,

$$M_i = \max_k (p_i(\Delta s_i(k), \Delta l_i(k))) . \quad (2.42)$$

Note that if M_i is larger than one, the predicted ego vehicle's trajectory has entered the predicted ellipse of vehicle i . However, this does not necessarily mean that a collision occurs; only the safety distance has been violated.

The costs of this violation can be set to infinity as, e.g., in [83]. However, this means that an actual collision with an obstacle has the same cost as violating the safety distance by, e.g., 1 m, which is not reasonable. Hence, in this work, the collision costs depends on further factors: first, in order to penalize collisions that happen with rear vehicles in the near future, the collision cost is weighted by time index of violation $k_{\text{collision},i}$. Note that collisions with rear vehicles that are predicted at the end of the prediction horizon might still be resolved. It is assumed that vehicles in front do not react to the ego vehicle, but rear vehicles might, for example, adapt their velocities. Moreover, if an obstacle is behind the ego vehicle, the corresponding cost of the trajectory is less than for front vehicles. These weights are necessary to prevent the ego vehicle from being "stuck" behind a slower front vehicle, since the ego vehicle will not overtake, if other vehicles are approaching from behind. Second, large velocity differences $\Delta v_{\text{collision},i}$ at the violation are worse than small velocity differences: since the other obstacles are assumed to have constant velocities, a small difference might be compensated by the other obstacle (especially in the case of rear vehicles). These additional weights are necessary to avoid too conservative behavior of the ego vehicle.

The cost computation for one obstacle i has thus been implemented as

$$f_{\text{obstacle},i} = \begin{cases} w_{\text{ob},i} \cdot M_i & \text{if } M_i < 1, \text{ no collision} \\ \frac{\alpha}{3} + |w_{\text{ob},i} \cdot M_i \cdot \Delta v_{\text{collision},i}| & \text{if } M_i > 1, \text{ collision,} \end{cases} \quad (2.43)$$

where α is a scaling factor of the total cost. It is used to give this obstacle cost a priority by representing at least one third of the total cost. The weight for a rear obstacle is given by

$$w_{\text{ob},i} = \begin{cases} 0.1(N_{\text{pred}} - k_{\text{collision},i}) & \Delta s_i(0) \geq 0, \text{ rear vehicle} \\ 1 & \Delta s_i(0) < 0, \text{ front vehicle.} \end{cases} \quad (2.44)$$

Then, the overall obstacle cost is computed as the sum of costs over all obstacles \mathcal{N}_o ,

$$f_o = \sum_{i \in \mathcal{N}_o} f_{\text{obstacle},i}. \quad (2.45)$$

In other publications, the maximum of the individual obstacle costs, i.e., $\max_i f_{\text{obstacle},i}$, has been chosen as overall cost. However, this means that a collision with one obstacle is equally rated as collisions with multiple obstacles. Hence, the sum over all obstacles has been considered.

2.2.4.2. Desired Lane

In Europe, the desired lane is the rightmost lane in standard highway driving scenarios. However, the driver can also set a desired lane, or the mission planning level requires that

2. Trajectory Planning for Automated Highway Driving

a certain lane at intersections is taken. The desired lane is thus an input to the trajectory planning level. The corresponding cost for reaching the desired lane is computed as

$$f_{\text{lane}} = \frac{|l(N_{\text{pred}}) - l_{\text{des}}|}{w_{\text{lane}}}, \quad (2.46)$$

with the width of the lane w_{lane} as scaling factor. However, this factor only depends on the final value. In addition, a dynamic part should be considered and hence the “time-to-merging” is used to penalize maneuvers that take longer to reach the desired value, i.e.,

$$f_{\text{merge}} = \sum_{k=0}^{N_{\text{pred}}-1} \frac{|l(k) - l_{\text{des}}|}{w_{\text{lane}}}. \quad (2.47)$$

In Figure 2.22, an example for the lane costs with two different trajectories is shown. The blue trajectory enters the desired lane faster than the green one and has thus lower lane costs, which are computed by the distances that are plotted between the trajectory samples (denoted by circles).

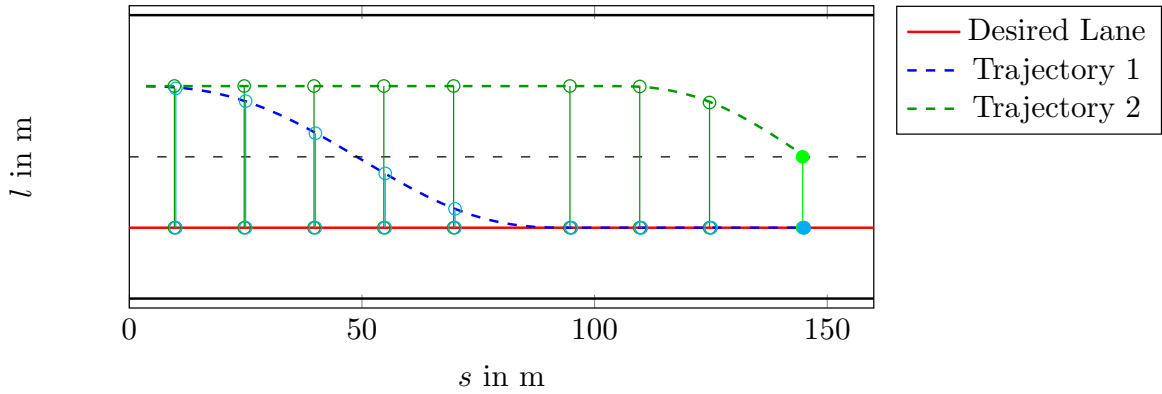


Figure 2.22.: Costs schematically depicted for the desired lane: static (filled circles at end points) and dynamic components (not filled circles) are used to penalize deviations from the desired lane.

The total lane costs are given by different weights for the static (2.46) and dynamic (2.47) parts,

$$f_1 = f_{\text{lane}} + w_{\text{merge}} f_{\text{merge}}. \quad (2.48)$$

2.2.4.3. Lane Center

The vehicle has to stay in the center of the lane, if the other participants behave correctly. In some cases, however, e.g., a truck that overlaps the lane marking, it is preferable to use a trajectory that allows to keep a safe distance to the truck. Thus, if more endpoints per lane are computed, the cost can be defined as in [83] by

$$f_{\text{center}}(k) = \frac{\cos\left(\frac{2\pi(2l(k) - w_{\text{lane}}(n_{\text{lane}} - 1))}{2w_{\text{lane}}}\right) + 1}{2}. \quad (2.49)$$

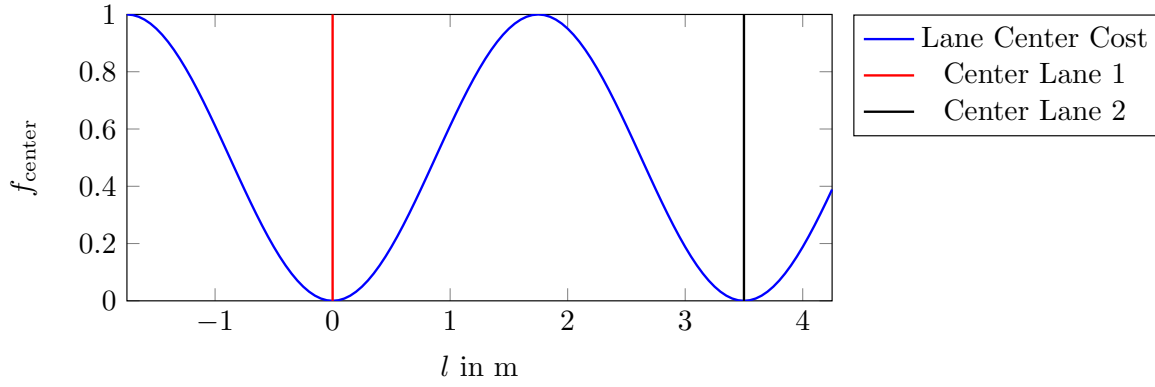


Figure 2.23.: Costs for the lane center with a lanewidth of $w_{\text{lane}} = 3.5$ m.

This cost is zero in the lane center, and increases towards the lane borders as shown in Figure 2.23.

Again, static and dynamic parts can be used and weighted differently similar to the lane costs (2.48). In this work, however, only one endpoint per lane has been considered to speed up computations.

2.2.4.4. Desired Velocity

If the velocity at the end of the prediction horizon is not corresponding to the desired value that is set by the behavioral planning level, a penalty is added,

$$f_{\text{vel,stat}} = \frac{|v_s(N_{\text{pred}}) - v_{\text{des}}|}{v_{\text{max}}}, \quad (2.50)$$

where v_{max} is the maximum velocity allowed by either traffic signs or operational constraints. Additionally, if a trajectory reaches its desired velocity faster than others, its costs are less. This dynamic part is computed as

$$f_{\text{vel,dyn}} = \sum_{k=0}^{N_{\text{pred}}-1} |v_s(k) - v_{\text{des}}|. \quad (2.51)$$

Then, the total velocity cost is a weighted sum of the static and dynamic part,

$$f_{\text{velocity}} = f_{\text{vel,stat}} + w_{\text{vel,dyn}} f_{\text{vel,dyn}}. \quad (2.52)$$

In Figure 2.24, three different trajectories are shown. The cyan trajectory does not reach the desired velocity and has thus large dynamic and static costs. The green and blue trajectory both reach the desired velocity, but the blue trajectory has a lower dynamic part. However, the blue trajectory has the largest acceleration. This maximum acceleration is defined in the velocity profile generation and has a huge impact on the performance. Since the maximum acceleration is a predefined value, adding a cost for the maximum is not reasonable. Instead, in order to avoid oscillating behavior in the velocity, i.e., frequent acceleration and deceleration, a cost is defined as described in the following subsection.

2. Trajectory Planning for Automated Highway Driving

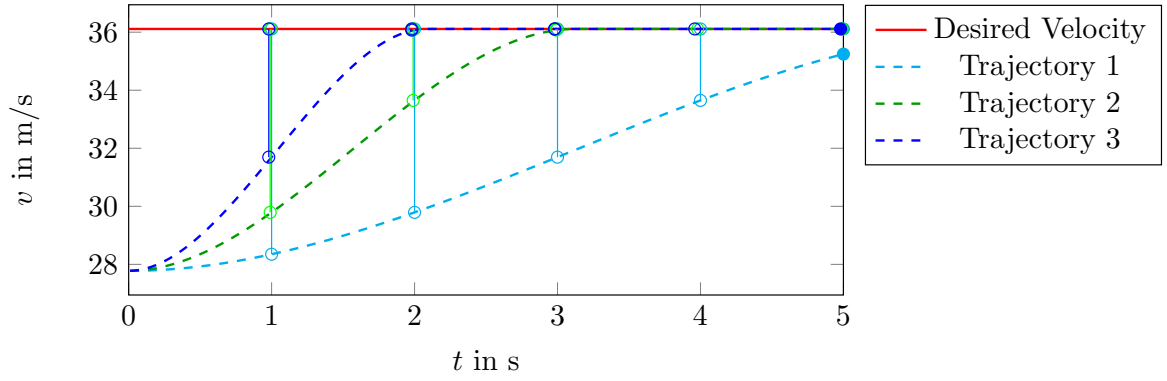


Figure 2.24.: Costs for the desired velocity schematically depicted: static (open circles) and dynamic parts (filled circle at end point) of the velocity cost component are used with different weights.

2.2.4.5. Acceleration

Frequent changes from one velocity end point to the other have to be avoided in order to prevent oscillations in the velocity of the ego vehicle. The following cost component penalizes high acceleration at the beginning of a trajectory, i.e., at the first sample of the trajectory, and is used for both lateral acceleration,

$$f_{\text{stat},l} = \frac{\ddot{i}(1)}{a_{\text{lat,max}}} , \quad (2.53)$$

and longitudinal acceleration

$$f_{\text{stat},s} = \frac{\ddot{s}(1)}{a_{s,\text{max}}} . \quad (2.54)$$

Moreover, if the vehicle is affected by high acceleration in one direction, the acceleration in the other direction should be very low. The weights of the lateral acceleration is thus increased if the longitudinal acceleration is high,

$$w_{\text{stat},l} = w_{\text{stat},s} + \max_k \ddot{s}(k) . \quad (2.55)$$

However, this is counter-productive in some scenarios (e.g., merging). A flag is thus used to switch this weighting on or off,

$$w_{\text{stat},l} = w_{\text{stat},s} + \max_k \ddot{s}(k) [1 - m] . \quad (2.56)$$

In simulated merging scenarios, the choice $m = 1$ has been found to be reasonable.

The cost component for static maneuvers considering acceleration (2.53), (2.54) with (2.56) is written as

$$f_{\text{stat}} = w_{\text{stat},s} f_{\text{stat},s} + w_{\text{stat},l} f_{\text{stat},l} . \quad (2.57)$$

Then, the velocity profile 1 in Figure 2.25 has lower costs than the velocity profile 2. Note, however, that these static costs are for a comfortable purpose only and hence, other cost components that are related to safety have larger weights.

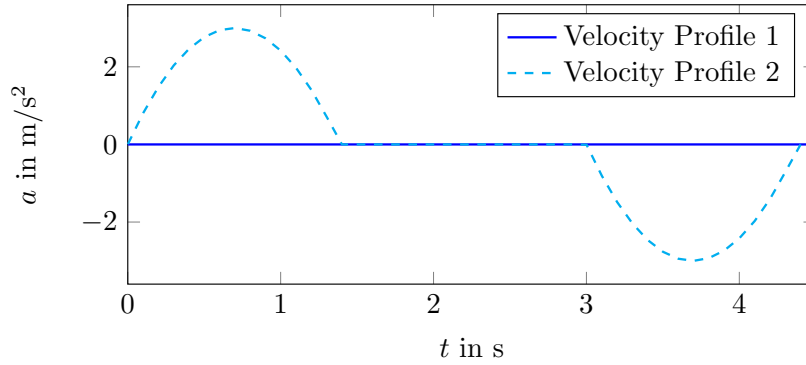


Figure 2.25.: Longitudinal acceleration for two different velocity profiles with the same endpoint

2.2.4.6. Overall Cost

The overall cost function includes all cost components (2.45), (2.48), (2.49), (2.52), (2.57), which are weighted in order to achieve the desired performance of the ego vehicle, and is computed by

$$J_k = \tanh \left(\frac{1}{\alpha} (w_o f_o + w_1 f_1 + w_{\text{center}} f_{\text{center}} + w_{\text{velocity}} f_{\text{velocity}} + w_{\text{stat}} f_{\text{stat}}) \right). \quad (2.58)$$

The weights $w_o, w_1, w_{\text{center}}, w_{\text{velocity}}, w_{\text{stat}}$ have been tuned in simulation according to the following priorities:

1. collision avoidance
2. desired velocity
3. desired lane
4. acceleration

It is important to note that desired lane cost may be higher than the obstacle cost in many situations (e.g., if no obstacle is present), but the maximum cost of obstacles has to be higher than the other costs combined (since collisions MUST be avoided if possible). The maximum cost of velocity should be higher than the maximum lane cost, and the acceleration costs should be lower than all other costs.

A reasonable choice leads hence to satisfying behavior, for example,

- collision avoidance is more important than reaching or maintaining the desired velocity and/or lane.
- if collisions cannot be avoided, a collision with less velocity difference and larger time-to-collision is selected.
- if no collision occurs, the desired velocity is reached and maintained, i.e., overtaking maneuvers are executed in case of a slow traffic participant in front of the ego vehicle.
- if the desired velocity is maintained and the desired lane is collision-free, the vehicle changes to or stays on the desired lane.
- a constant velocity is better than acceleration and deceleration intervals. Note that this only has an effect if the other costs are low or at the same value.

2. Trajectory Planning for Automated Highway Driving

- if merging is required by the behavioral planning level, the velocity can be adapted to the velocity of vehicles on the corresponding lane. Then, if there is a gap for the vehicle to merge, the lane change is performed, even if a collision is predicted in the far future due to small inter-vehicle spacings.
- obstacles in front of or next to the ego vehicle are more important than rear vehicles that are far away. If a fast rear vehicle is approaching, the overtaking maneuver will be finished by the ego vehicle. Otherwise, the ego vehicle might choose the “safer” collision; then, it changes the lane and thus causes a collision with the vehicle that it passes. However, if the rear obstacle is very close and too fast to avoid collisions, the ego vehicle can speed up, if no other rules are violated.
- if the velocity difference with respect to the vehicle in front is small, the ego vehicle does not overtake.

The overall cost (2.58) is scaled by α so that the cost value is in a reasonable range. The tangens hyperbolicus is additionally used in order to saturate the costs such that the value is between 0 and 1. This scaling α has to be adapted to the maximum cost values and depends hence on the other parameters. If the sum of costs for all trajectories is too high (wrongly scaled), the best trajectory can no longer be evaluated.

In a last step, the cost function is filtered in order to avoid frequent switching between trajectories. Note, however, that this filter function is only used in standard driving scenarios, i.e., in “comfortable mode”. If the three-cycle planning is active, no filter is used in order to be able to react as fast as possible. In discrete-time, the filter has been implemented as

$$\hat{J}_k = \hat{J}_{k-1} + \frac{(J_k - \hat{J}_{k-1})T_s}{\tau_f}, \quad (2.59)$$

with sampling time T_s and filter constant $\tau_f > T_s$ (further details in [83]). Consider the example shown in Figure 2.26 with two generated trajectories for the ego vehicle: the green trajectory reaches the desired velocity and ends on the left lane, while the orange trajectory does not reach the desired velocity, and stays on the right lane. The corresponding unfiltered (2.58) and filtered costs (2.59) of the scenario are shown in Figure 2.27. First, the orange trajectory has a low overall cost since the obstacle’s safety distance is not violated. Around $t = 1$ s, however, the predicted end point of the orange trajectory is inside the blue ellipse and the costs increase significantly. The green trajectory has lower costs, since it reaches the desired velocity and is collision-free, but the overall cost also increases due to a smaller distance to the obstacle. The different cost components for one time instance are shown in Figure 2.28. The lane costs of the green trajectory is larger since the right lane is desired, and the green trajectory has lateral and longitudinal acceleration, which results in the larger costs.

Instead of filtering, a new trajectory index can also be accepted if it has been the best trajectory index for a certain amount of time steps (e.g. 10 time steps). Simulation results of the proposed planning level can be found in Chapter 3.

Remark 6. *The parameter selection greatly influences the performance of the ego vehicle. Parameter tuning requires a lot of simulations, but parameters can be interpreted easily. However, the same is true for MPC planning, where similar costs have to be considered.*

2.2.5. Reference Computation

The best trajectories $l(t), v_s(t)$ have been found by the evaluation in Section 2.2.4. Then, the references \mathbf{r} are computed for the low-level controllers so that the path and velocities can be tracked also on the curved road, see Figure 2.3. Hence, the transformation to appropriate references is performed as last step of the planning level and depends on the low-level controller.

2.3. Trajectory Tracking

The tracking task is typically divided into longitudinal and lateral control as in Figure 2.3, which is reasonable if the vehicle does not drive at its limits, i.e., if the lateral accelerations are small. On highways, comfortable maneuvers are desired and hence, the separation is suitable. A survey on control schemes for these tracking controllers can be found in [145].

From the best trajectories over the entire prediction horizon $s(t), l(t), v_s(t)$ for $t \in (0, t_{\text{pred}}]$, a reasonable reference value $s_{\text{ref}} = s(t_{\text{ref}}), l_{\text{ref}} = l(t_{\text{ref}}), v_{\text{ref}} = v_s(t_{\text{ref}})$ has to be found for each time step of the controller. Some low-level tracking controllers require a so-called look-ahead distance s_{ref} several meters in front of the vehicle and the corresponding lateral deviation l_{ref} from a reference path, e.g., the road. For other controllers, the reference value need to be transformed back to (x, y) coordinates by using the local road data. Then, the references in local vehicle coordinates $x_{\text{ref}}, y_{\text{ref}}, \theta_{\text{ref}}, v_{\text{ref}}$ are tracked by these low-level tracking controllers. In the following section, the tracking controllers used in the simulations described in Chapter 3 are briefly discussed.

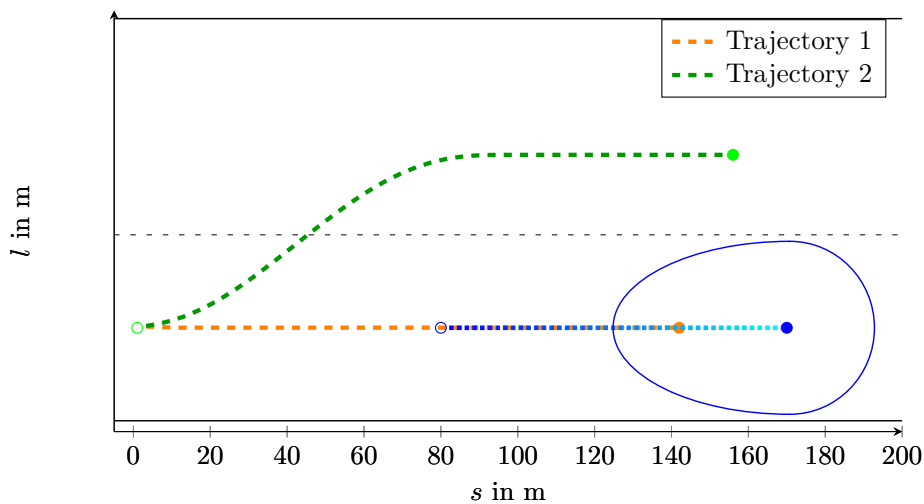


Figure 2.26.: Example with one slow obstacle in front and two generated trajectories for the ego vehicle. The current positions of the vehicles are depicted by open circles, while the predicted end position is depicted by closed circles. The ellipse of the obstacle is shown around the predicted end point of the obstacle.

2. Trajectory Planning for Automated Highway Driving

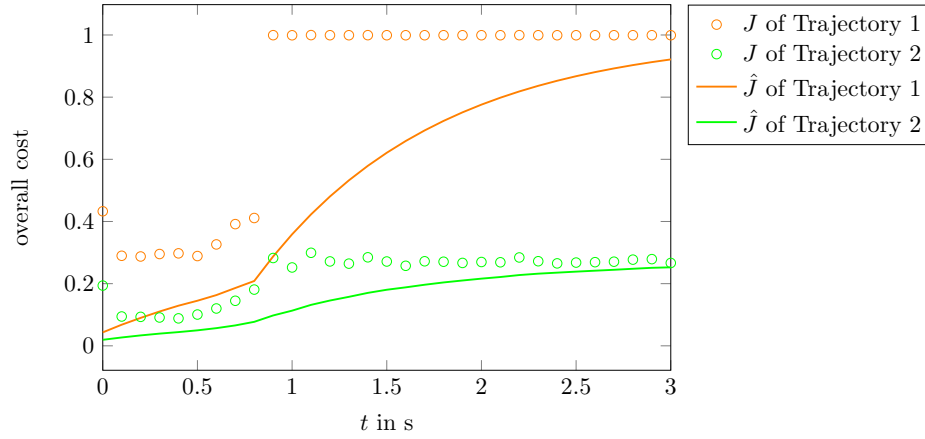


Figure 2.27.: Unfiltered and filtered overall costs of the two trajectories from Figure 2.26.

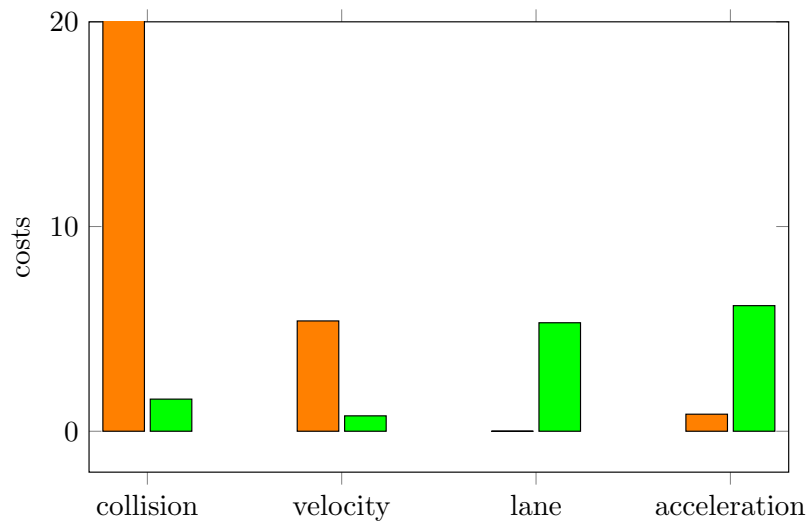


Figure 2.28.: Cost components of the two trajectories from Figure 2.26 at $t = 2s$.

2.3.1. Lateral Control

A simulation study in [145] has shown that different steering controllers perform similarly under the same settings and conditions; hence, a steering controller that has already been implemented for the LKA assistance system at the VIRTUAL VEHICLE Research Center has been used in the simulation [113], [114]. The steering controller executes the computations with sampling time T_{ctrl} and requires the following inputs at time instance k : the look-ahead distance s_{ref} , and the desired lateral offset l_{ref} at the look-ahead distance. The samples of the best trajectories $s(t), l(t)$ are chosen so that the look-ahead distance s_{ref} is approximately 15 meters (depending on the longitudinal velocity), which yields good results in a lane keeping scenario and is therefore also used for general reference tracking maneuvers. Note that too small look-ahead distances can result in oscillating behavior, while large look-ahead distances yield cutting behavior in bends.

The controller incorporates the actual data of the ego vehicle that is assumed to be available:

the lateral deviation, the angular offset with respect to the reference path, actual velocities in longitudinal and lateral direction of the vehicle, yaw rate, steering angle and steering angular velocity. Moreover, the parameters of the vehicle are used in the control law, which is described in [114]. The output of the controller is the steering wheel torque that is applied by the vehicle's actuator.

2.3.2. Longitudinal Control

The longitudinal control used in the simulation is also available from previous projects; the longitudinal controller is the basic ACC functionality that is already available in IPG CarMaker, and has been used for simulation purposes.

The input to the controller is the desired acceleration $a_{x,des}$. For this purpose, the desired acceleration a_s in (2.18) from the planning level can be used. However, if the actuator is affected by actuator dynamics, it cannot be guaranteed that the desired velocity can be reached, since the desired acceleration cannot be tracked perfectly by using feed-forward only. Hence, a proportional mid-level velocity controller is added that computes the desired input to the longitudinal low-level controller as

$$a_{x,des} = a_s + k_v(v - v_{ref}) , \quad (2.60)$$

with parameter k_v , and $v_{ref} = v_s(t_{ref})$ of the planned trajectory in (2.18) with proper choice of t_{ref} , and v the actual velocity of the vehicle.

2.3.3. Choice of Sampling Times

The performance of the MWC depends on the sampling time of the high-level planning. If the sampling time T_{TP} is too large, changes in the environment can remain unnoticed, which can be dangerous. However, the sampling time of the planning level cannot be arbitrarily small, depending on the computational resources. A sampling time of $T_{TP} = 100$ ms has been found to be reasonable in simulation. However, the controllers are executed at a sampling time of $T_{ctrl} \leq T_{TP}$. In the simulations performed in Chapter 3, it has been chosen as $T_{ctrl} = 1$ ms, in the experiments in Chapter 9 as $T_{ctrl} = 10$ ms .

If the output of the planning level changes only every sampling step T_{TP} , while the controller is executed several times, e.g. $T_{ctrl} = T_{TP}/100$, the control inputs will show peaks that appear due to the steps in the reference. The control output of the steering controller is shown in Figure 2.29, and behaves differently when using a reference that either changes step-wise or is extrapolated as in Figure 2.30. These peaks in the control input, however, have to be avoided. Thus, the reference has to be processed such that the controller receives smooth reference signals.

For this purpose, there exist several solutions: first of all, since the polynomial coefficients are known, these coefficients can be forwarded and the corresponding sample of the trajectory can be computed until the end of the trajectory, as depicted in Fig. 2.31. This procedure demands that the coefficients are sent as additional signals to the low-level controller. Moreover, it is assumed that the time instances have to be known. If the time

2. Trajectory Planning for Automated Highway Driving

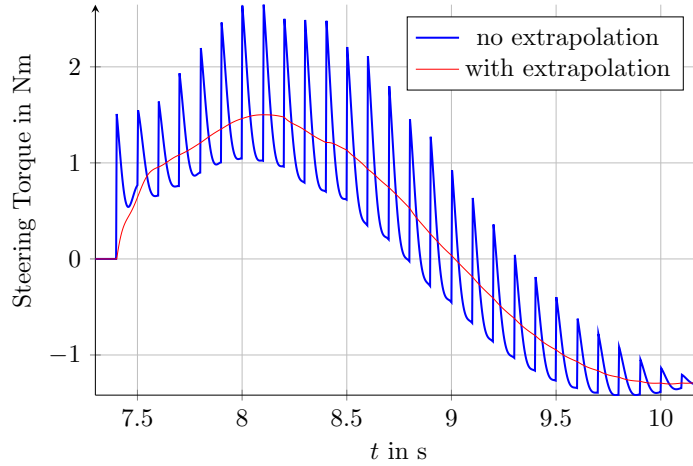


Figure 2.29.: Steering Torque using different sampling times for planning and tracking level.

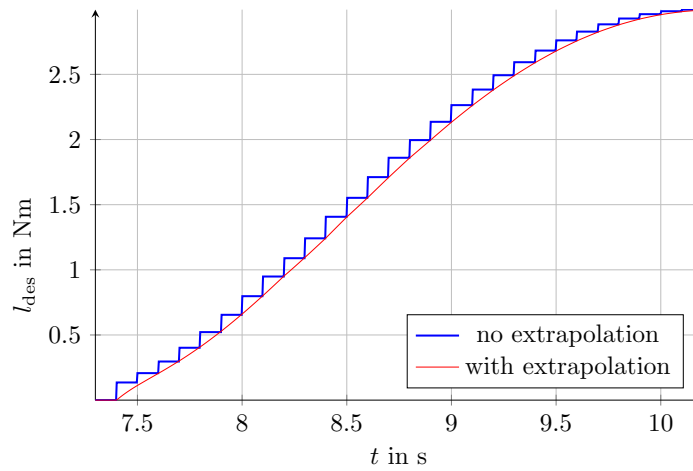


Figure 2.30.: Lateral Offset extrapolated to deal with different sampling times.

exceeds the computed final time of the trajectory, constant values need to be used, which requires also the knowledge of the final time.

Another possibility is to interpolate the lateral deviation based on its derivative, i.e., only $l(k)$ and $v_{\text{lat}}(k)$ are sent to the controller and the lateral deviation $l_{\text{des}}(p)$ with $p = 1, 2, \dots, T_{\text{TP}}/T_{\text{ctrl}}$ is extrapolated with a first order hold as $l_{\text{des}}(p) = l_{\text{des}}(p-1) + T_{\text{ctrl}}v_{\text{lat}}(k)$, and reset to $l_{\text{des}} = l(k)$ when a new sample from the TP arrives. Note that with this first order hold, the reference still exhibits a step at each planning sample step due to this reset. However, due to the extrapolation, the step is smaller than the zero-order hold signal, and the results are much better with the first order hold. In order to obtain smooth signals, a feedback loop is introduced and the output of the reference computation for the controller is computed as

$$\begin{aligned} l_1(p) &= l_1(p-1) + T_{\text{ctrl}}v_{\text{lat}}(k), \\ l_{\text{des}}(p) &= l_{\text{des}}(p-1) + k(l_1 - l_{\text{des}}(p-1))T_{\text{ctrl}}, \end{aligned} \quad (2.61)$$

with $k = 1/T_{\text{TP}}$ in the presented simulations. The reset $l_1(0) = l(k)$ is performed every time

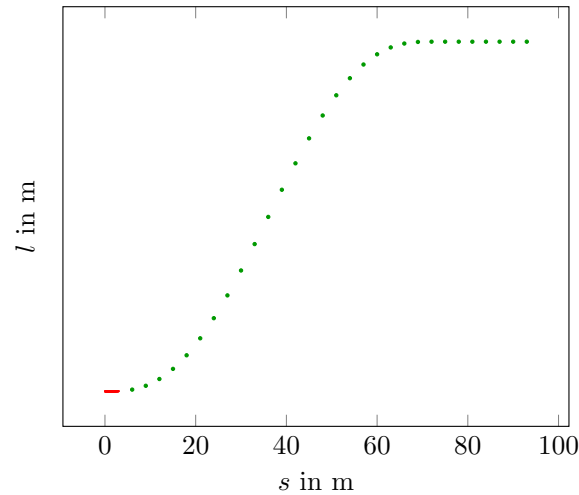


Figure 2.31.: Trajectory tracking: the samples of the reference are shown over the prediction horizon. All samples can easily be computed by knowledge of the polynomial coefficients and the final time. In red, the relevant samples for the controller are shown.

a new reference l arrives, and l_{des} is initialized once but never reset. The results using this approach are shown in Figures 2.29 and 2.30. Due to its easy implementation and its good performance, (2.61) has been used in simulations with MATLAB and IPG CarMaker.

A third way to solve this issue is to store the $m = T_{\text{TP}}/T_{\text{ctrl}}$ samples of the trajectory and forward all samples to the controller. In Fig. 2.31, these samples for the controller are shown in red, while the samples used in the generation and evaluation process of the planning level are shown in green. Then, for each sampling step p of the controller, $p = 1, 2, \dots, T_{\text{TP}}/T_{\text{ctrl}}$, the p -th sample of the trajectory can be used. A comparison to the first order hold approach is shown in Figures 2.32, where the bullets in a) indicate ten sampling steps of the controller level based on the black starting point and its derivative.

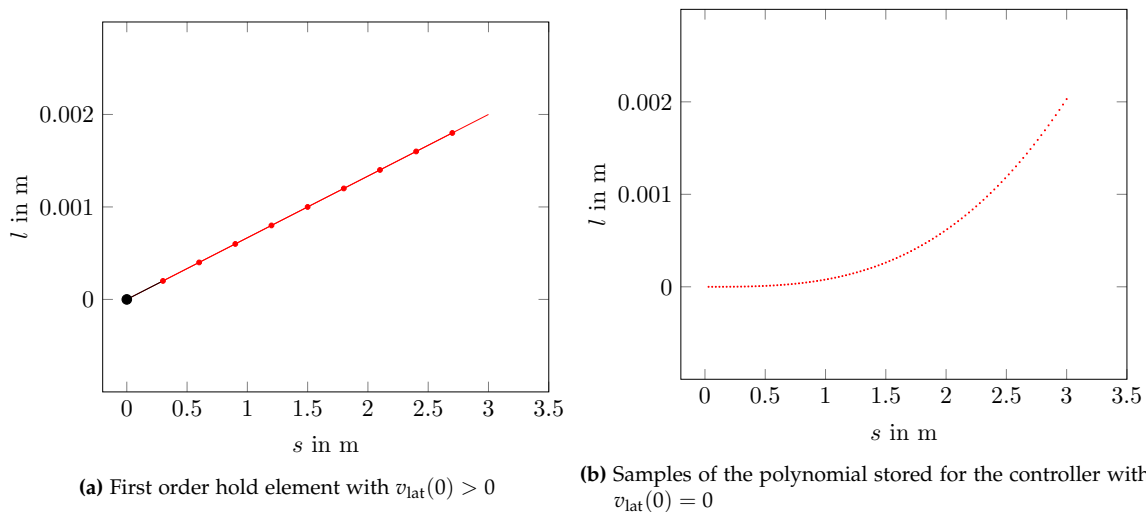


Figure 2.32.: Different approaches to deal with different sampling times of high-level planning and low-level control.

2. Trajectory Planning for Automated Highway Driving

Note, however, that if the vehicle does not track the reference exactly, a time-based use of the samples does not yield the desired results: if the vehicle is not tracking the reference correctly, this can, for example, result in oscillations in the steering angle. Hence, one can use the trajectory samples in a lookup table depending on the actual states of the vehicle. This approach has been used in the experiments on the small-scale vehicles and performs well, see Chapter 9.

3. Simulation Studies and Results

The proposed trajectory planning level and tracking controllers have been simulated in different simulation environments: first, the planning level has been tested in MATLAB/Simulink [5]. Then, MATLAB/Simulink has been used with IPG CarMaker [6] to test the tracking performance and the assistance system "Motorway Chauffeur" including sensor fusion, behavioral planning level and the complex vehicle model of IPG CarMaker. Third, MATLAB/Simulink has been coupled with SUMO [7] and different traffic scenarios with several other vehicles have been tested. In this chapter, the results of these simulation environments are presented.

3.1. MATLAB/Simulink

In the first MATLAB/Simulink studies, the trajectory planning level has been tested without global planning algorithms, i.e., the information from environment model and behavioral planning level are assumed to be available.

The following workarounds have been performed for this purpose:

- environment model: if other obstacles are in the vicinity of the ego vehicle and can be detected by any sensor on the vehicle, the local coordinates are computed. Road data around the vehicle is also known to the vehicle.
- behavioral planning: only a desired velocity for the MWC is defined and no state machine is used. Moreover, flags can be set, e.g., for merging scenarios. Operational constraints in the first simulations are the maximum accelerations and maximum velocities.

Parameter tuning has mainly been performed within this simple setup, and the parameters used in the simulations are listed in Table 3.1.

The following scenarios have been tested in MATLAB/Simulink:

1. Overtaking maneuvers: An obstacle drives slower than the ego vehicle in front on the same lane, and the ego vehicle has to overtake the obstacle in order to keep its desired velocity. In Figure 3.1, the results of the trajectory planning level of the ego vehicle (green) for one slow obstacle in front (blue) are shown for two time instances. The best trajectory of the ego vehicle is plotted as a thick line with a color gradient from current time (green) to the end of the time horizon (red). The blue vehicle's predicted trajectory is depicted with a color gradient from blue to red. The second possible trajectory of the obstacle is plotted as cyan dashed line, which is not selected in the obstacle's prediction, since the vehicles should keep right. The blue ellipse

3. Simulation Studies and Results

Table 3.1.: Parameters of the trajectory planning level that have been used in the simulations.

General Parameters		Cost Parameters	
number trajectories l	2	w_o	3
number trajectories v_s	5	w_l	2
lanewidth w_{lane}	3.75 m	w_{center}	0.5
number lanes	2	w_{merge}	0.1
$a_{s,\text{max}}$	5 m/s ²	w_{velocity}	5
$a_{\text{lat,max}}$	2.5 m/s ²	$w_{\text{vel,dyn}}$	0.1
min. distance long. D_{min}	5 m	$w_{\text{stat,s}}$	0.5
min. distance lat. $D_{l,\text{min}}$	0.5 m	w_{stat}	2
time-headway t_h	1 s	α	15
velocity difference c_v	1 s	β	5
t_{pred}	4.5 s	T_s	0.1 s
N_{pred}	45	τ_f	1 s
N_e	20	k_v	0.8
prediction tolerance off P_{tol}	20	$w_{s,p}$	0
prediction tolerance on $P_{\text{tol},2}$	40	$w_{l,p}$	1

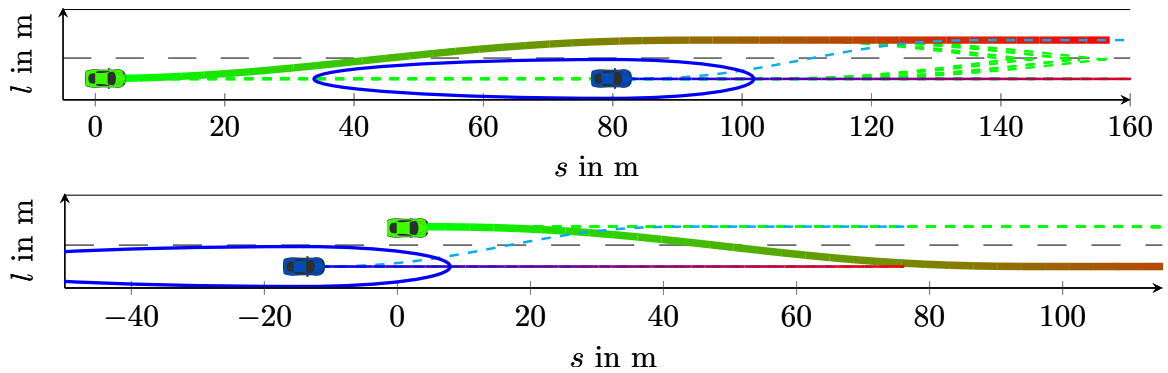
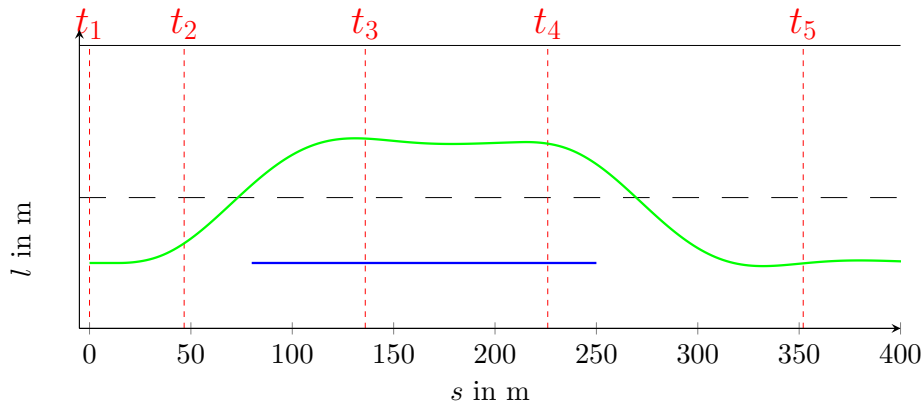


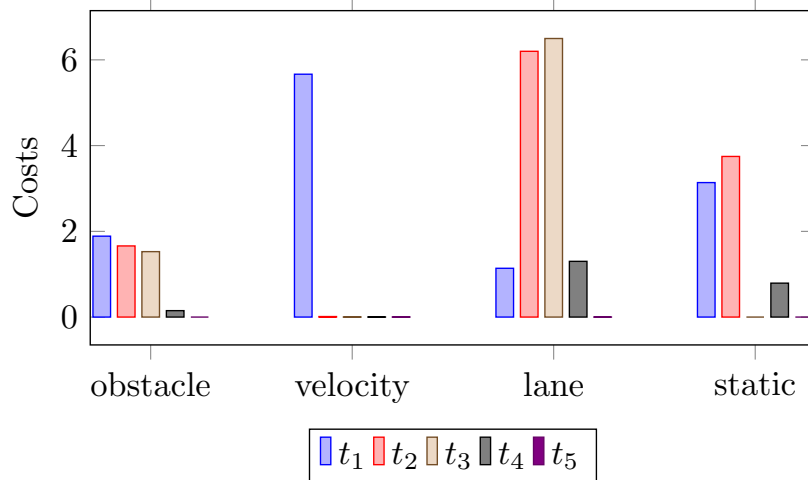
Figure 3.1.: Overtaking maneuver with one slow obstacle (blue) in front of the ego vehicle (green) at two time instances.

indicates the safety distance around the obstacle and is not entered during the whole prediction horizon.

The costs of the best trajectory at different time instances during the overtaking maneuver is shown in Figure 3.2. At $t = t_1$, the ego vehicle has a lower velocity than the desired one, and hence the velocity costs at the first time instance are the highest. At time instance $t = t_2$, the lane change maneuver has already been started and the obstacle costs are hence not very large. However, due to the lateral and longitudinal acceleration, the static costs (2.57) are rather high. Since the desired lane is the right lane, the driving on the left lane yields high lane costs. At time instance $t = t_3$, the lane costs are highest since the vehicle has already left the desired lane and will not change lane during the predicted time window. The next lane change is then triggered at $t = t_4$, where the predicted trajectory is collision-free, at the desired velocity, and will reach the desired lane. Lateral accelerations yield the small static cost. Finally, at



(a) Actual Trajectories of ego vehicle (green) and obstacle (blue).



(b) Cost components of the best trajectory

Figure 3.2.: Overtaking maneuver with cost components at different time instances.

$t = t_5$, all desired values are reached and no accelerations are necessary and thus all costs of the best trajectory are very low.

Moreover, a second obstacle can be added that is closing in from behind the ego vehicle. Depending on the relative distances and velocities, the ego vehicle either has to slow down and let the faster vehicle pass, or accelerate and overtake the first vehicle before the second obstacle. The parameters have been tuned so that the behavior of the ego vehicle is not too conservative.

- Collision avoidance: Two obstacles at standstill are considered. In this example, the ego vehicle cannot decelerate to a standstill safely due to a large velocity and thus has to change lanes in order to avoid collisions. Note that one-cycle planning cannot yield collision-free trajectories for arbitrary time horizons: all trajectories intersect the ellipse of either the left or the right vehicle, see Figure 3.3. Then, the time-horizon of the generated trajectories can be decreased so that at least one trajectory will be collision-free. However, decreasing the time horizon arbitrarily is not safe, since the vehicle cannot react arbitrarily fast. For this purpose, the three-cycle planning is used, which allows to consider more possible trajectories for collision avoidance and faster

3. Simulation Studies and Results

reaction to changes in the environment due to a sufficient time horizon. Simulation results are shown in Figure 3.3, where collision-free trajectories can be found. The ego vehicle then chooses the best trajectory that avoids both obstacles. In addition, the desired velocity can be set to a small value in order to slow the ego vehicle down after the maneuver. However, the desired low velocity has been set to a non-zero value in order to leave the dangerous situation of two vehicles at standstill; approaching vehicles from behind might not be able to stop, and the ego vehicle must not block the path, which is avoided by a low desired velocity. Note that several planning cycles result in more possibilities, but also high computational costs and hence, planning with less cycles should be preferred in standard scenarios.

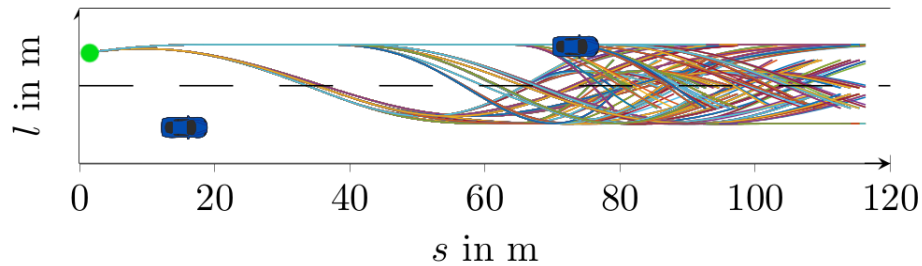


Figure 3.3.: Collision avoidance with three-cycle planning. The two obstacles are depicted by blue vehicles and the ego vehicle by the green dot.

3. Merging scenarios: A certain lane has to be taken in order to follow a specific route, and a corresponding flag is triggered by a higher planning level. Then, the ego vehicle has to drive on the desired lane before a predefined position, e.g., an intersection, is reached. If the traffic flow is low, this scenario is equivalent to a simple lane change. However, if there are many vehicles on the road, merging onto a certain lane might be difficult, since human drivers naturally adapt the distances to the preceding vehicle in dense traffic, but automated cars do not violate their safety distances. In Figure 3.4, the blue vehicle does not keep its safety distance to the red vehicle and hence, the green ego vehicle cannot enter the gap between the two cars. Hence, the cost function of the trajectory set evaluation is slightly adapted: first, the choice of the lane is more important than maintaining a desired velocity. Second, the safety distances to the vehicles on the desired lane are adapted in case of high traffic density. The results of the merging maneuver with adapted safety distances are shown in Figure 3.5, where the green ego vehicle can merge to the desired lane due to smaller inter-vehicle distances. However, it is not clear how far the safety distances can be decreased in order to still guarantee collision-free driving. Note that decreasing the time-headway t_h in the safety distance (2.29) below a certain value can lead to collisions as discussed, e.g., in [93]. In order to avoid collisions, these merging scenarios in presence of many vehicles have been investigated further and are discussed in detail in Part II.

The planning level has shown satisfactory results for simple maneuvers. However, there is one major drawback of the polynomial planning approach: the fifth order polynomials are prone to numerical errors. Hence, the following subsection shows the effects of the numerical error on the planned trajectories.

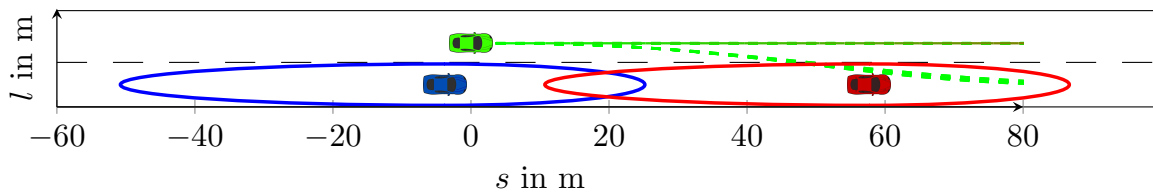


Figure 3.4.: Merging maneuver without adaptation of the safety distance

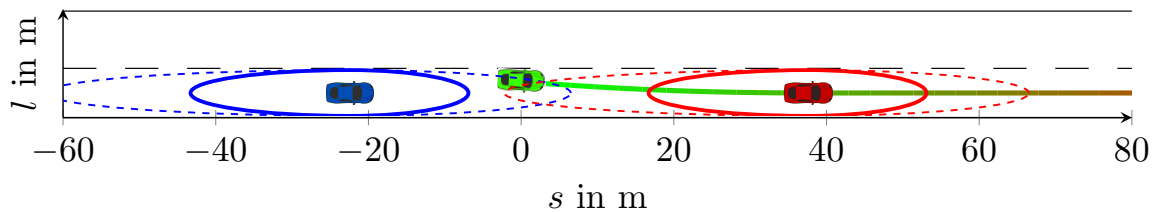


Figure 3.5.: Merging maneuver with adapted ellipses that depend on the velocity of the green ego vehicle.

Numerical Error

Since the planning level uses the reference states from the last planning instance and computes new trajectories at each planning step with sampling time T_{TP} , the polynomials are very often newly computed. However, due to numerical inaccuracy, these computations exhibit an error as shown exemplarily for the lateral deviation and its derivative in Figures 3.6, 3.7, respectively.

However, since the safety distances are spacious and the computational costs with the polynomial approach are fixed, the errors that arise from numerical inaccuracy are tolerated.

3. Simulation Studies and Results

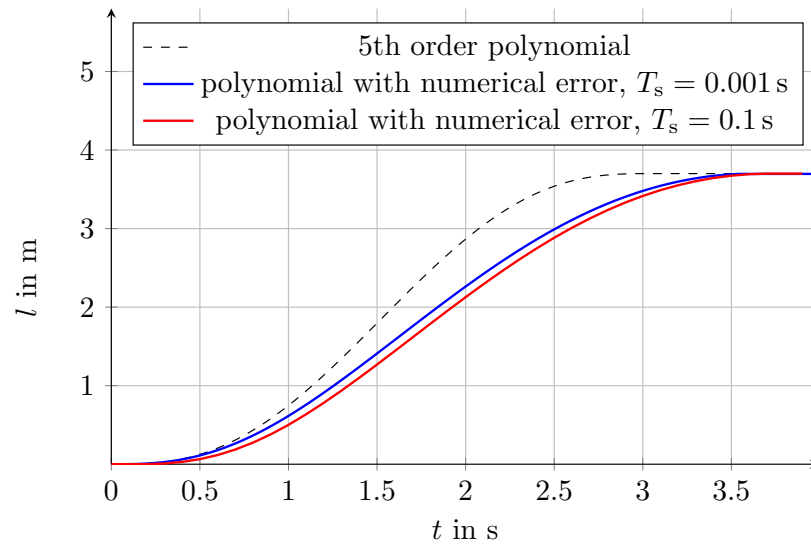


Figure 3.6.: Effect of the numerical error of the lateral deviation when planning is executed from the last time instance.

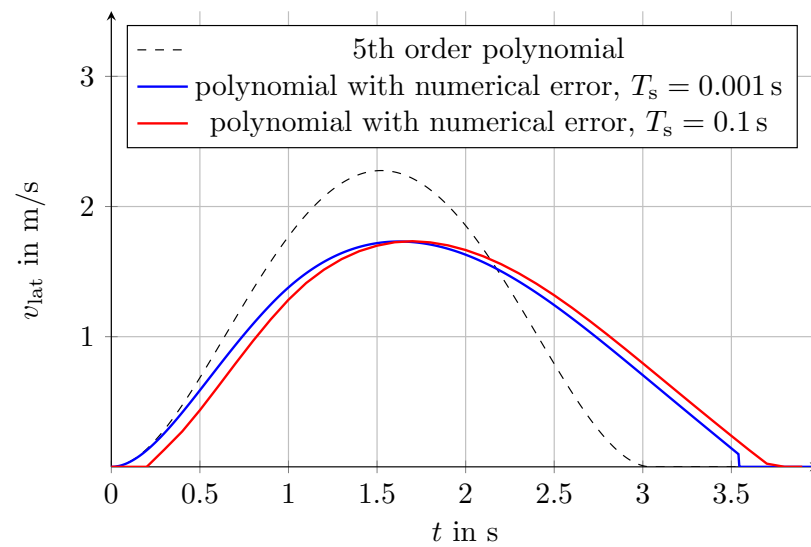


Figure 3.7.: Effect of the numerical error of the lateral velocity when planning is executed from the last time instance.

3.2. IPG CarMaker

After the planning level has been successfully tested in MATLAB/Simulink, IPG CarMaker [6] has been used to extend the vehicle model from a simple point-mass-model to a more complex model. IPG CarMaker represents the vehicles in more details, allows to generate different scenarios, and the results can be illustrated as shown in Figure 3.8.

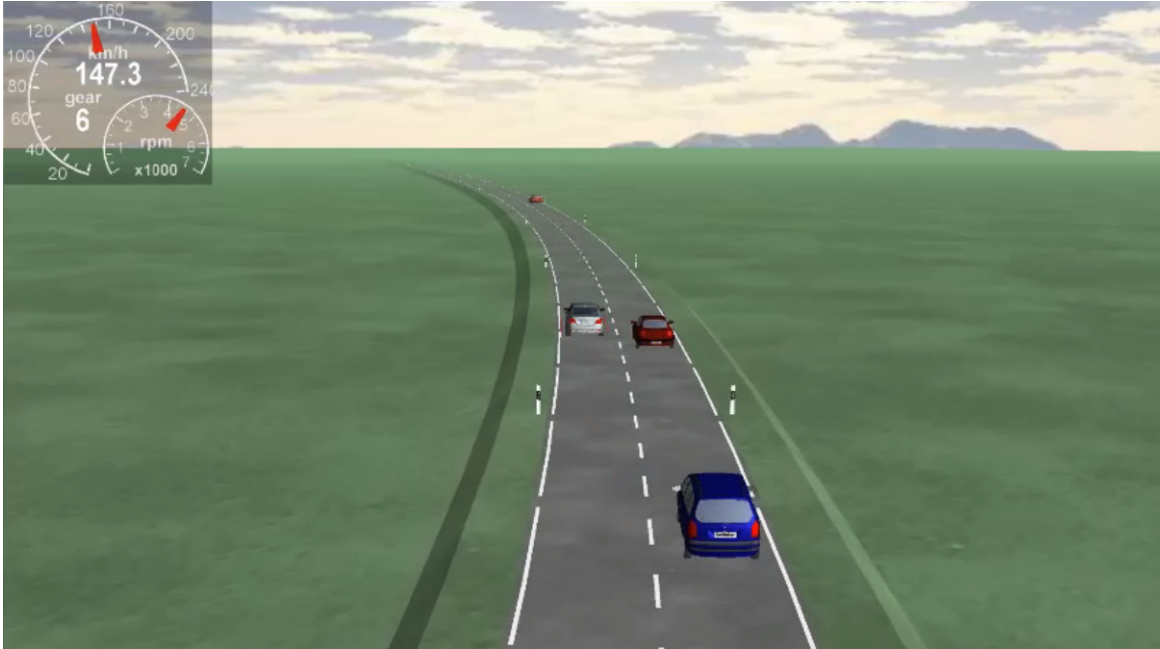


Figure 3.8.: Overtaking scenario simulated with IPG CarMaker

Moreover, the computation of environmental models, see [111], and a state machine for the behavioral planning level have been implemented. The simulations are thus closer to reality and different levels of automation can be tested. The block diagram from a control engineer's point of view is shown in 3.9, where the orange parts have been adapted to the IPG CarMaker environment. In comparison to the closed-loop system in Figure 2.3, other low-level controllers are used and the inputs to the vehicle model are different, since the vehicle model is now provided by IPG CarMaker, where many vehicle parameters can be set. Moreover, the higher-level behavioral planning level that interacts with the driver has been added to the simulation. From a user's point of view, it is possible to switch on/off the different ADAS, such as MWA, MWA+, and MWC, and set desired values such as a desired velocity. The MWA+ is an extension of the MWA, where lane changes are recommended to the driver. If the driver accepts the lane change, the ADAS executes the maneuver, otherwise the MWA is active.

In contrast to the MATLAB/Simulink simulation in the previous section, a complex vehicle model and appropriate tracking controllers are used, where the controllers have a different sampling time than the planning level. In the simulation with IPG CarMaker, the extrapolation method described in Section 2.3.3 has been implemented, which improves the performance significantly. The goal of the simulation with IPG CarMaker was to evaluate

3. Simulation Studies and Results

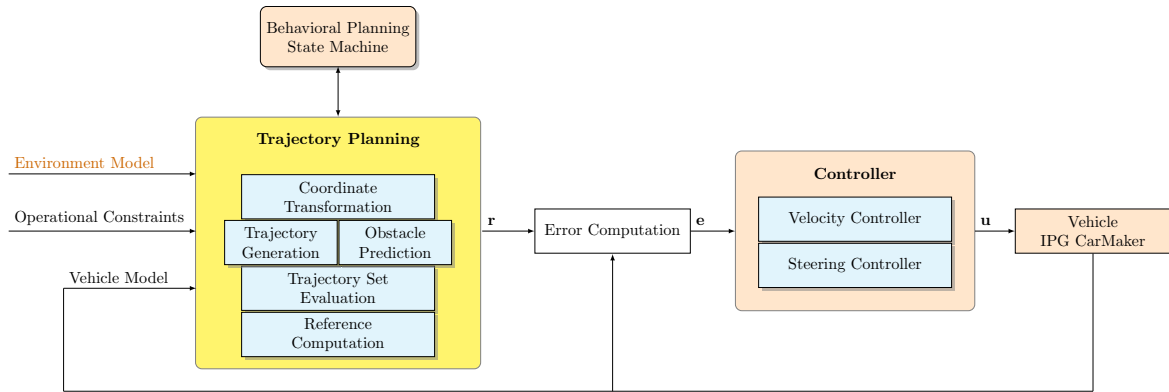


Figure 3.9.: Schematic of the closed-loop system for simulations with MATLAB and IPG CarMaker

the performance of the trajectory planning level of the different ADAS in a more detailed environment.

3.2.1. Motorway Assist

The Motorway Assist is an assistance system that keeps the vehicle on the current lane and adapts the velocity according to the ACC. In a first step, the MWA has been realized by combining ACC and LKA. However, the TP level in combination with the tracking controller can achieve similar or better performance, depending on the parameters of the TP.

The trajectory planning level has been used with the following settings:

- the endpoints of the lateral offset are only on the current lane,
- the velocity endpoints have been used as defined in (2.24), but can be extended to improve performance.

Note that if all trajectories result in collisions, the behavioral planning level can demand a collision avoidance maneuver by allowing more endpoints in lateral offset. In the following discussions, the standard MWA has been tested with five different velocity profiles on one lane.

The performance of the MWA is shown in Figure 3.10, where the results are compared to a standard ACC that is provided by IPG CarMaker. It can be seen that the parameters of the ACC are set differently, i.e., the ACC is reducing the distance slowly. The MWA has been tuned to imitate a human driver, closing in faster and then slowing down in a “sporty” way. The parameters can of course be tuned so that the planning level decreases the velocity more slowly, similar to the ACC behavior. For this purpose, the maximum acceleration of the set generation can be tuned.

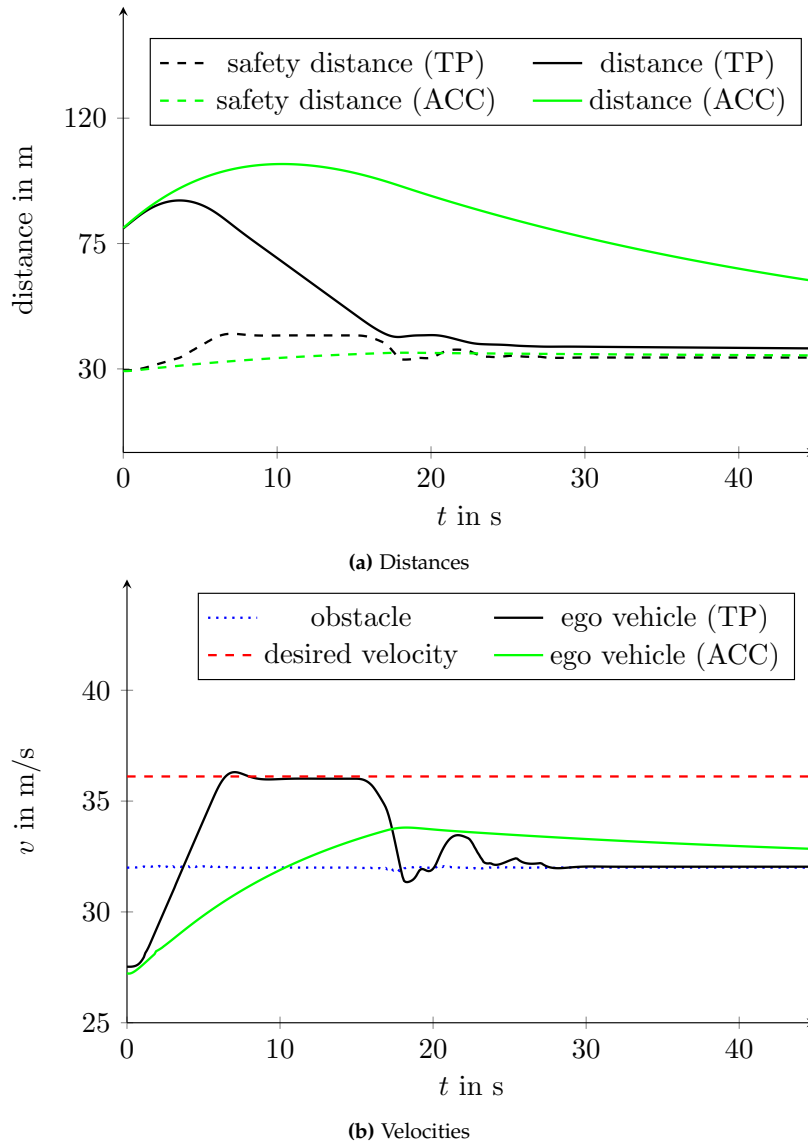


Figure 3.10.: The ACC and LKA functionalities have been replaced by the trajectory planning and tracking controller. Vehicle-following is possible when restricting the vehicle to one lane. The safety distances are not violated.

3.2.2. Motorway Assist Plus

The Motorway Assist Plus is an extension of the MWA: in addition to driving on one lane, a lane change recommendation is sent to the driver. If the driver accepts, e.g., by activating the turn signal, the lane change is performed by the system. If the driver does not react, no lane change is executed and instead, the velocity is adapted as in the MWA. For the lane change recommendation, an additional ellipse around the preceding obstacle is added: if the vehicle enters this additional larger ellipse, a recommendation is triggered. Since the axes of the ellipse depend on the velocity of the vehicle as in (2.29), (2.30), the parameter t_h can be set to a larger value, e.g., $t_{h,r} = 3$ s. Then, depending on the relative velocity

3. Simulation Studies and Results

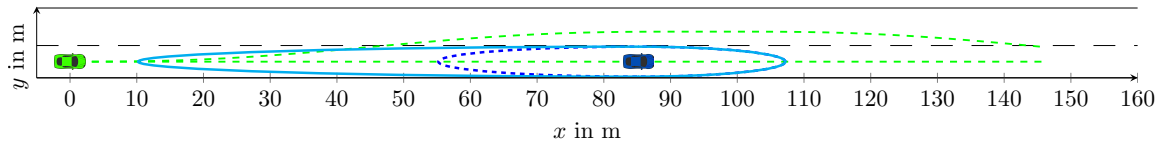


Figure 3.11.: Additional ellipses are used for a lane change recommendation of the MWA+

of the vehicles $\Delta v_i = v_{\text{ego}} - v_{\text{obst},i}$, the driver has $(t_{h,r} - t_h)v_{\text{ego}}/\Delta v_i$ seconds to accept or decline the lane change. If the driver does not accept the recommendation and the vehicle's predicted trajectory enters the smaller ellipse t_h , the vehicle has to slow down and adapt the velocity to the preceding vehicle according to the MWA. In Figure 3.11, the two ellipses around the preceding vehicle are shown. Once the lane change is accepted by the driver, the best trajectory to the respective lane is computed and tracked by the vehicle. Aside from the recommendation, the results of the MWA+ are equivalent to the MWC results.

3.2.3. Motorway Chauffeur

The Motorway Chauffeur is a system that does not require the driver to be active all the time, i.e., the system is capable of driving highly automated on a highway and the vehicle has to make various decisions on its own. However, the MWC can request a take-over from the driver within ten seconds if difficult situations occur.

Standard driving scenarios can be handled very well by the MWC and the scenarios tested in MATLAB in Section 3.1 have been successfully implemented also with IPG CarMaker. Different passing maneuvers, velocity adaptations, collision avoidance scenarios and merging maneuvers have been tested and the results are satisfactory. Since the results of the planning level have already been discussed in Section 3.1, the additional results of the tracking performance during a lane change is shown in Figure 3.12, with the corresponding lateral acceleration in Figure 3.13.

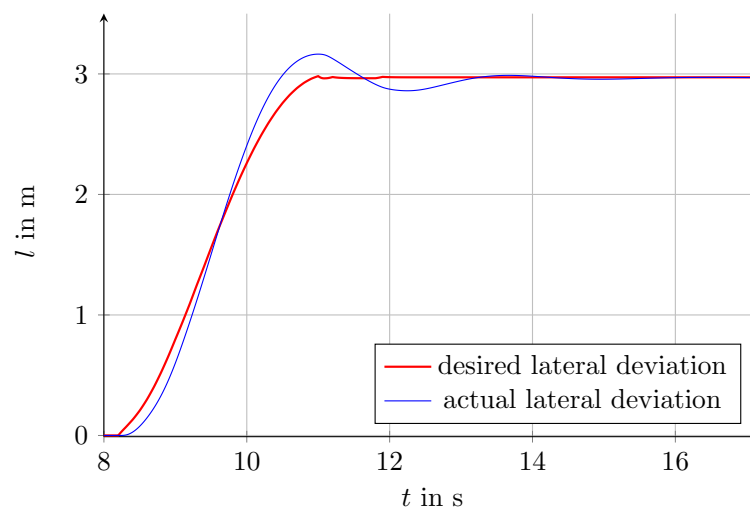


Figure 3.12.: Tracking performance of the vehicle in a lane change maneuver using the MWC

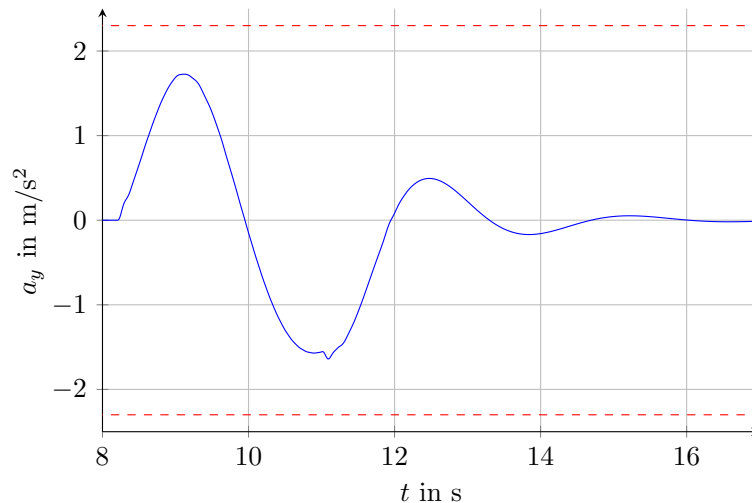


Figure 3.13.: Lateral acceleration of the vehicle in a lane change maneuver in the vehicle's local coordinates. The red dashed line indicates the bound for comfortable driving maneuvers.

Since IPG CarMaker 4.0 has been used, only two lanes have been considered. Moreover, the maneuvers of the other vehicles are difficult to generate or control. Hence, in the next section, the simulation tool Simulation of Urban MObility [7] has been used to evaluate the performance of the planning level on a highway with more lanes and multiple vehicles that can change lanes and adapt their velocities using already implemented driver models.

3.3. SUMO

The trajectory planning level has been tested in MATLAB/Simulink that has been coupled with Simulation of Urban MObility (SUMO) [7]. With this additional software tool, many vehicles can be simulated and driver models for the other vehicles can be used. In MATLAB, the references for the ego vehicle are computed based on the information of the other vehicles, which is provided by SUMO. Note that in these simulations, the tracking controllers are not considered; only the planning level in presence of many other obstacles is evaluated.

For this purpose, several vehicles have been generated and one vehicle has been controlled by the planning level, i.e., the desired velocity and the desired lane is computed by MATLAB and then used in SUMO. The coupling between these two software tools is shown in Figure 3.14.

Basic highway scenarios have been simulated. The ego vehicle has to overtake several vehicles that drive slowly on the right lane in order to maintain the desired velocity. After some time, however, a merging flag is set and the ego vehicle has to take the right lane, where other vehicles are driving. As discussed for merging scenarios in Section 3.1, the safety distance parameters are changed in order to allow merging to the right lane. In Figures 3.16, 3.18, 3.20, 3.21, 3.23, 3.24, 3.26, the SUMO environment with the ego vehicle (red) and several obstacles are shown at different time instances. The planned trajectories

3. Simulation Studies and Results

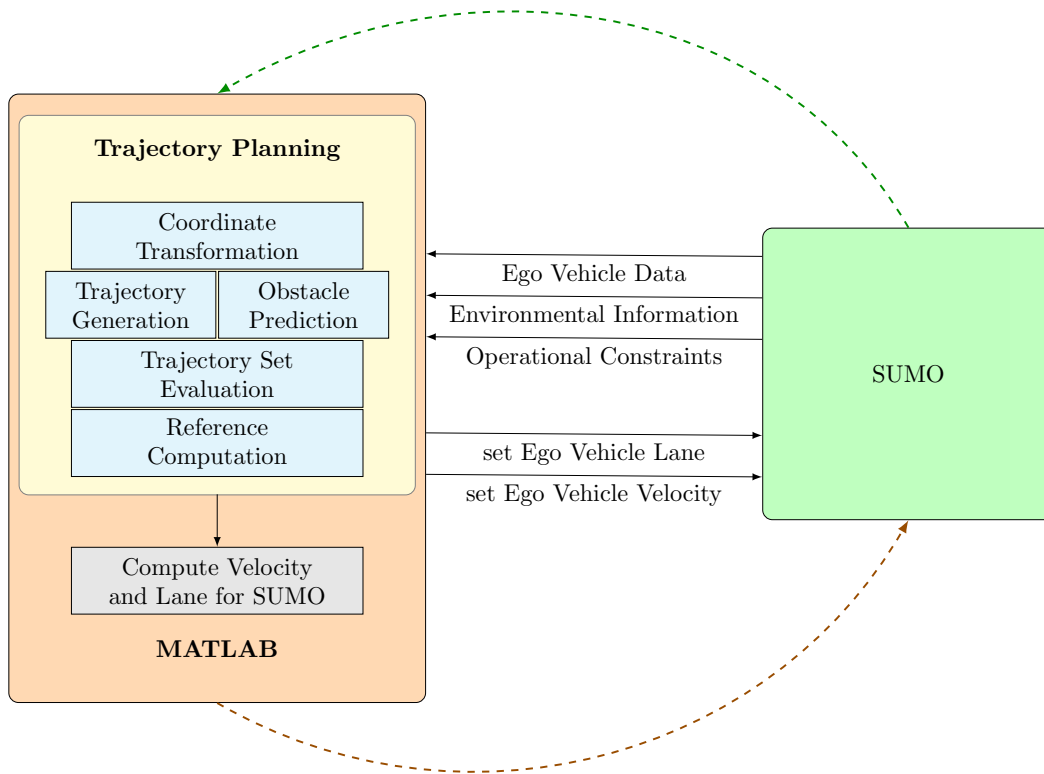


Figure 3.14.: Coupling between MATLAB and SUMO

at the corresponding time instance are depicted in Figures 3.15, 3.17, 3.19, 3.22, 3.25, where the set of trajectories is green and the best trajectory is red. The obstacles are displayed by blue circles.

The tests show satisfactory behavior, if the traffic volume is low. However, the automated driving assistance systems should assist the driver especially in the case of high traffic volume, which is not possible with the current implementation, and the individual planning level has to be extended. Cooperative driving maneuvers are very important to increase the traffic performances, which is discussed in the second part of this work.

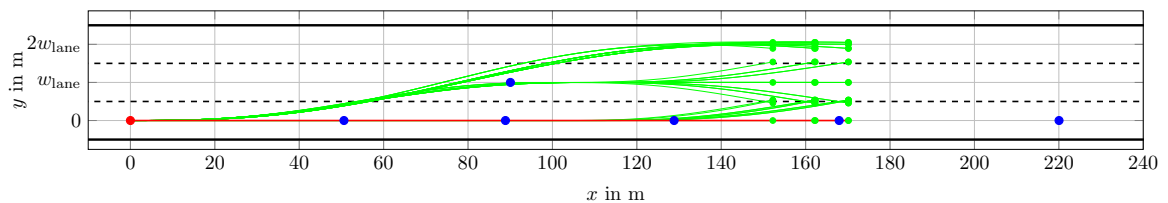
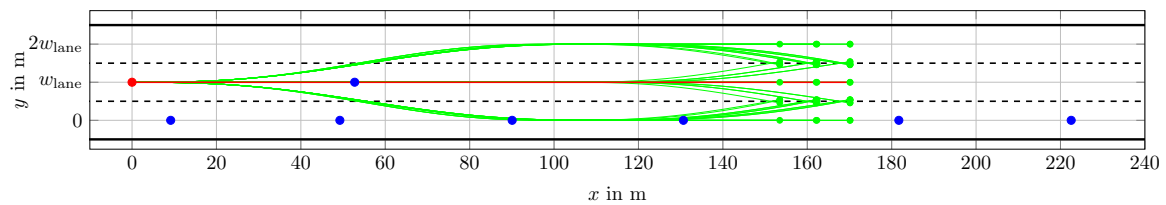
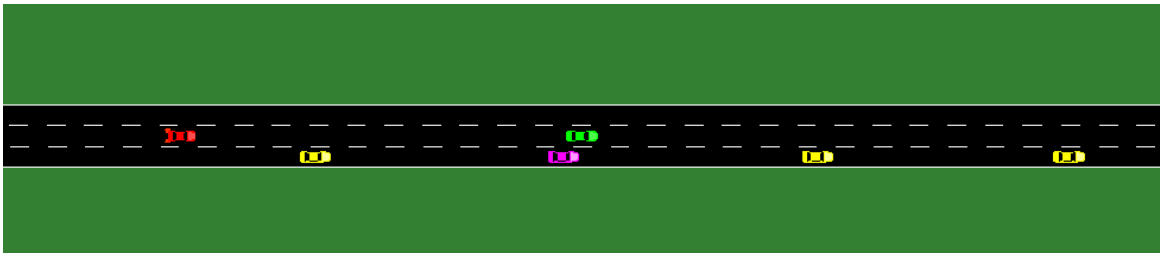
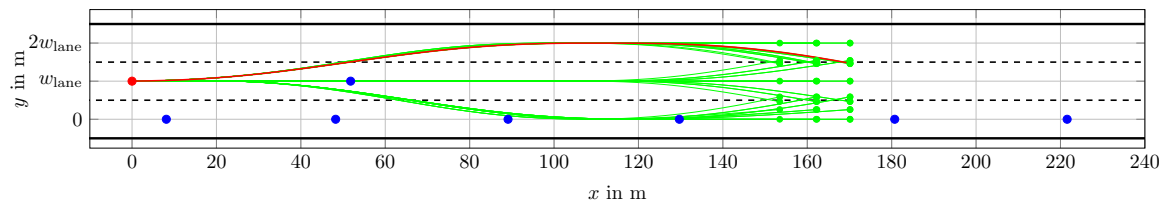
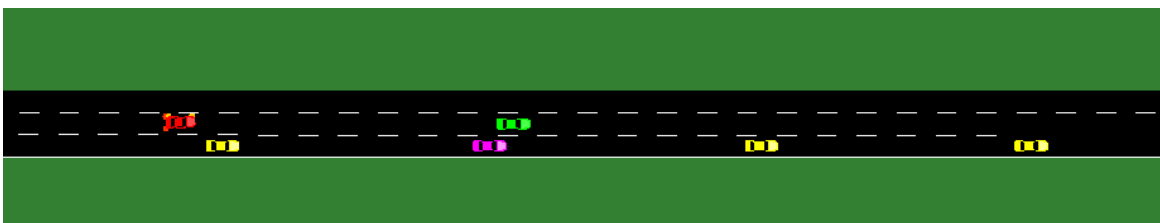


Figure 3.15.: Generated trajectories in MATLAB and the environment from SUMO at t_1 . The ego vehicle is depicted by the red dot, other traffic participants by blue dots, trajectories by green lines and their end points by green dots.

Figure 3.16.: Simulation studies of the MWC visualized in SUMO at t_1 (ego vehicle in red)Figure 3.17.: Generated trajectories in MATLAB and the environment from SUMO at t_2 (ego vehicle in red, obstacles in blue)Figure 3.18.: Simulation studies of the MWC visualized in SUMO at t_2 (ego vehicle in red)Figure 3.19.: Generated trajectories in MATLAB and the environment from SUMO at t_3 (ego vehicle in red, obstacles in blue)Figure 3.20.: Simulation studies of the MWC visualized in SUMO at t_3 (ego vehicle in red)

3. Simulation Studies and Results



Figure 3.21.: Simulation studies of the MWC visualized in SUMO after t_3 (ego vehicle in red)

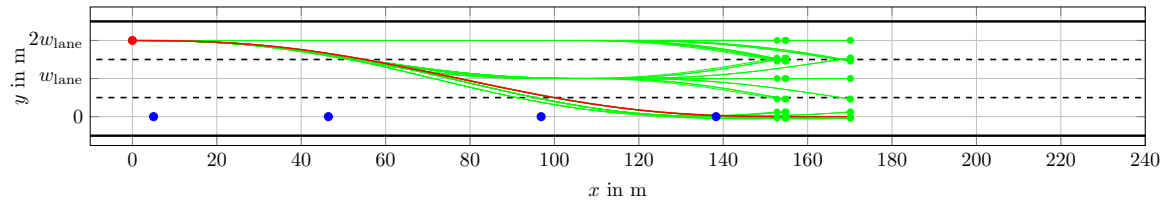


Figure 3.22.: Generated trajectories in MATLAB and the environment from SUMO at t_4 (ego vehicle in red, obstacles in blue)



Figure 3.23.: Simulation studies of the MWC visualized in SUMO after t_4 (ego vehicle in red)

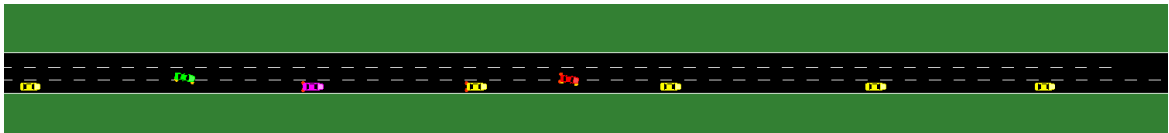


Figure 3.24.: Simulation studies of the MWC visualized in SUMO between t_4 and t_5 (ego vehicle in red)

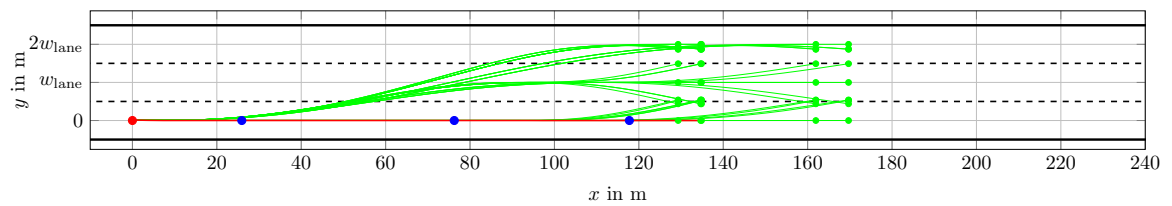


Figure 3.25.: Generated trajectories in MATLAB and the environment from SUMO at t_5 (ego vehicle in red)

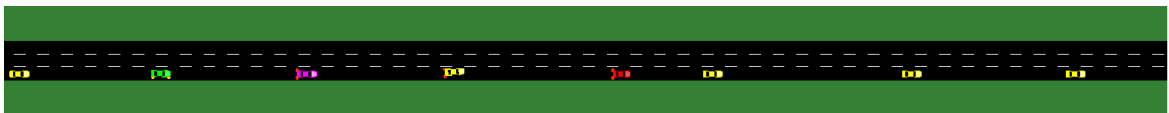


Figure 3.26.: Simulation studies of the MWC visualized in SUMO at t_5 (ego vehicle in red)

Part II.

Cooperative Automated Driving

4. Literature Review on Cooperative Automated Driving

The simulation studies of the different scenarios using the planning algorithm for individual vehicles presented in Part I have demonstrated the limits of non-cooperative driving:

1. Merging scenarios are difficult to handle, especially if the traffic density is high.
2. Large inter-vehicle distances cannot be maintained in practice due to human drivers cutting in.
3. The prediction of other traffic participants is uncertain.

In order to circumvent these drawbacks, cooperative driving with or without communication has been introduced. If Car-to-Car communication (C2C), also known as Vehicle-to-Vehicle communication (V2V), between the vehicles is used, vehicles can send their intentions or other information in order to improve the prediction step. Moreover, inter-vehicle distances can be safely decreased, yielding improved merging performances and increased traffic throughput at intersections. However, communication imperfections also raise several issues [38]: for example, time-delays and packet dropouts can seriously degrade the performance, and security in the data exchange must be ensured, see also [76].

Cooperative driving has been investigated for many decades, starting with the California PATH Program [22], which was founded in 1986. Therein, different research projects have been treated and are investigated also presently, dealing with, e.g., connected and automated vehicles including truck platooning and assisting systems, traffic control and analysis, or intersection collision avoidance systems. The focus lies primarily on cooperation between the vehicles and the infrastructure in order to improve traffic flow, reduce fuel consumption and, most important, increase safety. Currently, cooperation between vehicles (C2C) and the infrastructure (C2X), also known as Vehicle-to-infrastructure (V2I), or Vehicle-to-everything (V2X) is a very popular research field, see, e.g., the projects SARTRE [23] from 2009-2012, Ko-HAF [26], which started in 2015, or the German “Schwerpunktprogramm 1835” [24] from 2014 on, which focuses on cooperatively interacting automobiles. The Grand Cooperative Driving Challenges (GCDC) in 2011 [98] and 2016 [57] have also advanced this progress in the last years. In Europe, a truck platooning challenge (ETPC) was conducted in 2016 including several countries [25]. Both the GCDC and the ETPC have focused on platooning, i.e., vehicle-following, which has been of great interest lately.

Cooperating vehicles can be regarded as a network of multiple agents with a certain goal that depends on other agents. Platooning can be formulated as a one-dimensional consensus problem. Hence, a short literature review on consensus problems is given in the following section.

4.1. Consensus Control

In consensus problem statements, a network of so-called agents considering neighboring or connected entities is investigated. The main objective of a consensus problem is to reach an agreement on certain quantities of interest based on local interaction rules [137]. Initially, consensus problem formulations have been used for synchronization of agents, or for the rendezvous problem using multiple robots, as summarized in [118]. Today, various applications have been found for multi-agent systems, including energy storage systems in wind farms [31], reactive power sharing in microgrids [151], optimization problems [86], and formation control for different autonomous vehicles, such as underwater vehicles [54], ships [88], spacecrafts [135], or automated ground vehicles [122]. In autonomous driving scenarios, each vehicle can be considered as one agent in the network.

Basic consensus problem formulations can be found in [61], [85], [134], [136], or [137]. The dynamics of the agents are often modeled as single or double integrators, see, e.g., [45], [134]. This work focuses on double integrator dynamics, where the states of one agent correspond to position and velocity of the respective agent. In order to track a reference trajectory, e.g., a leading vehicle with a time-varying reference acceleration, a control law for double integrator dynamics with bounded control inputs has been proposed in [134]. For the leader-follower problem therein, the general idea is to use a group reference velocity that is communicated to all or a fraction of the following agents. In many applications, the network of the agents is not fixed, i.e., the connections between the agents may change. These so-called switching topologies have been discussed, e.g., in [118].

Due to their robust performance with respect to certain classes of disturbances or uncertainties, sliding mode based control techniques have been presented for consensus problems where linear controllers fail. Terminal sliding mode control [92], the twisting algorithm [126], using a discontinuous extension [69], and integral sliding mode control [127] have been shown to reach consensus for perturbed systems.

Formation control is an extension of the consensus problem statement: instead of reaching a common position, a desired distance is maintained between the agents. The controllers of the consensus problem statement can thus be used for formation control as well. A detailed survey of multi-agent formation control can be found, e.g., in [117]. Note that the control algorithms typically do not consider obstacle avoidance. Extensions in that regard have to be formulated separately.

4.2. Formation Control of Robots

The formation can be defined in multiple dimensions, which requires to take into account more complex agents' dynamics. In [49], a vision-based formation control framework has been published, where a leader-follower controller capable of avoiding obstacles is presented. The two-dimensional formation of the robots may change according to switching rules in order to avoid collisions with static obstacles. The robots that follow the leader can choose between an in-line formation or a triangular shaped formation depending on the surrounding obstacles.

Another switching topology using triangular and in-line formation has been proposed in [46]. The switching condition depends only on time, which limits the application of this approach for highway driving. Although a highway bottleneck is the main focus of discussion, the proposed formation of the robots is of triangular form, which is not reasonable on highways.

Sliding mode control has been used in [51] for formation control of robots, where a fixed formation based on the distance and the angle between the robots is used. The robots maintain a constant euclidean distance and also consider the bearing angle between the robots. Note that this procedure is typical for robotic formation control, but may not be desirable in automated driving scenarios; instead of fixed euclidean distances and angles, formations should consider the course of the road.

In [97], a testbed for a network of agents has been proposed, where the agents consist of aerial vehicles, i.e., drones, and small-scale trucks. This testbed gave rise to the testbed which will be presented in Chapter 8, where automated driving scenarios such as platooning can be tested easily and at low cost.

Another approach has been discussed in [109], where a so-called virtual structure that resembles the reference point for all agents is used for formation control. Agents can enter or leave the topology, while the remaining agents are not necessarily affected. However, the virtual center is required to collect all data of the agents and communication is assumed to be available.

All mentioned approaches arising from robotics, however, are not applicable on highways without modification: the topologies can switch depending on the current situation instead of time. Moreover, the number of agents can be very large and the impact on other vehicles must be analyzed for real driving scenarios. Note that in robotics, formation control has been considered for multiple agents that have to avoid an obstacle while maintaining a fixed distance to the neighboring agents. However, while obstacle avoidance is a very important aspect, the effect of the formation control in presence of many robots is not considered. Thus, the effect on traffic throughput or error amplifications is not analyzed in classical formation control publications.

Platooning has been described as a one-dimensional formation control problem for multi-agent systems, e.g., in [149], [175]. However, the standard consensus problem statement does not consider all necessary conditions for collision-free platooning; position overshoot and error amplification are typically not considered. In order to analyze the behavior of a string of vehicles in a platoon, the so-called string stability has been introduced, which will be discussed in Chapters 5 for linear controllers and in Chapter 6 for sliding mode controllers. Existing approaches on platooning do not solve all issues, which is summarized in the next section.

4.3. Platooning

Platooning is said to be the first cooperative automated driving system in mixed traffic, and the advantages are increased traffic flow, reduced fuel consumption due to reduced air drag, safety, and driver relief [165]. Hence, the focus of the Grand Cooperative Driving

4. Literature Review on Cooperative Automated Driving

Challenge (GCDC) in 2011 and 2016 has been on the application of platooning in different scenarios, e.g., standard platooning in 2011 [98] and merging of two platoons onto one lane in 2016 [57]. Communication between the vehicles has been a major part of these challenges. Details on the interaction protocols for platoon merging that have been developed for GCDC 2016 can be found in [40]. In [94], a truck from KTH Stockholm is described that has been used in the GCDC 2016, where model predictive control (MPC) has been used for longitudinal control. In [43], MPC has been used as well, where the focus is on the formation shape and also splitting and rejoining from a platoon. Daimler AG's implementation of platooning can be found in [56], where the trucks participated in the EU Truck Platoon Challenge [25].

However, the experiments mentioned do not focus on the theoretical aspects such as string stability or robustness with respect to disturbances. In the former aspect, position errors or accelerations of one vehicle should not amplify along the string of vehicles, which has been discussed, e.g., in [59], [157]. There are typically two challenges: first, collisions can occur when linear controllers are used with a constant distance spacing policy, and the platoon is string unstable, even if the initial errors are zero. To ensure safety, the spacing is typically increased by a velocity-dependent term, which is called constant time-headway spacing [58], where the time-headway parameter has to be chosen carefully as discussed in [93]. However, this leads to large inter-vehicle spacings, which might not be desirable; one of the main advantages of platooning is the fuel consumption reduction due to small inter-vehicle distances, which reduce the air drag of the vehicles. In order to decrease these distances between the vehicles, communication between the cars has been used, e.g., in [102], [112], [121], [174], [177], and also in the PATH program [165]. The second challenge is how to safely reach the desired distance, since positions overshoots lead to collisions that must be avoided. Standard planning procedures can be used to reach the platoon, e.g., model predictive control in [87]. In the latter aspect, the performance has to be robust with respect to external disturbances, which is hard to guarantee for model predictive approaches. Instead, the robust sliding mode based control approaches are discussed in this work.

A first order sliding mode controller is used in [55] for vehicle-following. Different sliding surfaces are used in order to avoid too large accelerations. However, the assumption that the velocity of the predecessor is constant is very limiting for platooning applications, and string stability has not been considered.

In [66], a sliding mode based approach for platooning has been presented using the constant time-headway policy, and string stability has been proven. However, the assumption that the initial errors in position and velocity are zero is very restrictive. In practical applications, a robust method to reach the formation has to be found before the proposed controller can be used. In order to guarantee string stability for a certain range of non-zero initial errors, bidirectional control, which incorporates additional information about the following vehicle, has been used in [77] and [96].

Since string stability for unidirectional control without communication has not been considered so far, an adaptive approach using sliding mode techniques for platooning has been developed, which has been published in [143], and details can be found in Chapter 6. As a next step, the platooning scenarios can be extended to merging scenarios with multiple

lanes as in the GCDC 2016. Several approaches for safe and efficient merging scenarios have been presented in literature, which are summarized in the following section.

4.4. Cooperative Merging

There are two different research areas that focus both on increasing the traffic flow in merging scenarios: on a macroscopic level, traffic management systems with a centralized controller are used that act on traffic lights and the infrastructure. On a microscopic level, i.e., from an individual vehicle's point of view, decentralized cooperation between vehicles can be used to increase the traffic throughput, e.g., by reducing inter-vehicle distances. One of the most common scenarios is merging at lane reductions, e.g., at a highway ramp, where the location is usually known in advance due to a fixed infrastructure, or at construction sites, which can vary in time and location. Thus, it is reasonable to use centralized approaches for highway ramp metering and decentralized approaches for locally varying lane reductions.

Centralized Approaches for Traffic Control and Ramp Metering

In the famous PATH program, highway entries have been analyzed, e.g., in [78]. A vehicle sorting strategy is therein proposed with the aim to increase traffic throughput by maximizing the distance that platoons stay intact using the destination of each vehicle, which requires information of all involved vehicles at the central control unit.

In [36], a ramp metering approach via traffic lights to regulate on-flow traffic is proposed. Different control methods are discussed for ramp metering, including feedback control, model predictive control, and artificial intelligence. These methods are compared with respect to computational complexity, inclusion of hard constraints or predictions in the control design, model-based design, and scalability. Different control frameworks and projects are discussed (e.g. PATH, Dolphin, Auto21 CDS, CVIS, SafeSpot, PREVENT, see references therein), where the open issues are pointed out as the choice of the platoon formation and the scalability.

Simulation tests for lane reduction merging scenarios have been performed with the tool MOTUS in [47], where driver models for longitudinal and lateral motions are used. V2X communication has been assumed to be available and single-lane on-ramps have been considered, with all vehicles cruising at the same speed. The proposed assist then merges between a certain rear and front vehicle, where both vehicles have to agree to the merge request and create a space for the merging vehicle. This merging assist is implemented on the decision level by using state diagrams and assuming perfect communication between the vehicles. However, robustness cannot be ensured, and communication imperfections can degrade the performance of the assistance system.

On-ramp merging has also been discussed in [104], using a slot-based approach for a traffic management system. An evaluation of so-called "slot models" has been performed in VISSIM, where the models have been generated via a central Road Side Unit (RSU)

4. Literature Review on Cooperative Automated Driving

based on a hierarchical approach using V2X, or via group communication and coordination, which is a distributed approach that requires V2V.

In [50], a random mixture of connected vehicles in the traffic has been analyzed and it has been shown that if 40% of the vehicles are connected near bottlenecks, the traffic flow in these bottlenecks can be improved by 52%. For this improvement, the speed is reduced in dedicated slow-down regions, and communication to the other vehicles is required. If a congestion is building up and detected, the information is communicated to other vehicles, and a slow-down velocity is computed as the minimum over all velocities of the connected vehicles. The traffic flow, the average velocity, the effects of communication, and the impact of using ACC with different percentages of connected vehicles have been analyzed and the results have been summarized. However, no cooperative merging algorithm is proposed.

Receding horizon control for autonomous or cooperative car-following control has been used in [171]. A cost function of non-quadratic form for a centralized scheme for a platoon of ten vehicles is proposed. The authors argue that it is still challenging to implement receding horizon controllers for autonomous and cooperative vehicle systems in large-scale traffic simulations.

Another centralized approach for cooperative merging at highway on-ramps has been proposed in [139], which aims at reducing fuel consumption while guaranteeing collision avoidance. The same authors summarized the coordination of connected vehicles at highway on-ramps using centralized approaches in a survey [140].

A cooperative merging assistant called CoopMA has been presented in [150] to reduce congestion at motorway junctions. The main cause for traffic jam at bottlenecks is the difference in speed as described in [99]. Hence, a single platoon is approaching on the ramp due to appropriately controlled traffic lights, and one platoon per cycle can merge onto the highway. A cooperative vehicle on the main lane is assumed to reduce the speed such that gaps are created for the platoon on the ramp. The travel time from traffic light to merge location is assumed to be known, which is reasonable on highway entries, and V2I is required to send information to the leader of the platoon. The evaluation is based on reduction of the occurrence of congestion at merges, reduction of duration of congestion, of late-merging vehicles with low speed and reduction of the merging position.

The locations of highway on-ramps are fixed and centralized approaches can be implemented. However, there exist many situations where the location is not fixed and a centralized road side unit is not available, e.g., in front of construction sites. Decentralized approaches can be used to deal with these locally varying lane reductions, which will be discussed subsequently.

Decentralized Approaches for Lane Reduction Scenarios

In lane reduction scenarios, vehicles on two or more lanes have to merge onto one lane. For this purpose, the velocities of the vehicles have to be adapted in order to allow safe lane change maneuvers. When the vehicles maintain a certain distance to the surrounding ones, the vehicles can change the lane at a certain point, e.g., a few meters before the own lane ends.

A survey that focuses on the communication and decentralized approaches for lane change and merge maneuvers can be found in [41]. Therein, different longitudinal and lateral control strategies are discussed under the assumption of V2V communication, and the effects of imperfection in communication and string stability are shortly discussed.

In [44], a centralized approach is compared to a stand-alone control system. A decentralized receding horizon control framework for a merging vehicle is used, with the possibility to extend it to a centralized computation. However, the framework plans trajectories similar to the planning level presented in Part I, and the traffic flow is not considered separately. The simulations have only been run with constant velocity or acceleration of the vehicle on the main lane, and robustness with respect to large accelerations of the preceding vehicle or external disturbances cannot be guaranteed.

Longitudinal control via platooning and a lateral lane change strategy is performed separately in [75] for merging scenarios. Three linear approaches have been used for longitudinal control, but string stability has not been considered. Note, however, that the sliding mode controller mentioned therein exhibits no sliding mode; instead, the control law is derived from the desired error dynamics without robustness guarantees. Two lateral trajectory generation processes are compared: first, a fifth order polynomial is used, and the second approach uses a cycloidal representation. The different controllers are simulated in a MATLAB toolbox called SimPlatoon and compared with respect to lateral acceleration and time needed. However, the focus of the publication is the lateral trajectory generation for a merging scenario, and the longitudinal controllers are not applicable in presence of disturbances or in high traffic densities.

In [131], a leader-follower approach is investigated, where a vehicle (A) splits from the platoon formation to an adjacent lane in order to give way to another vehicle (B) on the merging lane. Vehicle A then overtakes the platoon leader and becomes the new leader. In addition, a virtual leader-follower approach is discussed, where the entire platoon changes lane, and vehicle B from the merging lane has a separate virtual reference. However, vehicle A is assumed to be “pre-selected”, and the strategic choice of the virtual reference is unclear. Moreover, this approach is not applicable to the two-to-one lane merge scenario, since overtaking is not possible.

Different interaction protocols are discussed in [90], and the merging scenario is divided into basic maneuvers with corresponding messages. Two scenarios have been considered: first, merging of one car into a platoon and second, platoon merging into a platoon. It is assumed that both platoons receive a message from an RSU, which requires once more a V2X communication. The focus of the publication lies on the messages sent and distributed decision-making based on relative distances. The major challenges of the proposed platoon merging approach are stated as follows: first, simultaneous gap making of the entire platoon results in huge deceleration at the tail of the platoon, i.e., string instability, and second, serial gap making is not time-efficient.

In [110], a merging algorithm is presented that decelerates to a certain predefined velocity, then accelerates from this initial velocity to the predefined desired merging velocity, where a second order polynomial for acceleration is used. However, the preceding vehicle is assumed to be known and fixed in advance. Moreover, if all vehicles are decelerating, traffic throughput is reduced.

5. String Stability with Linear Controllers

String stability has been discussed primarily in the context of platooning and many results with linear controllers exist. String stability has been defined in different ways; string stability in a general form has been discussed, e.g., in [130] or [157], whereas [58] has focused on position error attenuation and in [120], the velocities and the accelerations are incorporated additionally. To clarify the important aspects for the velocity adaption of a vehicle string, this chapter summarizes the results on string stability in the linear case.

In a platoon, multiple vehicles are aligned on one lane, where each vehicle adapts the velocity so that a certain distance to the vehicle in front is maintained. The first vehicle of the platoon is the leading vehicle and is hence called “leader”. This leader can be modeled as a double integrator with a reference acceleration u_r ,

$$\begin{aligned}\dot{x}_0 &= v_0, \\ \dot{v}_0 &= u_r,\end{aligned}\tag{5.1}$$

where x_0 is the position and v_0 the velocity of the leader. The reference acceleration of the leader can be arbitrary, but is bounded by $|u_r| \leq u_{r,\max}$. Note that the time argument has been dropped due to readability issues. Depending on the availability of C2C communication, this reference acceleration can be known or unknown to the following vehicles.

The predecessor-following vehicles, or “followers”, are also represented by double integrator dynamics,

$$\begin{aligned}\dot{x}_i &= v_i, \\ \dot{v}_i &= u_i,\end{aligned}\tag{5.2}$$

with the position x_i , the velocity v_i , and the acceleration u_i of agent i and $i = 1, 2, \dots, N$, with N the total number of following agents. The difference between the actual distance $x_{i-1} - x_i$ and the desired distance Δ_i , which can be time-varying, is called spacing error and is defined as

$$e_{x,i} = x_{i-1} - x_i - \Delta_i.\tag{5.3}$$

The primary goal of each following vehicle i is to reach and maintain the desired distance Δ_i to its preceding vehicle, i.e.,

$$\lim_{t \rightarrow \infty} e_{x,i}(t) = 0.\tag{5.4}$$

Besides individual vehicle stability, the overall goal is to affect following vehicles as little as possible, i.e., to not cause accidents or traffic jams upstream. For example in everyday traffic situations, a certain behavior of one vehicle can influence the vehicles upstream and cause

5. String Stability with Linear Controllers

stop-and-go traffic. If the accelerations of the vehicles increase along the string, accidents may occur due to the limited deceleration of the vehicles.

Consider the example in Figure 5.1, where the red vehicle brakes hard (indicated by the red rectangle). While the gray vehicles are not affected, the green vehicles behind the red one have to slow down. Due to a slow reaction of human drivers, each vehicle applies more deceleration than the preceding one in order to stop on time, and the velocities decrease along the string of green vehicles. However, if the maximum deceleration is not sufficient, accidents may occur. Moreover, if a vehicle decides to change the lane to the adjacent one as indicated by the blue vehicle on the right lane, the blue vehicle on the adjacent lane is also affected by the low velocities. Then, vehicles that are upstream, i.e., behind the affected vehicles, might be influenced and the velocities are decreased along the string of vehicles until stop-and-go traffic occurs. This results in bad traffic performance on both lanes, which has been caused by one braking vehicle and small inter-vehicle distances. In order to alleviate these effects, a velocity-dependent distance, also known as constant time-headway spacing, can be used between the vehicles. Then, it is possible for human drivers to react on time, which can improve safety at the cost of larger inter-vehicle spacings.

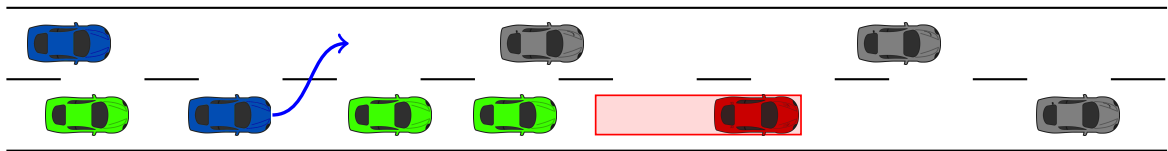


Figure 5.1.: The deceleration of the red vehicle is propagated upstream and causes stop-and-go traffic.

In order to analyze these effects, the concept of string stability has been introduced. The first requirement for string stability can be stated as follows: When starting in the formation with $e_{x,i}(t_0) = 0$ with $t_0 = t(0)$ for all i , attenuation of the spacing errors (5.3) along the vehicle string has to be guaranteed in order to avoid collisions, i.e.,

$$\|e_{x,i}\|_{\infty} \leq \|e_{x,i-1}\|_{\infty}, \quad (5.5)$$

where the leader's acceleration u_r acts as a bounded disturbance and can be arbitrary. The second requirement focuses on the amplification of accelerations: in the string of vehicles, it has to be guaranteed that

$$\|u_i\|_{\infty} \leq \|u_{i-1}\|_{\infty} \quad (5.6)$$

holds, otherwise collisions may occur for saturated actuators.

Note that in the following investigations, the relative position $x_{i-1} - x_i$ and the velocity v_{i-1} of each vehicle's predecessor are assumed to be available via measurements or estimations. In some control laws, the acceleration of the leader is also available, which requires C2C communication between the vehicles. The control task is to find the control input u_i of vehicle i such that (5.4) is achieved, while spacing error amplification is avoided according to (5.5). Several linear control approaches have been analyzed and are summarized in the following discussions.

5.1. Constant Distance Spacing

Consider a constant distance $\Delta_i = \Delta$ in (5.3). The velocity errors using (5.1), (5.2), and (5.3) are defined as

$$e_{v,i} = \dot{e}_{x,i} = v_{i-1} - v_i . \quad (5.7)$$

If the vehicles are in formation, then both the position errors and the velocity errors are zero for all agents, i.e., $e_{x,i} = 0$, $e_{v,i} = 0$. For this to be the case, the accelerations of all vehicles have to be identical, with $u_i = u_r$, and hence the accelerations are not amplified according to (5.6). However, the acceleration of the preceding vehicle has to be known, e.g., via C2C communication. If no communication is available, the unknown acceleration of the leader acts as a disturbance on the platoon. Then, a PD controller

$$u_i = k_p e_{x,i} + k_d \dot{e}_{x,i} , \quad (5.8)$$

with the parameters k_p and k_d , and the errors defined in (5.3) and (5.7) can be used.

Assuming identical controllers for all agents, the error dynamics can be written using (5.1), (5.2), with control input (5.8) as

$$\begin{aligned} \dot{e}_{x,i} &= e_{v,i} , \\ \dot{e}_{v,i} &= u_{i-1} - u_i , \\ &= \begin{cases} k_p e_{x,i-1} + k_d e_{v,i-1} - k_p e_{x,i} - k_d e_{v,i} & i > 1 \\ u_r - k_p e_{x,i} - k_d e_{v,i} & i = 1 . \end{cases} \end{aligned} \quad (5.9)$$

An example with six agents each using a PD controller (5.8) and the constant distance spacing (5.3) with $\Delta_i = 10$ is shown in Figure 5.2 for $k_p = 1$ and $k_d = 1$, where the acceleration of the leader has been chosen as

$$u_r = 2 \sin(t) . \quad (5.10)$$

As can be observed in Figure 5.2a, collisions occur at the end of the platoon, although the initial spacing errors (5.3) and velocity errors (5.7) are zero. In Figure 5.2b, collisions occur when the error crosses the orange dashed line, i.e., when a position error exhibits an undershoot that is larger than the minimum distance. In the given example, collisions occur at the last vehicle $i = 6$ at times $t_1 \approx 16$ s, $t_2 \approx 23$ s, $t_3 \approx 29$ s. It should be noted that it is not sufficient to investigate the error between only the first two vehicles, since disturbances propagate upstream in the platoon when linear controllers are used: although the first following vehicle can react to the leader's change in velocity, the spacing errors are amplified along the string and eventually one vehicle collides with its preceding vehicle. Moreover, as shown in Figure 5.2c and d, the velocities and the accelerations along the platoon are amplified, which has to be avoided due to the limited actuation in vehicles.

In order to deal with the leader's disturbance, the controller parameters of the followers have to be chosen appropriately. However, different frequencies in the leader's acceleration require different controller parameters. This is shown using Figure 5.2, 5.3 and 5.4, where the same scenario with different parameters is discussed. The "fast" controller with $k_p = 1$,

5. String Stability with Linear Controllers

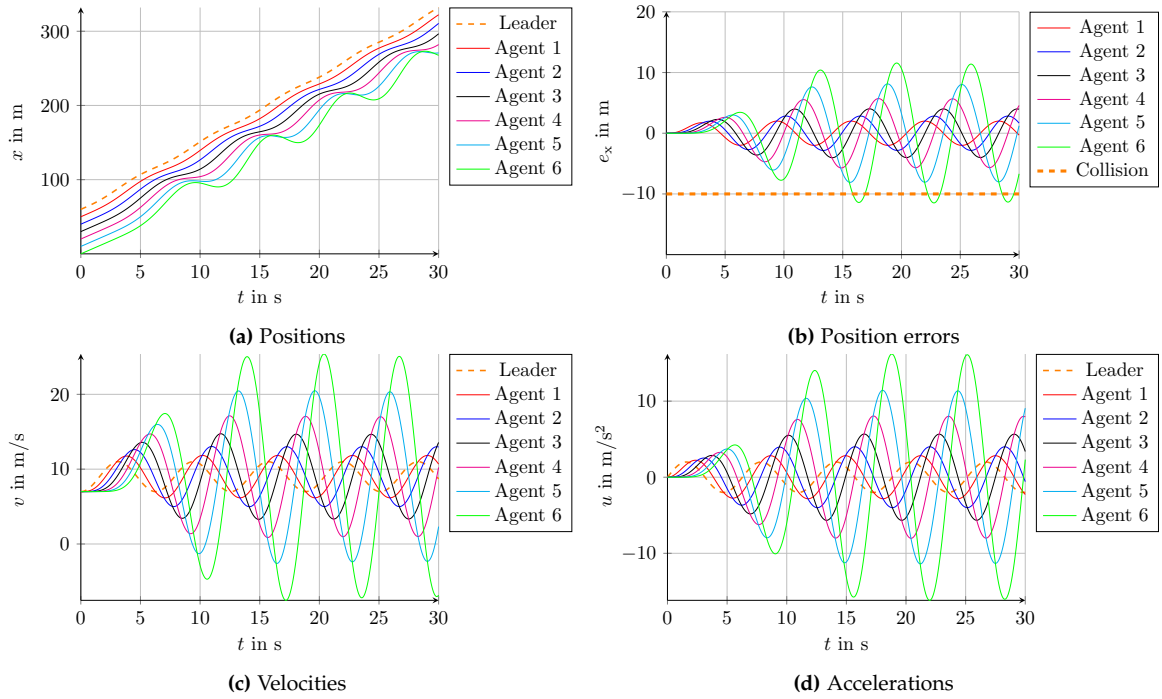


Figure 5.2.: Results using a PD controller with $k_p = 1$ and $k_d = 1$ and constant distance spacing policy with $\Delta_i = 10$

$k_d = 1$ results in position error amplification and the vehicle collides with its predecessor in Figure 5.2. A “slow” controller with $k_p = 0.03$, $k_d = 1$ in Figure 5.3 performs better for the given disturbance (5.10). The position errors are above the orange dashed line, and hence no collisions with the preceding vehicles occur. In Figure 5.4, however, the results are shown for a different frequency of the leader’s acceleration $u_r = 1 \sin(0.1t)$. It can be seen that with the same parameter set $k_p = 0.03$ and $k_d = 1$, collisions occur when the frequency of the disturbance differs.

One can see that it is not possible to design a PD controller with constant distance spacing errors such that collisions are avoided for arbitrary frequencies in the leader’s acceleration.

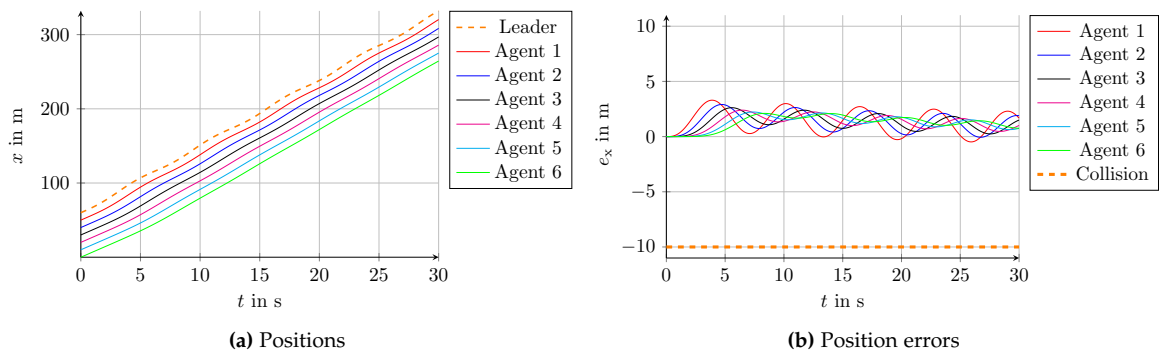


Figure 5.3.: Results using a PD controller with $k_p = 0.03$ and $k_d = 1$ and constant distance spacing policy with $\Delta_i = 10$

5.1. Constant Distance Spacing

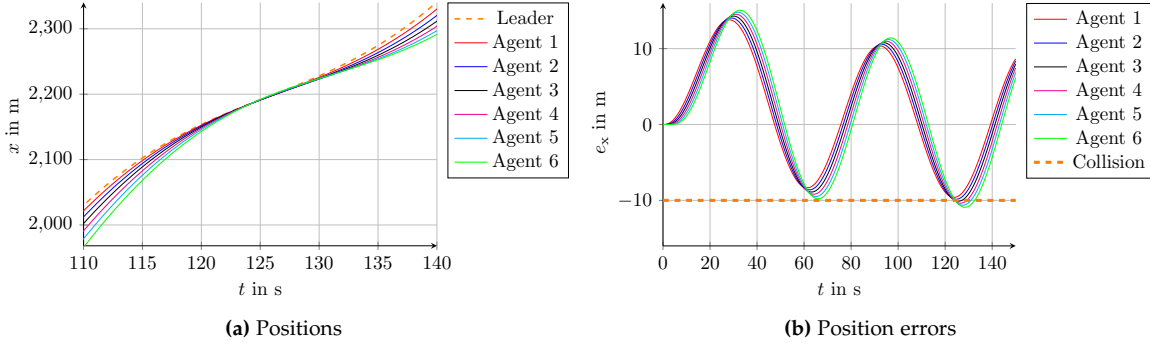


Figure 5.4.: Results using a PD controller with $k_p = 0.03$ and $k_d = 1$ and constant distance spacing policy $\Delta_i = 10$ for $u_r = 1 \sin(0.1t)$

Since the error amplification depends on the frequency, it is reasonable to perform the analysis of string stability in frequency domain. A block diagram of the closed-loop system is shown in Figure 5.5. The transfer function of the plant is given by

$$P(s) = \frac{X_i(s)}{U_i(s)} = \frac{1}{s^2}, \quad (5.11)$$

with the Laplace transformed signals for positions $X_i(s) = \mathcal{L}\{x_i(t)\}$ and control inputs $U_i(s) = \mathcal{L}\{u_i(t)\}$. The transfer function of the controller is defined by

$$R(s) = \frac{U_i(s)}{E_{x,i}(s)} = k_p + k_d s, \quad (5.12)$$

with the Laplace transform of the spacing error (5.3) denoted by $E_{x,i}(s) = \mathcal{L}\{e_{x,i}(t)\}$. Note that in the following investigations, only ideal PD controllers (5.12) are considered, since the velocities are assumed to be measured.

Then, under the assumption of zero initial errors, the error dynamics (5.9) for $i > 1$ in frequency domain are given by

$$s^2 E_{x,i} = k_p E_{x,i-1} + k_d s E_{x,i-1} - k_p E_{x,i} - k_d s E_{x,i}. \quad (5.13)$$

Separating the spacing errors of the two vehicles yields

$$(s^2 + k_d s + k_p) E_{x,i} = (k_p + k_d s) E_{x,i-1}. \quad (5.14)$$

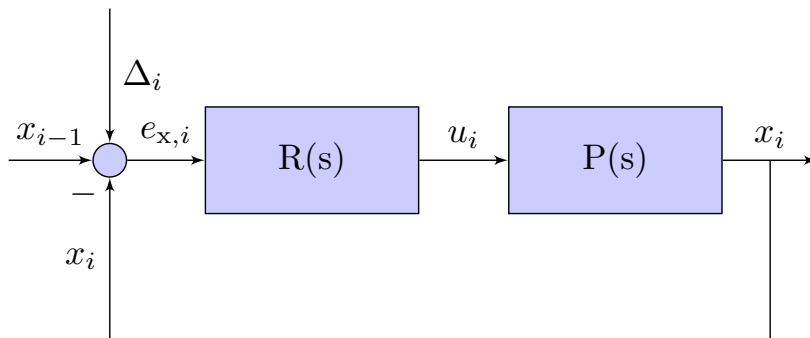


Figure 5.5.: Closed-loop system with the constant distance policy of one agent i

5. String Stability with Linear Controllers

The closed loop system of the errors of one vehicle i with respect to the preceding vehicle $i - 1$ in frequency domain is hence given by

$$T_i(s) = \frac{E_{x,i}(s)}{E_{x,i-1}(s)} = \frac{k_p + k_d s}{s^2 + k_d s + k_p} \quad (5.15)$$

for $i > 1$, where the bode plot for different parameters k_p is shown in Figure 5.6. In order to avoid error amplification (5.5), it is necessary that the magnitude of this transfer function is less than or equal to one. Regardless of the parameter choice of k_p , however, there are always frequencies where $\|T_i(j\omega)\|_\infty > 0$ dB in the bode plot.

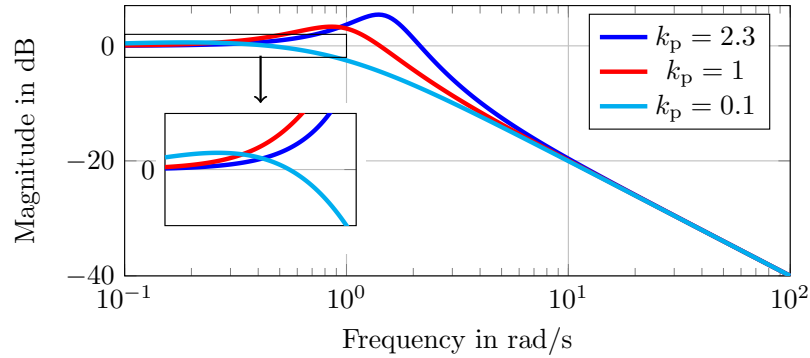


Figure 5.6.: Magnitude of T_i with the constant distance policy and different controller parameters k_p

Hence, it is not possible to guarantee error attenuation for all frequencies, if a PD controller with constant distance spacing error as in (5.8) is used. Note that human drivers are also not capable of maintaining a fixed constant distance to the predecessor due to slow reactions. For this reason, a safety distance that depends on the current velocity of the vehicle is introduced, which is recommended to be approximately 2 to 3 seconds for human drivers, see e.g. [28], [82]. This idea has also been adopted for platoons, and is known as constant time-headway spacing as described in the next section.

5.2. Constant Time-Headway Spacing

The constant time-headway spacing, where the distance is increased proportional to the velocity of the vehicle, is given by

$$e_{t,i} = e_{x,i} - t_h v_i, \quad (5.16)$$

with $e_{x,i}$ according to (5.3). The parameter t_h is called “time-headway”, since it represents the time in seconds that should be maintained between two vehicles, while Δ_i represents the minimum distance at standstill. Consider the desired error dynamics

$$\dot{e}_{t,i} = -k e_{t,i}, \quad (5.17)$$

with the parameter of convergence $k > 0$. With (5.2) and (5.16) this yields the relation

$$(v_{i-1} - v_i - t_h u_i) = -k(x_{i-1} - x_i - \Delta_i - t_h v_i), \quad (5.18)$$

which can be solved for the control input as

$$\begin{aligned} u_i &= k_p(x_{i-1} - x_i - \Delta_i - t_h v_i) + k_d(v_{i-1} - v_i) \\ &= k_p e_{t,i} + k_d e_{v,i}. \end{aligned} \quad (5.19)$$

This procedure results again in a PD controller with $k_p = k/t_h$ and $k_d = 1/t_h$ that additionally uses a velocity-dependent distance with time-headway t_h , which shifts the equilibria of the spacing errors. The error dynamics with control input (5.19) are given by

$$\begin{aligned} \dot{e}_{x,i} &= e_{v,i} \\ \dot{e}_{v,i} &= \begin{cases} k_p e_{x,i-1} - k_p t_h v_{i-1} + k_d e_{v,i-1} - k_p e_{x,i} + k_p t_h v_i - k_d e_{v,i} & i > 1 \\ u_r - k_p e_{x,i} + k_p t_h v_i - k_d e_{v,i} & i = 1. \end{cases} \end{aligned} \quad (5.20)$$

The positions and position errors using the PD controller (5.19) with parameters $k_p = 1$, $k_d = 1$, and constant time-headway with $\Delta_i = 10$ and $t_h = 1$ s are shown in Figure 5.7, where the leader's acceleration is $u_r = 2 \sin(t)$, as in the constant distance case. Due to the time-headway, the negative effects of the disturbance are damped along the string of vehicles. Note that the spacing errors $e_{t,i}$ are small for all times, while the spacing errors $e_{x,i}$ depend on the velocity of the vehicle with $t_h v_i$. The corresponding velocities and accelerations are shown in Figure 5.7c and d. It can be seen that position errors, velocities, and accelerations are attenuated.

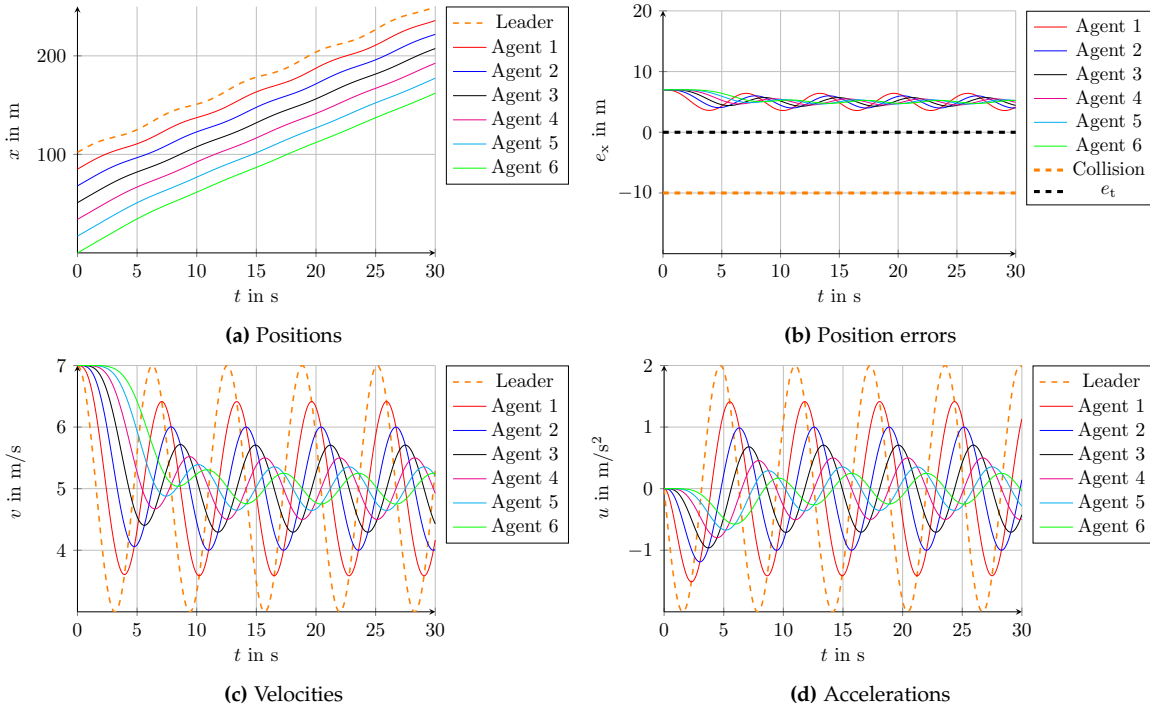


Figure 5.7.: Results using a PD controller with $k_p = 1$, $k_d = 1$, and a constant time-headway policy with $t_h = 1$. Note that the constant time-headway spacing errors $e_{t,i}$ can be kept at zero for all times, while the constant distance spacing errors $e_{x,i}$ are not zero.

5. String Stability with Linear Controllers

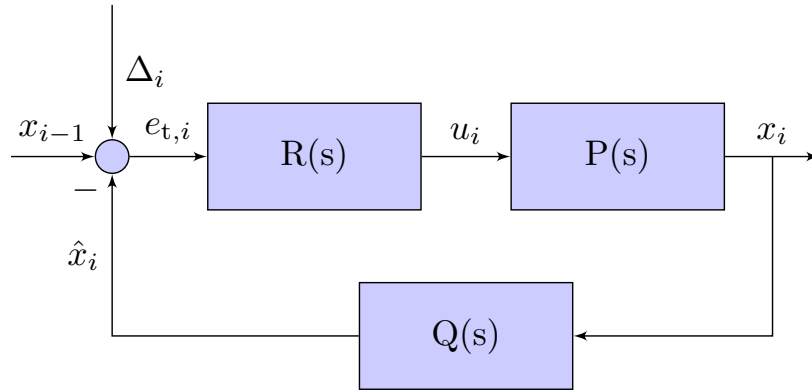


Figure 5.8.: Closed loop system of one agent with constant time-headway spacing, which introduces an additional block $Q(s) = 1 + t_h s$

Similar to the analysis in the constant distance spacing, the frequency domain is investigated. The overall system using the constant time-headway spacing is shown in Figure 5.8. The plant and controller are given by (5.11) and (5.12). Due to the time-headway in (5.16), an additional block Q has been added in the feedback loop compared to Figure 5.5. The Laplace-transform of $\hat{x}_i = x_i + t_h v_i$ is denoted by $\hat{X}_i(s) = \mathcal{L}\{\hat{x}_i(t)\}$ and yields

$$Q(s) = \frac{\hat{X}_i(s)}{X_i(s)} = 1 + t_h s . \quad (5.21)$$

The transfer function of the spacing errors using control law (5.19) then reads as

$$T_{t,i}(s) = \frac{E_{t,i}(s)}{E_{t,i-1}(s)} = \frac{k_d s + k_p}{s^2 + (k_d + t_h k_p) s + k_p} , \quad (5.22)$$

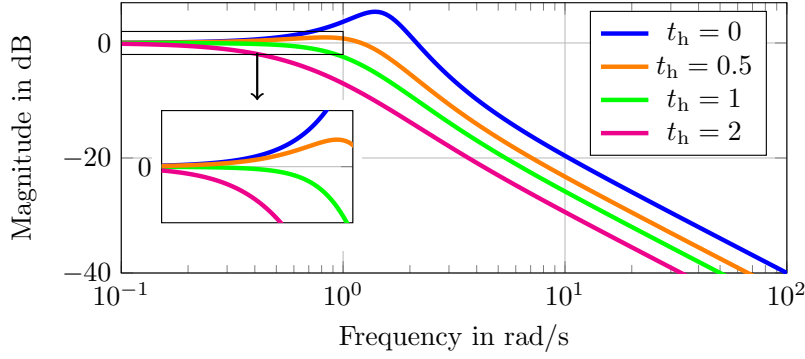
which has an additional damping factor compared to (5.15) that results from the time-headway, and thus the spacing errors are attenuated as shown in Figure 5.7 with appropriate choice of the time-headway.

The bode plots for a fixed parameter setting $k_p = 1$, $k_d = 1$ with different time-headways are shown in Figure 5.9. It can be seen that by increasing the time-headway, the damping increases and errors are attenuated along the string. It should be noted, however, that too large time-headways are not reasonable on highways in scenarios with dense traffic.

5.3. String Stability

Since the safety of a platoon depends strongly on spacing error amplifications, string stability for arbitrary platoon lengths has to be analyzed in order to guarantee collision-free vehicle-following for arbitrary accelerations u_r of the leader.

The first requirement for string stability can be formulated based on the spacing error as in [59]: if spacing error attenuation along the string of vehicles can be guaranteed, the platoon is string stable. Therein it is also discussed that in addition, it is beneficial to avoid

Figure 5.9.: Magnitude of $T_{t,i}$ for different time-headways t_h

position error undershoot. In many applications, however, a small position error undershoot is acceptable as long as the position errors are attenuated.

Let $T_{e,i}$ denote the general spacing error transfer function from vehicle $i - 1$ to vehicle i , i.e.,

$$T_{e,i}(s) = \frac{E_i(s)}{E_{i-1}(s)}, \quad (5.23)$$

and let $g(t)$ denote the impulse response corresponding to $T_{e,i}(s)$. The following stability properties have been described for linear controllers in [59]:

- (S_1) L_2 string stability: the energy of the output error e_i is not larger than the energy of the input error e_{i-1} . This is the case if $\|T_{e,i}\|_\infty \leq 1$,
- (S_2) L_∞ string stability: the maximum magnitude of the output error e_i is not larger than the maximum magnitude of the input error e_{i-1} , with $\|g\|_1 \leq 1$,
- (S_3) String stability without spacing error undershoot: the impulse response fulfills $\|g\|_1 \leq 1$ and in addition $g(t) \geq 0 \quad \forall t$.

The following results have been discussed in [59]: if the impulse response is not positive, frequency-domain analysis can only guarantee L_2 string stability (S_1). Note that the three properties (S_1), (S_2), (S_3) are only equivalent if $g(t) \geq 0 \quad \forall t$. It has been found that L_∞ string stability (S_2) may result in more restrictive controller parameters, and no error overshoot according to (S_3) may in some cases not be achievable, even if L_2 string stability (S_1) and L_∞ string stability (S_2) hold.

However, in later publications, the string stability analysis has been extended in order to include also velocities and accelerations of the agents. Two further properties have been introduced in frequency domain, as described, e.g., in [120]:

The first property is called “weak” string stability and analyzes the behavior of each vehicle with respect to the acceleration of the leading vehicle. In frequency domain the conditions

$$\left\| \frac{E_i(s)}{U_r(s)} \right\|_\infty \leq \gamma_e, \quad \left\| \frac{V_i(s)}{U_r(s)} \right\|_\infty \leq \gamma_v, \quad \left\| \frac{U_i(s)}{U_r(s)} \right\|_\infty \leq 1. \quad (5.24)$$

are important, with γ_e and γ_v defined in [120]. The weak string stability requirement corresponds to an analysis of the maximum amplification in the platoon. This is usually

5. String Stability with Linear Controllers

done to analyze the macroscopic behavior, i.e., the effect of the disturbance u_r on the vehicle string.

The second, more stringent, property is called “strong” string stability and analyzes the string stability between two vehicles in succession. In addition to the spacing errors, velocities and accelerations are analyzed, i.e.,

$$\left\| \frac{E_i(j\omega)}{E_{i-1}(j\omega)} \right\|_{\infty} \leq 1, \quad \left\| \frac{V_i(j\omega)}{V_{i-1}(j\omega)} \right\|_{\infty} \leq 1, \quad \left\| \frac{U_i(j\omega)}{U_{i-1}(j\omega)} \right\|_{\infty} \leq 1, \quad (5.25)$$

for collision avoidance of a vehicle with respect to the preceding one. The additional requirements on the velocity and the accelerations are reasonable: velocities and accelerations are not allowed to increase along the string, since both signals are limited in practice.

In the following discussion, the focus lies on strong string stability, since it is more important from a safety point of view: the spacing error attenuation between two vehicles ensures that two vehicles do not collide when starting in formation. Amplification in velocity may result in “ghost traffic jams” far away from its source, if the velocities along the stream decrease. Avoiding amplification of acceleration ensures applicability in presence of actuator constraints and reliable operation. All vehicles are assumed to have the same bounds on the accelerations, i.e., $u_i \leq u_{\max}$ for all i .

In time-domain, the spacing error attenuation can be formulated using

$$\|e_i\|_{\infty} = \|g * e_{i-1}\|_{\infty} \leq \|g\|_1 \|e_{i-1}\|_{\infty}, \quad (5.26)$$

where $g(t)$ is the impulse response, $*$ is the convolution operator, and e_i is the spacing error of agent i as defined in (5.3) or (5.16). If $\|g\|_1 \leq 1$, string stability can be guaranteed in the \mathcal{L}_{∞} sense. However, analyzing $\|g\|_1$ in time-domain can be difficult, thus switching to frequency-domain is desirable. In linear system theory, the following general statement is valid:

$$|T_e(0)| \leq \|T_e\|_{\infty} \leq \|g\|_1, \quad (5.27)$$

with the Laplace transform

$$T_e(s) = \int_0^{\infty} g(t)e^{-st} dt, \quad (5.28)$$

and for $s = 0$,

$$|T_e(0)| = \left| \int_0^{\infty} g(t) dt \right| \leq \int_0^{\infty} |g(t)| dt = \|g\|_1. \quad (5.29)$$

If the impulse response remains non-negative for all times, the inequality in (5.29) becomes an equality and analysis of spacing error attenuation in frequency domain for \mathcal{L}_2 and \mathcal{L}_{∞} is equivalent, with

$$|T_e(0)| = \|T_e\|_{\infty} = \|g\|_1. \quad (5.30)$$

This means that if

$$\|T_e\|_\infty = \max_{\omega} \|T_e(j\omega)\| = \|g\|_1 \leq 1 \quad (5.31)$$

is fulfilled, string stability in the sense (S_1) , (S_2) , (S_3) can be guaranteed.

According to [158], a non-negative impulse response corresponding to a stable, rational strictly proper transfer function is guaranteed if

- (P_1) the dominant pole is real;
- (P_2) the right-most real zero lies to the left of the dominant pole.

For identical vehicles in the string, it can be shown that the transfer function for all three requirements for strong string stability in (5.25) is the same, thus analyzing T_i in (5.15) or $T_{t,i}$ in (5.22) is sufficient for strong string stability. Then, string stability of the constant distance and constant time-headway policies with linear PD controllers can be discussed in detail. However, note that these definitions are only valid for the linear case with zero initial errors and a non-negative impulse response.

5.3.1. Constant Distance Spacing

The closed loop system using the constant distance spacing is given by (5.15), and computing the magnitude of this transfer function yields

$$\begin{aligned} |T_i(j\omega)| &= \left| \frac{k_d j\omega + k_p}{-\omega^2 + k_d j\omega + k_p} \right| \\ &= \frac{|k_d j\omega + k_p|}{|-\omega^2 + k_d j\omega + k_p|} \\ &= \frac{\sqrt{(k_d \omega)^2 + k_p^2}}{\sqrt{(k_d \omega)^2 + (k_p - \omega^2)^2}}. \end{aligned} \quad (5.32)$$

The denominator is greater than the numerator if $(k_p - \omega^2)^2 > k_p^2$, which guarantees that $|T_i(j\omega)| < 1$. This inequality is equivalent to

$$k_p^2 - 2k_p\omega^2 + \omega^4 > k_p^2 \quad \Rightarrow \quad \omega^4 > 2k_p\omega^2. \quad (5.33)$$

The magnitude of the transfer function becomes

$$|T_i(j\omega)| = \begin{cases} 1 & \omega = 0 \\ < 1 & \omega^2 > 2k_p \\ \geq 1 & \text{otherwise,} \end{cases} \quad (5.34)$$

which means that the constant distance policy with a PD controller cannot guarantee string stability for all frequencies ω . Unfortunately, string unstable behavior occurs at low frequencies, which are of particular interest in highway scenarios.

5. String Stability with Linear Controllers

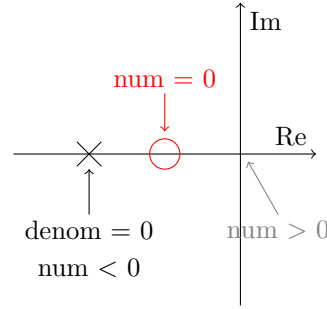


Figure 5.10.: The zero of the transfer function with constant distance spacing is to the right of the rightmost pole.

In addition, the zeros and poles of T_i can be examined: the zero of the transfer function (5.15) is located at $z = -k_p/k_d$, and the poles are located at

$$p_{1,2} = -\frac{k_d}{2} \pm \frac{\sqrt{k_d^2 - 4k_p}}{2}. \quad (5.35)$$

Since the dominant pole has to be real according to (P_1) , $k_d^2 > 4k_p$ has to hold. Note that moving the poles without changing the zero is not possible, and (P_2) cannot be guaranteed. Suppose the poles are real and negative. Then, $s^2 + k_d s + k_p = 0$ for the denominator at a pole can be written as

$$k_d s + k_p = -s^2, \quad (5.36)$$

where $k_d s + k_p$ corresponds to the numerator in (5.15), which is negative at the pole. For the numerator at $s = 0$, one has that $k_d s + k_p = k_p > 0$, which means that the zero is between the rightmost pole and $s = 0$, see Figure 5.10.

5.3.2. Constant Time-Headway Spacing

The poles of the transfer function $T_{t,i}$ in (5.22) are computed as

$$p_{1,2} = -\frac{k_d + k_p t_h}{2} \pm \sqrt{\frac{(k_d + k_p t_h)^2 - 4k_p}{4}}, \quad (5.37)$$

and the zero is located at $z = -k_p/k_d$, as was the case for the constant distance spacing. A first idea to guarantee a positive impulse response according to (P_1) , (P_2) is to set $k_d = 0$ and hence remove the zero. Then, a positive impulse response is given if the poles are real,

$$p_{1,2} = -\frac{k_p t_h}{2} \pm \sqrt{\frac{k_p^2 t_h^2 - 4k_p}{4}} \Rightarrow t_h \geq \frac{2}{\sqrt{k_p}}. \quad (5.38)$$

Counter-intuitively, this means that not using the velocity of the predecessor actually improves the behavior compared to a bad parameter choice of k_d . However, the time-headway may be unrealistically large.

Another way to obtain a positive impulse response is zero-pole cancellation with $t_h = 1/k_d$ as described in [153], yielding

$$\begin{aligned} p_{1,2} &= -\frac{k_d + \frac{k_p}{k_d}}{2} \pm \sqrt{\frac{k_d^2 + 2k_d \frac{k_p}{k_d} + \frac{k_p^2}{k_d^2} - 4k_p}{4}} \\ &= -\frac{k_d + \frac{k_p}{k_d}}{2} \pm \frac{k_d - \frac{k_p}{k_d}}{2} \Rightarrow p_1 = -\frac{k_p}{k_d}, \quad p_2 = -k_d, \end{aligned} \quad (5.39)$$

which guarantees a stable system and a positive impulse response.

In general, one pole can be moved independently when using a velocity-dependent policy, making it easier to achieve a positive impulse response according to (P_1) , (P_2) . A time-headway that is larger than a critical value $t_h > t_{\text{crit}}$ results in a positive impulse response and string stability can be guaranteed. This critical value t_{crit} depends on the controller parameters, since the impulse response of (5.22) will be positive according to [59], if

$$t_h \geq \begin{cases} \frac{1}{k_d} & \frac{k_d^2}{k_p} \geq 1 \\ -\frac{k_d}{k_p} + \frac{2}{\sqrt{k_p}} & \frac{k_d^2}{k_p} < 1. \end{cases} \quad (5.40)$$

In the following it is shown why this is the case.

The poles in (5.37) have to be negative and real, thus

$$\sqrt{\frac{(k_d + k_p t_h)^2 - 4k_p}{4}} \geq 0 \quad (5.41)$$

has to hold. Then, solving for the time-headway yields the relation

$$(k_d + k_p t_h)^2 - 4k_p \geq 0, \quad (5.42)$$

and thus

$$t_h \geq -\frac{k_d}{k_p} + \frac{2}{\sqrt{k_p}}, \quad (5.43)$$

which corresponds to the second case in (5.40). Note that a negative time-headway is not reasonable and hence, only the positive solution of (5.42) is used. The right-most pole has to be to the right of the zero, otherwise the zero-pole cancellation (5.39) is applied, which results in the first case in (5.40), and for $k_d = 0$, (5.38) holds. Note that the requirement for a positive impulse response is more restrictive than having a magnitude of the spacing error transfer function less than one (see [58]).

Remark 7. Note that if the errors $e_{t,i}$ in (5.16) can be kept zero for all times, the signal attenuation is given by

$$G(s) = \frac{V_i(s)}{V_{i-1}(s)} = \frac{1}{1 + t_h s}, \quad (5.44)$$

5. String Stability with Linear Controllers

which has been discussed, e.g., in [66]. Then, string stability can be concluded from

$$\|G\|_{\infty} \leq 1 \quad (5.45)$$

for $t_h \geq 0$. However, a robust control technique has to be used in order to ensure that the errors $e_{t,i}$ in (5.16) are zero for all times for arbitrary disturbances u_r .

In addition, the position error undershoot in response to a lead vehicle acceleration should be avoided for the first following agent $i = 1$. Since the spacing error e_0 does not exist, the transfer function

$$T_r(s) = \frac{E_1(s)}{U_r(s)} = \frac{1 - k_d t_h}{s^2 + (k_d + k_p t_h)s + k_p} \quad (5.46)$$

is analyzed instead. Note that the poles are identical to the spacing error transfer function (5.22). Since no zero is present, the impulse response is positive if the poles are real. Then, position error undershoots that might propagate upstream do not occur.

In the control design, the parameters k_p , k_d of the PD controller $R(s)$ in (5.12) together with the constant time-headway in (5.16) have to be chosen appropriately using the results from (5.40). However, for safety reasons, typical values for the time-headway for ACC assistance systems are usually proposed as $1 \leq t_h \leq 2.2$ seconds. The controller parameters k_p, k_d are then chosen accordingly.

5.3.3. Non-Autonomous Control

The PD controller with constant time-headway spacing is string stable for a proper parameter choice and is often used for Adaptive Cruise Control (ACC) applications. As discussed in the previous section, string stability is guaranteed with (5.31), which depends on the parameters of the controller and the plant. Large time-headways, however, might be necessary to guarantee collision-free driving, which can lead to low acceptance of the assistance system by human drivers. In case of high traffic density, it might not be possible to maintain, for example, a time-headway of two seconds: human drivers usually do not maintain large inter-vehicle distances and cut in front of the vehicle, which causes the automated vehicle to brake. Moreover, in platooning scenarios, one of the major benefits is the reduced fuel consumption, which only applies for small inter-vehicle distances that reduce the air drag. Large time-headways are not desired in these scenarios.

The performance of the ACC can be improved by using communicated information of the predecessor, which is also called non-autonomous control in platoons. For example, the control input u_i of vehicle i with acceleration u_{i-1} of the predecessor as feed-forward reads as

$$u_i = u_{i-1} + k_p e_{x,i} + k_d e_{v,i} . \quad (5.47)$$

The information u_{i-1} is assumed to be communicated over a perfect network, i.e., no network-induced imperfections will be analyzed in the first step. For practical string stability analysis with time-delays, the interested reader is referred to [174].

The advantage of this procedure can be seen immediately: for the double-integrator dynamics (5.2) with the constant distance spacing error (5.3), the velocity error (5.7) and the new control input (5.47), the error dynamics are governed by

$$\begin{aligned}\dot{e}_{x,i} &= e_{v,i} , \\ \dot{e}_{v,i} &= k_p e_{x,i} + k_d e_{v,i} .\end{aligned}\tag{5.48}$$

The dependence on the preceding vehicle's error $e_{x,i-1}$ vanishes, hence $T_i = 0$. There is no error propagation and no position error undershoot when starting with zero initial spacing errors.

When using the constant time-headway policy (5.16), then the control law with feed-forward is given by

$$u_i = u_{i-1} + k_p e_{t,i} + k_d e_{v,i} ,\tag{5.49}$$

where the feed-forward part u_{i-1} counteracts the controller, which tries to steer the spacing error $e_{t,i}$ defined in (5.16) to zero; if $u_i = u_{i-1}$, the distance between the vehicles will remain constant in the case of equal velocities. However, according to the time-headway spacing error definition, the distance has to change with the velocity of the following vehicle. This means that using the feed-forward control input u_{i-1} with the PD controller and the constant time-headway spacing error as in (5.49) does not work very well since feed-forward and feedback pursue different objectives. Hence, using the constant distance spacing policy with feed-forward is more reasonable, which can also be seen in Figures 5.11, 5.12. Using (5.47), the constant distance spacing error is zero for all times for zero initial errors. The constant time-headway spacing error is not zero when feed-forward is used, since the attenuation of the signals counteracts the feed-forward signal.

Hence, when using feed-forward, a trade-off has to be made. If a perfect communication is assumed, the constant distance spacing policy is better with respect to traffic throughput. However, if the communication is subject to packet drop-outs or time-delays, the additional safety distance of the constant time-headway spacing is necessary for collision-free driving.

The constant distance spacing with feed-forward has been used in consensus problems to solve simple formation control tasks. The vehicle-following application using the consensus formulation is hence discussed briefly in the following section.

5.4. Consensus Formulation

Arising from the field of robotics, consensus problems in a multi-agent system can be reformulated for platooning, which is a one-dimensional formation control problem. Consensus is said to be achieved if certain variables of interest reach a common value, e.g., if all agents reach a common position

$$\lim_{t \rightarrow \infty} |x_i(t) - x_j(t)| = 0 \quad \forall i, j ,\tag{5.50}$$

5. String Stability with Linear Controllers

or, if desired distances Δ_i are kept,

$$\lim_{t \rightarrow \infty} |x_i(t) - x_j(t)| = \Delta_i \quad \forall i, j. \quad (5.51)$$

In consensus problems, the agents are connected in either a fixed or a time-varying network, which can be represented by a graph. The so-called network topology is described by the graph's adjacency matrix, which has entries $a_{ij} = 1$ if agent i receives information from agent j and $a_{ij} = 0$ otherwise, i.e., if no information is shared or measured.

Directed information flow, i.e., agent i receives information of agent j but not vice versa, or undirected information flow with $a_{ij} = a_{ji}$ may be used. In addition, a unidirectional information flow means that the information flow is directed in only one direction, e.g., only vehicles in front are considered, while bidirectional flow refers to a flow in both directions, e.g., the predecessors and the following vehicles are considered. In vehicle-following scenarios, every following agent has information of one or more agents, and solutions for various topologies exist. In the platooning applications discussed in this work, only data of the preceding vehicle is assumed to be available, i.e., unidirectional flow without access to the leader's data by all agents. The adjacency matrix for platooning is

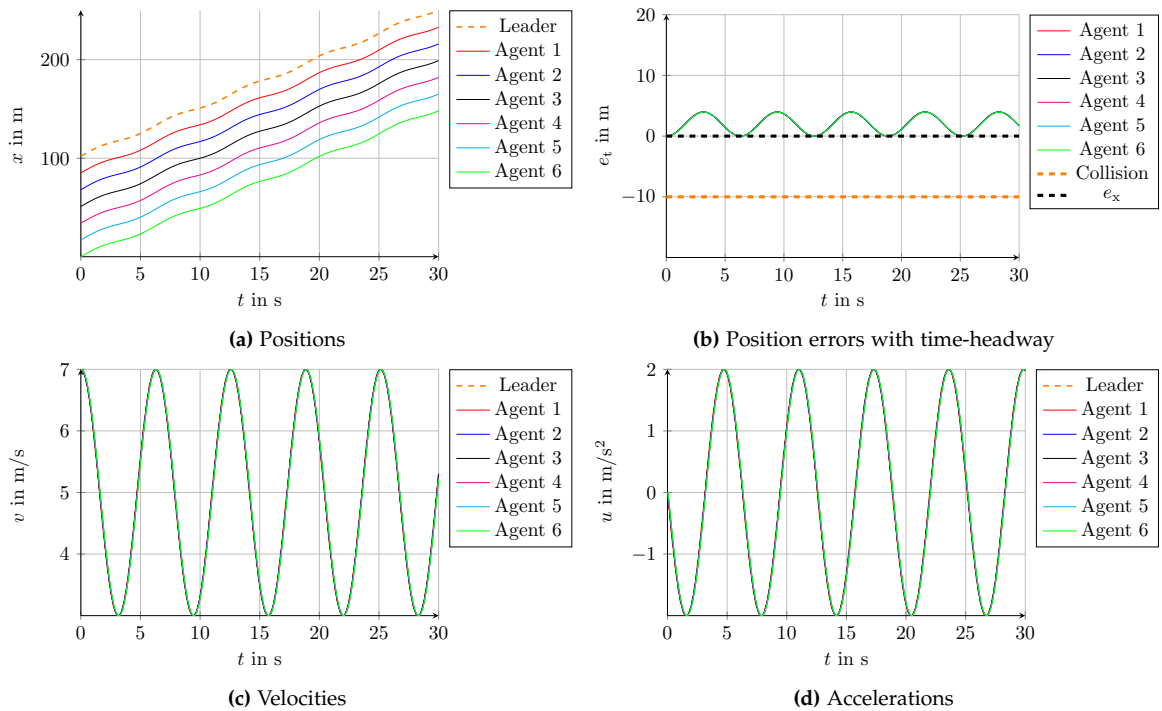


Figure 5.11.: Results using a PD controller with feed-forward and a constant distance spacing

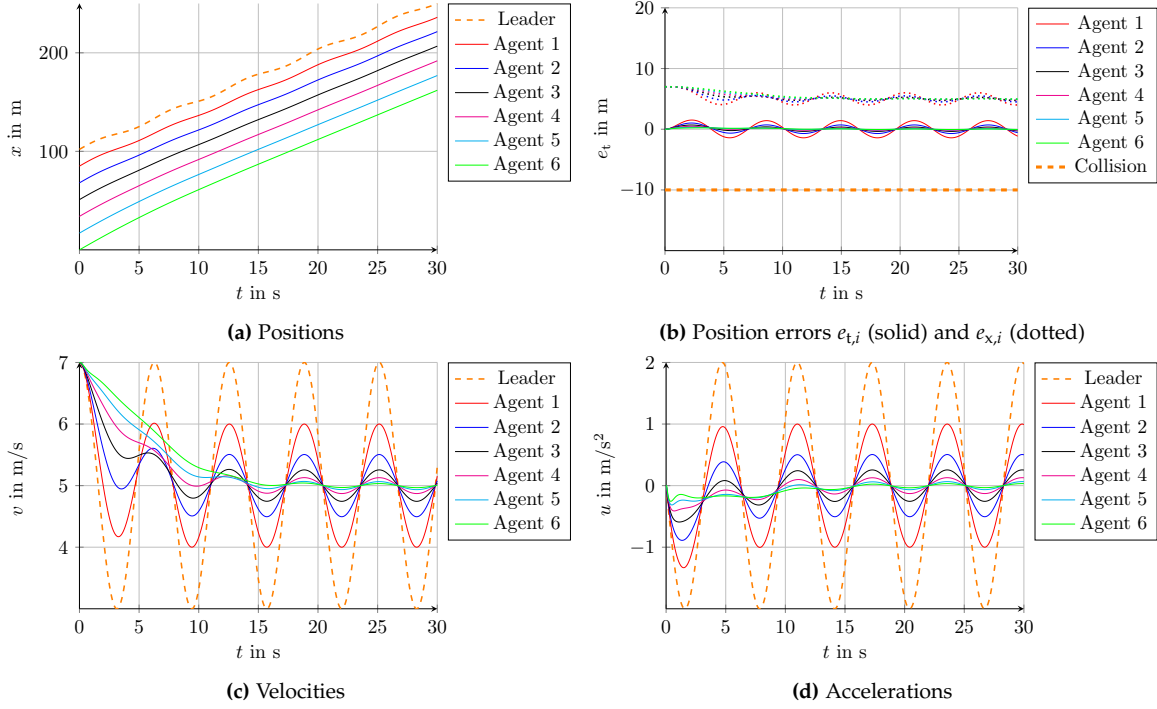


Figure 5.12.: Results using a PD controller with feed-forward and a constant time-headway spacing

thus given by

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}. \quad (5.52)$$

The so-called in-degree vector is represented by $\mathbf{d}_{\text{in}} = [0 \ 1 \ 1 \ 1 \ \cdots \ 1]$, which is the row sum of \mathbf{A} and describes the number of incoming information. The corresponding diagonal in-degree matrix is given by

$$\mathbf{D} = \text{diag}(\mathbf{d}_{\text{in}}). \quad (5.53)$$

The Laplacian matrix is then defined using the matrices (5.52), (5.53) as

$$\mathbf{L} = \mathbf{D} - \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}. \quad (5.54)$$

The position error definition of the consensus problem (5.51) in vector notation reads as

$$e_i = -\mathbf{1}_i^T \mathbf{x} - \Delta_i, \quad (5.55)$$

5. String Stability with Linear Controllers

where \mathbf{l}_i^T is the row of the Laplacian that corresponds to agent i , and the vector \mathbf{x} comprises all agents' positions, i.e., $\mathbf{x} = [x_0 \ x_1 \ \cdots \ x_N]^T$ from (5.1), (5.2). Note that this definition using vector notation is equivalent the one used for platooning in (5.3). The formation distance $\Delta_i = \Delta$ is considered to be equal for all agents i , but can in general also vary. The error of all agents in vector representation $\mathbf{e}^T = [0 \ e_1 \ e_2 \ \cdots \ e_N]^T$ can then be written for all agents as

$$\mathbf{e} = -\mathbf{L}\mathbf{x} - \Delta\mathbf{1}_N, \quad (5.56)$$

where $\mathbf{1}_N$ is an all-ones vector of length N . Then, a standard consensus controller for the desired error dynamics

$$\dot{\mathbf{e}} = -c_e \mathbf{e} \quad (5.57)$$

can be computed, where c_e is a positive parameter. In the platooning application, the error definition

$$\text{agent } i: \quad e_i = x_{i-1} - x_i - \Delta + c(v_{i-1} - v_i), \quad (5.58)$$

$$\text{all agents:} \quad \mathbf{e} = (-\mathbf{L}\mathbf{x}) - \Delta\mathbf{1}_N + c(-\mathbf{L}\mathbf{v}), \quad (5.59)$$

of relative degree one with respect to the control input can be used with (5.1) and (5.2). Note that this error definition considers the *relative* velocity instead of the absolute velocity of the constant time-headway policy (5.16), yielding a constant distance spacing error. Differentiation of (5.58) with (5.57) yields

$$\mathbf{L}\mathbf{u} = -k_1(\mathbf{L}\mathbf{x} + \Delta\mathbf{1}_N) - k_2\mathbf{L}\mathbf{v}, \quad (5.60)$$

where $k_1 = \frac{c_e}{c}$ and $k_2 = (c_e - \frac{1}{c})$. The control input for agent i reads as

$$u_i = u_{i-1} - k_1(x_{i-1} - x_i - \Delta) - k_2(v_{i-1} - v_i), \quad (5.61)$$

which is exactly the same control law as in the constant distance policy with feed-forward as discussed in (5.47). In [134], the leader's acceleration or a group reference acceleration has to be known if the leader's acceleration is time-varying. Otherwise, consensus can only be achieved for $u_r = 0$. By assuming that the preceding vehicle's acceleration u_{i-1} is available to agent i , consensus (5.50), or the formation (5.51), can be achieved.

The results using (5.61) are shown in Figure 5.13. The results correspond to the constant distance spacing policy with feed-forward, which is string stable.

In standard consensus problems, the initial spacing error is not zero. The goal is to eventually reach consensus (5.50) or the formation (5.51). However, string stability is typically not considered; once the formation is reached, it can be maintained and for zero initial errors, this approach is string stable, if feed-forward is used.

Note that a collision with the predecessor can still occur if the initial spacing errors are non-zero and the controller parameters are not tuned appropriately. Position error undershoots cannot be avoided completely if the initial spacing errors are large as shown in Figure 5.14, but the formation can be reached as required by (5.51). The amplification of velocities and accelerations with the constant distance spacing error (5.56) can be seen in Figure 5.14c and

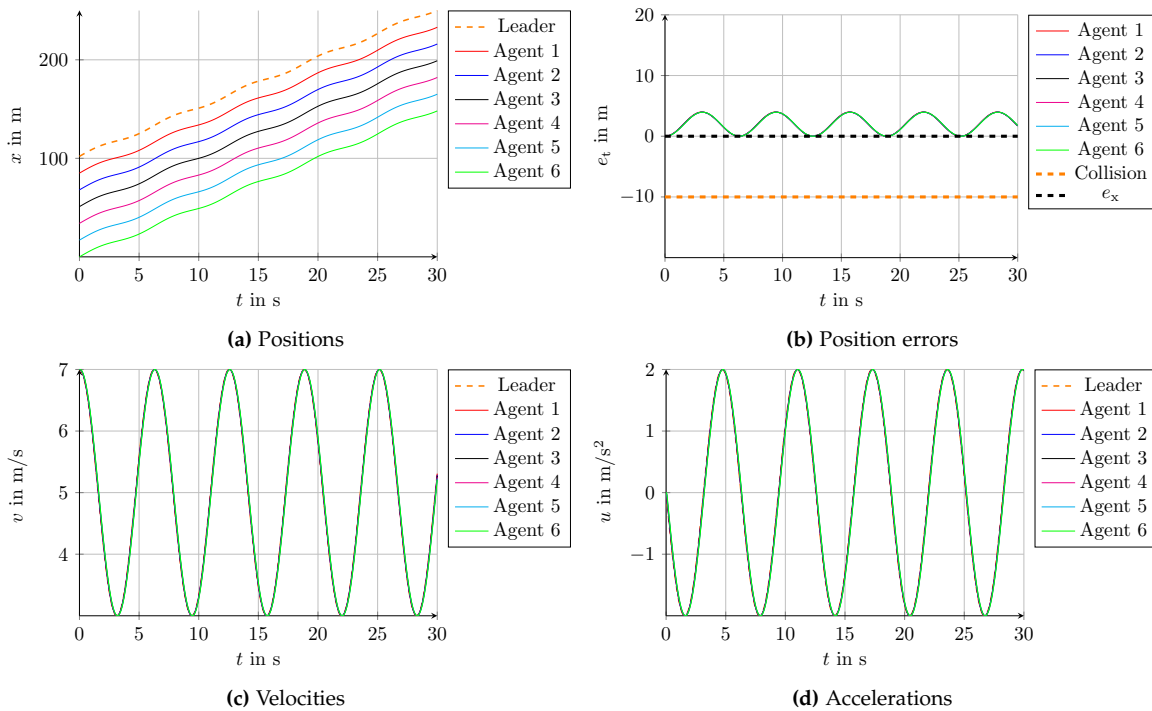


Figure 5.13.: Results using the consensus formulation, i.e., a PD-controller and feed-forward with constant distance spacing

d as long as the error in b is non-zero. Hence, these linear approaches are not applicable in practice for platooning with a large number of vehicles and non-zero initial spacing errors.

Hence, a different approach has to be used for collision-free platooning with non-zero initial spacing errors. Note that the linear controllers with feed-forward can only be used when starting in the formation with zero initial spacing errors; otherwise, string stability analysis in frequency domain is not valid, and collisions cannot be avoided for arbitrary initial errors.

5. String Stability with Linear Controllers

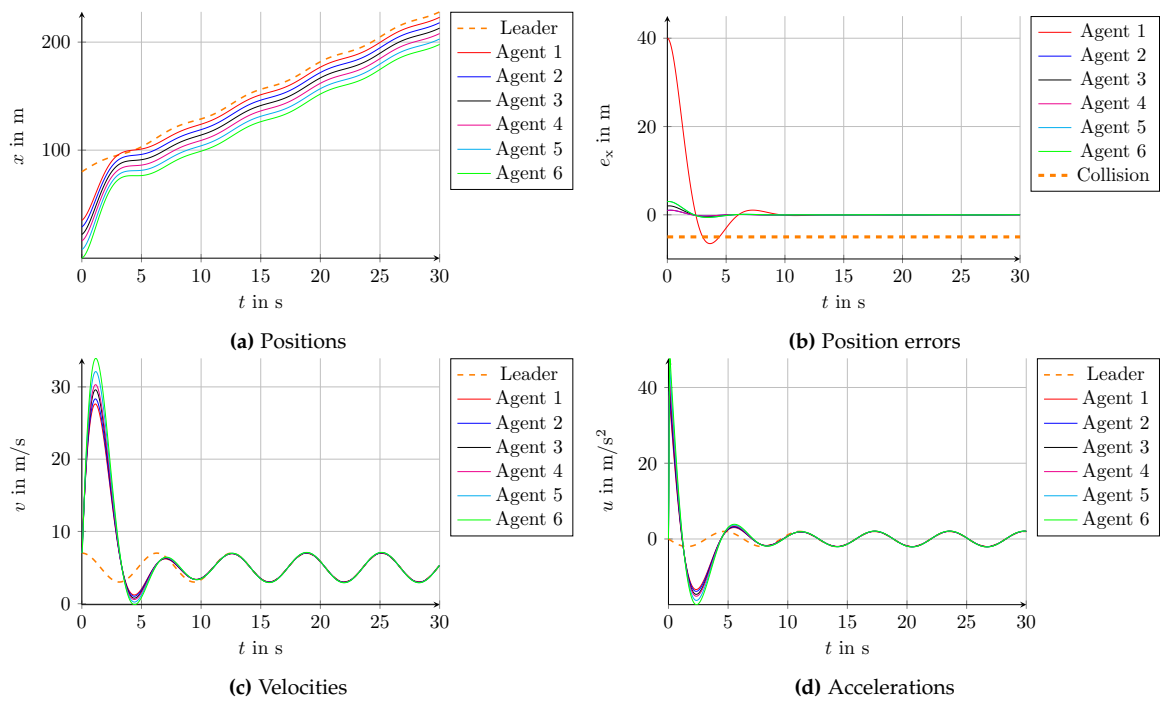


Figure 5.14.: Results using the consensus formulation, i.e., PD-controller and feed-forward with constant distance policy with non-zero initial spacing errors

Summary of String Stability Analysis with Linear Controllers

The analysis of linear approaches presented in this chapter can be summarized as follows under the assumption of zero initial spacing errors:

- a) PD Controller with $\Delta_i = \text{const.}$ from (5.8) is string unstable,
- b) PD Controller with $\Delta_i = \text{const.}$ and feed-forward from (5.47) is string stable and corresponds to the consensus problem statement with zero initial spacing errors,
- c) PD Controller with time-headway $\Delta_i + t_h v_i$ from (5.19) is string stable for $t_h > t_{\text{crit}}$,
- d) PD Controller with $\Delta_i + t_h v_i$ and feed-forward u_{i-1} from (5.49) is string stable for $t_h > t_{\text{crit,ff}}$, where $t_{\text{crit,ff}}$ is the critical time-headway when communication is used. There is a trade-off between feed-forward part and the error definition (5.16).

In vehicle-following applications, the controllers with time-headway spacing c) and d) are typically used. Feed-forward is introduced to reduce inter-vehicle spacings for the platooning scenario, and the time-headway error is used to avoid amplification of position errors, velocities and accelerations. However, the choice of the time-headway is not straightforward, as is also discussed in [93].

Moreover, when reducing inter-vehicle distances and using the acceleration of the preceding vehicle, i.e., the constant distance spacing with feed-forward (5.47), the performance of the controller depends on the communication, which can be subject to imperfections. For example, a delay in the communication can degrade the performance and hence endanger the safety of the passengers in the vehicles. The results of the controller with a time-delay of $\tau = 0.2$ s are shown in Figure 5.15, where collisions will occur for arbitrary platoon lengths due to the amplification of all relevant quantities $e_{x,i}, v_i, u_i$.

Note that the acceleration of the leader can be interpreted as a disturbance to the platoon, which has to be compensated by the following vehicles. Hence, robust control techniques may be used to deal with these disturbances, e.g., H_∞ controllers [129], [176], or sliding mode based techniques. The latter are the focus of the next chapter.

5. String Stability with Linear Controllers

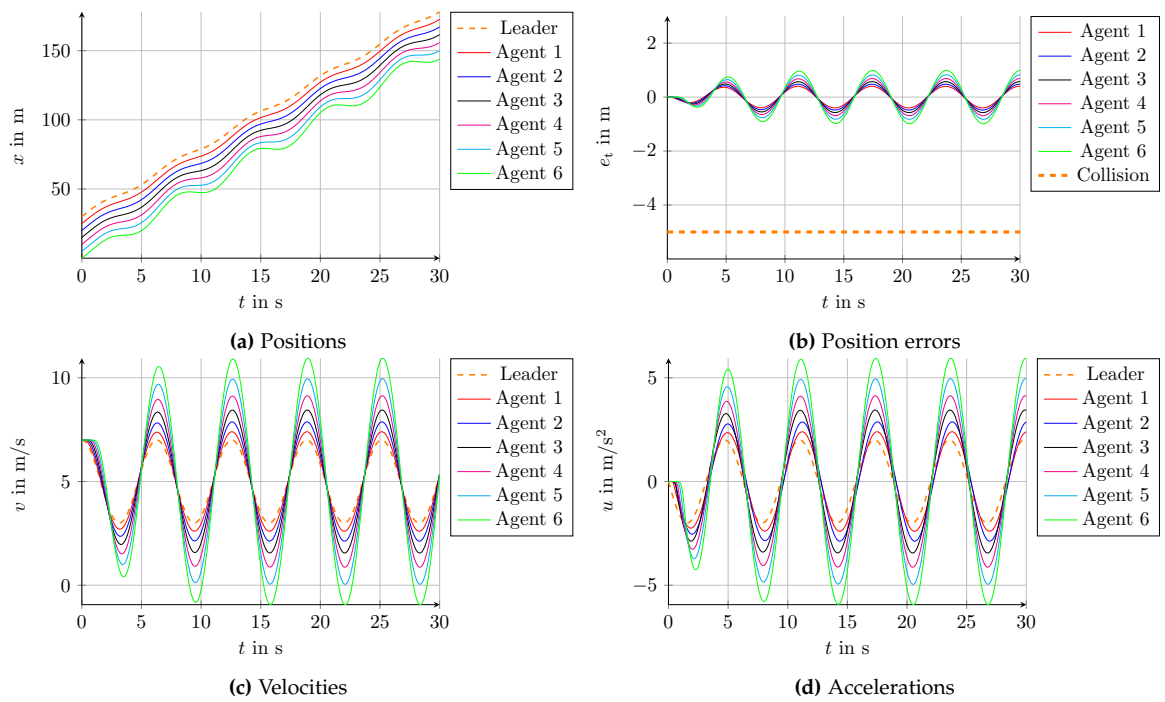


Figure 5.15.: Time-delays in the communication result in string unstable platoons with amplification of all relevant quantities.

6. String Stability in the Context of Sliding Mode Control

In order to reject matched disturbances of a single system, i.e., disturbances acting in the system's input channel, sliding mode controllers can be used, which are known for their robust performance. For longitudinal controller design on highways, these matched disturbances can either be external effects such as wind or road curvature, or they can represent unknown information: in the platooning dynamics discussed in the previous chapter, the acceleration of the leader may be unknown and can thus be interpreted as a disturbance acting on the platoon.

The design procedure of sliding mode controllers consists of two stages: first, a so-called sliding variable σ_i has to be defined. Then, an appropriate control algorithm is applied that drives this sliding variable to zero in finite time T and ensures that it stays at zero, i.e., $\sigma_i(t) = 0$ for all $t \geq T$. The system states are said to be in sliding mode if the sliding variable can be maintained at zero. For sliding mode controllers of higher order r , also the derivatives up to $r - 1$ are zero in sliding mode, i.e., $\sigma_i = \dot{\sigma}_i = \dots = \sigma_i^{(r-1)} = 0$. Note that the degree of the sliding mode controller is not the same as the relative degree of the sliding variable. The super-twisting algorithm, for example, is a second-order sliding mode controller, since it drives the sliding variable and its first derivative to zero $\sigma_i = \dot{\sigma}_i = 0$, whereas the sliding variable has to have relative degree one.

The system is said to be in the reaching phase, if the system is not yet in sliding mode, but approaching the sliding surface, i.e., if $\sigma_i(t) \neq 0$. For sliding variables of relative degree one, the reaching condition

$$\sigma_i \dot{\sigma}_i < -\alpha |\sigma_i|, \quad (6.1)$$

ensures finite-time convergence for some $\alpha > 0$, see, e.g., [154].

The sliding variable can, for example, be an arbitrary error $\sigma_{1,i} = e_i$, but also error dynamics can be defined, such as $\sigma_{2,i} = e_i + c\dot{e}_i$. In the first case, as soon as sliding is established with $\sigma_{1,i} = 0$, the error is zero. The relative degree of the sliding variable depends on the error dynamics and an appropriate controller has to be applied. In the second case, the error dynamics in sliding are governed by

$$\dot{e}_i = -\frac{1}{c}e_i, \quad (6.2)$$

which is exponentially stable in the origin for $c > 0$. Then, the relative degree for this sliding surface is less than the relative degree of the error dynamics, and finite time convergence of $\sigma_{2,i}$ does not imply finite time convergence of the position error.

6. String Stability in the Context of Sliding Mode Control

Sliding mode based controllers have successfully been used for formation control and platooning applications due to their robust performance. In [39], formation control with collision avoidance in aircraft applications has been presented. Therein, two different sliding surfaces have been defined, depending on the sign of the position error, and first order sliding mode control has been used. Although collisions can be avoided when only two vehicles are considered, error amplification in the reaching phase makes this approach unsuitable for platooning. In [157], an adaptive law to reject parameter uncertainties based on sliding mode techniques has been proposed, yielding string stable dynamics for zero initial errors. However, the desired dynamics are only achieved asymptotically, and the control inputs are not bounded for non-zero initial errors. In [55], ACC is described with a constant time-headway spacing, and a first order sliding mode controller has been used to ensure that no position undershoot occurs. The controller parameter is adapted so that less acceleration is applied in comfortable maneuvers, and the controller is more aggressive in safety critical situations. The acceleration of the preceding vehicle, i.e., the disturbance, has been assumed to be zero and has thus been neglected in the analysis. However, an overshoot in position error is not entirely excluded. Note that amplification of position errors and accelerations along a string of vehicles is not considered, since the focus lies on ACC for one vehicle only.

String stable platooning using the suboptimal controller with a constant time-headway spacing for zero initial spacing errors has been proposed in [66], with extensions for slip control [29], lateral control [65], and collision avoidance [68]. This approach will be discussed further in this chapter. The suboptimal controller for longitudinal vehicle control has also recently been published in [64].

In order to solve the problem of non-zero initial spacing errors, bidirectional control has been used in [96], where the vehicle i receives information about its predecessor $i - 1$ and its follower $i + 1$. An extension of this work has been presented in [77], where an integrated sliding variable is used, resulting in assumptions on the initial errors that differ from [96].

In the unidirectional case, the non-zero initial spacing errors problem has not been solved previously. Hence in the following sections, several different error definitions and sliding mode controllers are discussed to achieve string stability with non-zero initial spacing errors in the unidirectional case.

Due to the fact that the vehicles can be interpreted as interconnected system, special requirements for collision-free platooning have to be considered in the controller design, which are presented in Section 6.1. Since string stability has been defined differently in different publications, the problem statement for a string stable platooning performance with non-zero initial spacing errors are stated first. In analogy to the linear case in Chapter 5, different error definitions are discussed using sliding mode controllers in Section 6.2 and 6.3. The focus of these two sections is how the initial spacing error is driven to zero by different sliding mode controllers. Then, a novel approach is presented in Section 6.4 that guarantees string stability for non-zero initial spacing errors without communication.

6.1. Problem Statement

For the longitudinal control design, it is assumed in analogy to the linear case that each vehicle, or “agent”, can be represented by double integrator dynamics. The first vehicle in the string of vehicles, or “leader”, is as in Chapter 5 modeled by

$$\dot{x}_0(t) = v_0(t), \quad \dot{v}_0(t) = u_r(t), \quad (6.3)$$

where $x_0(t) \in \mathbb{R}$, $v_0(t) \in \mathbb{R}$, and $u_r(t) \in \mathbb{R}$ denote position, velocity and acceleration of the leader, respectively. The dynamics of a leader-following agent is assumed to be captured by

$$\dot{x}_i(t) = v_i(t), \quad \dot{v}_i(t) = u_i(t), \quad (6.4)$$

with $i = 1, 2, \dots, N$, and N the number of following agents. $x_i(t) \in \mathbb{R}$, $v_i(t) \in \mathbb{R}$, and $u_i(t) \in \mathbb{R}$ are the position, velocity and acceleration of agent i , respectively. As described in the previous chapter, it is assumed in a general consensus problem statement that an agent i receives the position $x_j(t)$ and the velocity $v_j(t)$ of agent j , if j is a neighbor of i . In the platooning application, agent i receives only information about its preceding vehicle $i - 1$, i.e., in terms of the entries a_{ij} of the adjacency matrix defined in (5.52) $a_{ij} = 1$ for $j = i - 1$, and $a_{ij} = 0$ otherwise. Moreover, it is assumed that the preceding vehicle is constant and predefined, i.e., the network topology of the agents is known and fixed, given by the adjacency matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}. \quad (6.5)$$

In addition, the following assumption is made for the subsequent sliding mode based controller design (see [143]).

Assumption 1. *The acceleration of the leading vehicle in (6.3) is bounded by a maximum acceleration $u_{r,\max}$, i.e., $|u_r(t)| \leq u_{r,\max}$. Moreover, the accelerations of the leader-following agents (6.4) are bounded by a maximum acceleration u_{\max} , i.e., $|u_i(t)| \leq u_{\max}$ for all i , where the leader's acceleration is smaller than the maximum of the leader-following agents $u_{r,\max} < u_{\max}$. The velocities v_i of the agents are positive.*

Note that the control inputs are bounded due to the physical limitation on the accelerations of the vehicles in practice. The leader's acceleration being smaller than the other acceleration bounds is reasonable, because otherwise the vehicles may not be able to follow the leader, or brake in time if the leader decelerates. The assumption on positive velocities is necessary for meaningful application: on a highway, stop-and-go traffic is possible, but driving backwards is not considered in this work.

As in the linear case in Chapter 5, consensus for a system of agents described by (6.3) and (6.4) is achieved if

$$\lim_{t \rightarrow \infty} (x_i(t) - x_j(t)) = \Delta_{i,j} \quad \forall i, j \in \{0, 1, \dots, N\} \text{ where } a_{ij} = 1 \quad (6.6)$$

6. String Stability in the Context of Sliding Mode Control

holds, where $\Delta_{i,j}$ is the desired distance to be maintained between two agents i and j . In contrast to tracking applications of individual systems, consensus has to be achieved cooperatively, considering all neighboring agents. In bidirectional control, the distances to both the predecessor and the following vehicle are considered in the control design, and the results using sliding mode based techniques differ from unidirectional control as described, e.g., in [96]. This work, however, focuses on unidirectional control without communication only.

In the following investigations, the time argument t is omitted for the sake of readability. The position and velocity errors for platooning applications with the connection between the agents as in (6.5) are given by

$$e_{x,i} = x_{i-1} - x_i - \Delta_i, \quad e_{v,i} = v_{i-1} - v_i, \quad (6.7)$$

with the desired distance to the predecessor $\Delta_i = \Delta_{i,i-1}$ in (6.6). Then with (6.3), (6.4), the error dynamics for platooning are governed by

$$\dot{e}_{x,i} = e_{v,i}, \quad \dot{e}_{v,i} = u_{i-1} - u_i. \quad (6.8)$$

The formation of the platoon is achieved if

$$\lim_{t \rightarrow \infty} e_{x,i}(t) = 0, \quad \forall i \in \{1, \dots, N\}. \quad (6.9)$$

In addition to reaching the desired formation (6.6), it is important that no collisions occur; for this purpose, large position error undershoot has to be avoided. Hence, a sufficient condition for collision avoidance is a non-negative position error for platooning,

$$e_{x,i}(t) \geq 0 \quad \forall t, \quad (6.10)$$

where the initial position error is assumed to be non-negative $e_{x,i}(t_0) \geq 0$. Note that the same condition has been used in [59] or [156]. However, the condition can be relaxed for scenarios with multiple lanes, as discussed in Chapter 7.

As third design goal, the attenuation of the position errors and accelerations along the string of vehicles, i.e., string stability, has to be guaranteed. Note that frequency domain analysis as used in Chapter 5 is only applicable for linear systems with zero initial errors and cannot be investigated with sliding mode controllers and/or non-zero initial spacing errors. As discussed in [129], there exist various definitions for string stability, e.g., considering position error undershoot as in [59], or attenuation of position errors, velocities and accelerations [121]. Therein, \mathcal{L}_2 and \mathcal{L}_∞ string stability conditions to analyze the amplification of either position error, velocity or acceleration have been proposed. Since the focus in this work is on collision avoidance and attenuation of position errors, the following definition adapted from [129] is used (see also [143]).

Definition 1 (\mathcal{L}_∞ string stability). *The interconnected system (6.3) and (6.4) with unidirectional control $u_i = u_i(x_i, v_i, x_{i-1}, v_{i-1})$ is called \mathcal{L}_∞ string stable if there exist class \mathcal{K} functions (see [91]) γ and η such that for any initial state $\bar{e}(t_0) \in \mathbb{R}^{2N}$ at initial time t_0 and any acceleration u_r satisfying Assumption 1, it holds that*

$$\|e_i\|_{\mathcal{L}_\infty} \leq \gamma(\|u_r\|_{\mathcal{L}_\infty}) + \eta(\|\bar{e}(t_0)\|) \quad \forall i, \quad (6.11)$$

where the errors are defined with (6.7) by $e_i = \{e_{x,i}, e_{v,i}\}$, and the lumped error vector is denoted by $\bar{e}(t) = [e_{x,1} \ e_{v,1} \ \dots \ e_{x,N} \ e_{v,N}]^T$.

In addition, position undershoot, which is not included, is desirable as discussed in [59]; the control design task is thus summarized as follows.

6.1.1. Summary of Design Goals

A control law $u_i = u_i(x_i, v_i, x_{i-1}, v_{i-1})$ for the interconnected system (6.3), (6.4) has to be designed that fulfills all of the following requirements:

1. A formation with a constant distance (6.6) is achieved.
2. The accelerations of the vehicles are not increasing along the string (and satisfy Assumption 1).
3. The position and velocity errors are independent of the length of the platoon N and the position in the platoon i , i.e., there is no error amplification along the platoon, fulfilling (6.11).
4. There are no position error undershoots, i.e., (6.10) holds and thus collisions are avoided although non-zero initial conditions are present.

Existing control approaches are not capable of fulfilling all requirements, as is discussed in the following two sections. Then in Section 6.4, a novel adaptive sliding variable is proposed using a first order sliding mode controller to achieve all design goals.

6.2. Constant Distance Sliding Variable

A sliding variable for the constant distance spacing can be defined in various ways. In this thesis, two different sliding variables with different relative degrees are presented subsequently.

6.2.1. Constant Distance Sliding Variable with Relative Degree Two

For the constant distance spacing (6.6), a sliding surface with relative degree two can be defined as

$$\sigma_i = x_{i-1} - x_i - \Delta_i = e_{x,i} , \quad (6.12)$$

with the desired distance Δ_i and the positions of the vehicles from (6.3), (6.4). As a controller, the twisting algorithm is used, which is a popular second-order sliding mode controller for relative degree two sliding surfaces [101]. The control input reads as

$$u_i = a_i \text{sign}(\sigma_i) + b_i \text{sign}(\dot{\sigma}_i) . \quad (6.13)$$

The sliding dynamics are obtained by differentiating (6.12) twice, and with (6.8), (6.13) are governed by

$$\ddot{\sigma}_i = u_{i-1} - a_i \text{sign}(\sigma_i) - b_i \text{sign}(\dot{\sigma}_i) . \quad (6.14)$$

6. String Stability in the Context of Sliding Mode Control

If u_{i-1} is interpreted as a disturbance that is bounded according to Assumption 1, then a_i and b_i have to be chosen so that this disturbance can be compensated, i.e.,

$$a_i > b_i + u_{\max,i-1}, \quad b_i > u_{\max,i-1}. \quad (6.15)$$

One can see that this requires $u_{\max,i} > u_{\max,i-1}$, which yields parameters a_i, b_i that depend on the position in the string and increase along the string. However, since all agents are constrained by the same maximum acceleration due to Assumption 1, such a choice of parameters is not possible and hence, reaching cannot be guaranteed.

In consensus problems, the goal is to achieve a zero position error as in (6.6). Position error undershoot or amplifications of certain states are typically not part of the problem statement and are not considered. Eventually, as has been shown for multi-agent systems in [126], consensus can still be achieved with the twisting controller (6.13) (see Appendix A for details). From a practical point of view, this means that a following vehicle can reach the leader, if the leader does not accelerate at a maximum for all times. This is usually the case in the platooning application: a large constant acceleration for all times is not reasonable since velocities on the highway are constrained and hence at some point, the vehicles have to stop accelerating. Then, if the leading vehicle stops accelerating, the following vehicle can reduce the distance and will finally be able to reach the leader. If the leader has been reached by vehicle 1, vehicle 1 will keep the velocity of the leader and the next follower, i.e., vehicle 2, will be able to reach the desired distance, and so on. Hence, if the leader does not apply maximum acceleration for all times, then the leader can be reached and eventually, all followers will reach consensus, i.e., a desired distance.

Note, however, that the twisting controller does not avoid position error undershoot and is thus not suitable for platooning applications with non-zero initial errors. The results of a simulation using the twisting algorithm are shown in Figure 6.1 with $a = 12$, $b = 6$ and $\Delta_i = \Delta = 5$. It can be seen that consensus (6.6) can be achieved, but collisions occur due to position error undershoot in the reaching phase. Additionally, the velocities and position errors along the string are amplified, as can be seen in Figure 6.1b and d. The phase plane in Figure 6.2 shows the typical behavior using the twisting algorithm, which does not avoid position error over- or undershoot, and collisions occur if the orange dashed line is crossed. Thus, it is reasonable to investigate sliding mode controllers that ensure that there is no position error undershoot.

Suboptimal Controller

The generalized suboptimal controller (SOC) can be used in order to avoid overshoot or undershoot in the sliding variable as described in [32], [33]. It is a second order sliding mode controller for relative degree two systems, with the control input

$$u_i = \alpha_i(t)k_{SO}\text{sign}(\sigma_i - \beta\sigma_{M,i}), \quad (6.16)$$

with parameter k_{SO} chosen appropriately and $\beta = 0.5$. Therein, $\sigma_{M,i}$ is the value of the sliding variable at the latest time instance at which the corresponding time derivative was

6.2. Constant Distance Sliding Variable

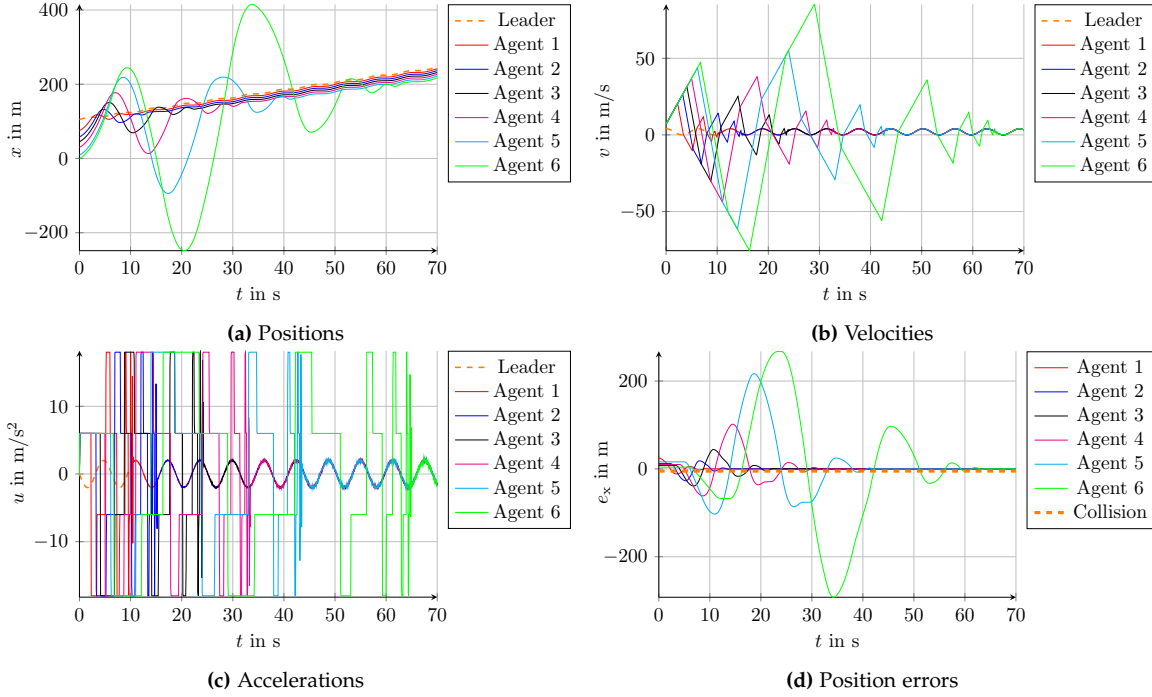


Figure 6.1.: Results using the twisting algorithm with constant distance spacing

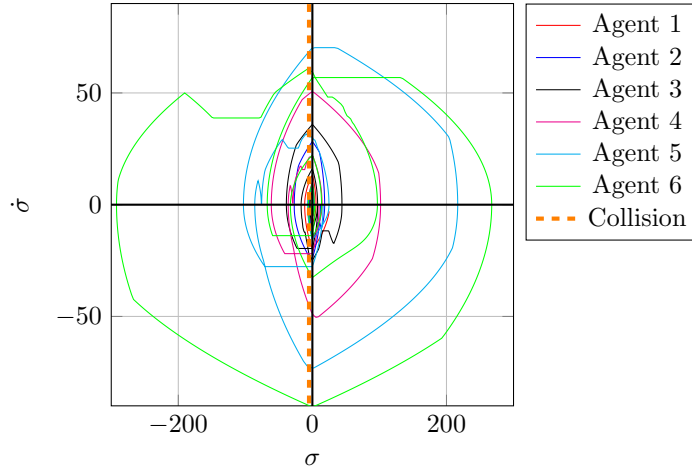


Figure 6.2.: Phase plane of the sliding variables using the twisting algorithm

zero, and $\alpha_i \geq 1$ is an adaptive parameter that is used to ensure monotonic convergence of the sliding variable. According to [34], this adaptation is given by

$$\alpha_i(t) = \begin{cases} 1 & \text{if } (\sigma_i - \beta\sigma_{M,i})\sigma_{M,i} \geq 0 \\ \alpha_i^* \in [1; \infty) \cap \left[\frac{D+(1-\beta)k_{SO}}{\beta k_{SO}}; \infty \right) & \text{if } (\sigma_i - \beta\sigma_{M,i})\sigma_{M,i} < 0, \end{cases} \quad (6.17)$$

where D is the upper bound of the matched disturbance, and $k_{SO} > D$ holds.

For the vehicle-following problem, the sliding variable of relative degree two is given by the constant distance spacing (6.7) as $\sigma_i = e_{x,i}$. The sliding dynamics using (6.16) are then

6. String Stability in the Context of Sliding Mode Control

governed by

$$\begin{aligned}\ddot{\sigma}_i &= u_{i-1} - u_i \\ &= u_{i-1} - \alpha_i(t)k_{\text{SO}}\text{sign}(\sigma_i - \beta\sigma_{\text{M},i}).\end{aligned}\quad (6.18)$$

In order to reach and maintain sliding, the condition

$$|u_{i-1}| < \alpha_i(t)k_{\text{SO}} = |u_i| \quad (6.19)$$

has to be satisfied. However, this implies that the parameters α_i and thus the accelerations increase along the platoon with $|u_0| < |u_1| < |u_2| < \dots < |u_N|$. Hence, it is not possible to apply this control input to a large number of vehicles in the platoon, and the platoon is string unstable due to amplification of accelerations. Such an amplification also occurs if communication is used, i.e., if

$$u_i = u_{i-1} + \alpha_i(t)k_{\text{SO}}\text{sign}(\sigma_i - \beta\sigma_{\text{M},i}), \quad (6.20)$$

where α_i is either constant or adaptive, which results in the sliding dynamics

$$\begin{aligned}\ddot{\sigma}_i &= u_{i-1} - u_i \\ &= -\alpha_i(t)k_{\text{SO}}\text{sign}(\sigma_i - \beta\sigma_{\text{M},i}).\end{aligned}\quad (6.21)$$

In Figure 6.3, the results using the suboptimal controller are shown, where the constant distance spacing has been used with $\Delta_i = 5$ and feed-forward as in (6.20) with constant $\alpha_i = 1$. There is no position error undershoot, but the velocities and accelerations are amplified along the string. Moreover, the convergence of the position error exhibits multiple plateaus that arise from $\dot{\sigma}_i = 0$ as shown in Figure 6.4, and are uncomfortable for the passengers of the vehicles. Hence, the controller (6.20) cannot be used without modifications in practice.

Better convergence while maintaining monotonic convergence, for example, can be achieved by using the controller proposed by Bhat and Bernstein [42] with discontinuous extension to ensure robustness with respect to disturbances [52]. However, all controllers using the constant distance spacing error (6.7) suffer from the same problems with non-zero initial spacing errors: using feed-forward or unbounded acceleration amplifies the accelerations along the string, which yields string unstable behavior. Without amplification of accelerations on the other hand, reaching cannot be guaranteed and position error overshoots occur. Before moving on to the constant time-headway spacing, the constant distance sliding surface with relative degree one is investigated.

6.2.2. Constant Distance Sliding Variable with Relative Degree One

For the constant distance spacing (6.7), the desired error dynamics can be achieved using a sliding variable of relative degree one as

$$\sigma_{\text{d},i} = e_{\text{x},i} + ce_{\text{v},i}. \quad (6.22)$$

Then in sliding, i.e., with $\sigma_{\text{d},i} = 0$, the position error converges asymptotically to zero with

$$e_{\text{v},i} = \dot{e}_{\text{x},i} = -\frac{1}{c}e_{\text{x},i}, \quad (6.23)$$

6.2. Constant Distance Sliding Variable

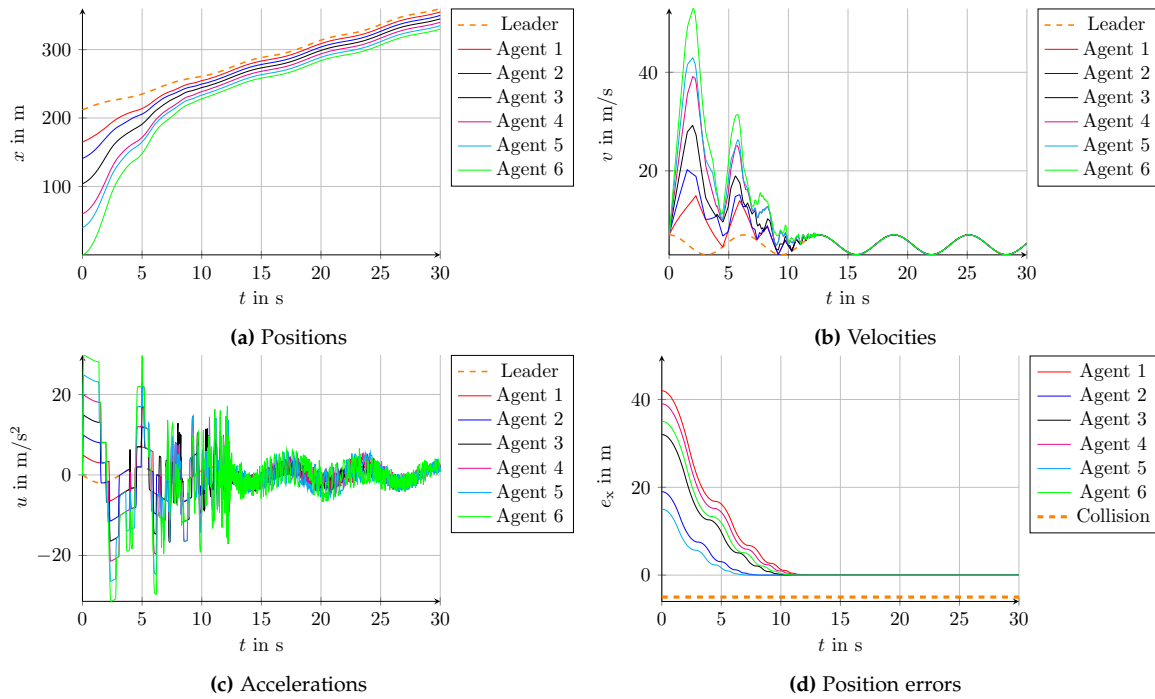


Figure 6.3.: Results using the SOC with constant distance spacing and feed-forward

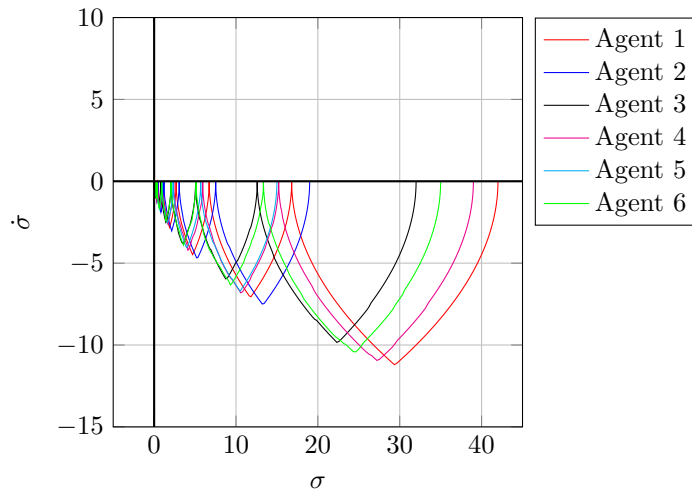


Figure 6.4.: Phase plane of the sliding variables using the SOC with monotonic convergences

where $c > 0$ is the convergence parameter. Note that an appropriate sliding mode controller for sliding variables with relative degree one has to be used. Alternatively, similar to [67] sliding mode controllers for relative degree two may be used, the derivative of the acceleration \dot{u}_i can be used as auxiliary control input, which increases the relative degree by one.

6. String Stability in the Context of Sliding Mode Control

Suboptimal Controller

The suboptimal controller can be used for the auxiliary control input \dot{u}_i , and the control law becomes

$$\dot{u}_i = \alpha_i(t)k_{\text{SO}}\text{sign}(\sigma_{\text{d},i} - \beta\sigma_{\text{M},\text{d},i}) , \quad (6.24)$$

which results in the dynamics

$$\ddot{\sigma}_{\text{d},i} = u_{i-1} - u_i + c(\dot{u}_{i-1} - \alpha_i(t)k_{\text{SO}}\text{sign}(\sigma_{\text{d},i} - \beta\sigma_{\text{M},\text{d},i})) . \quad (6.25)$$

In order to guarantee reaching, the information of the predecessor \dot{u}_{i-1} , u_{i-1} has to be communicated and incorporated into the control law

$$\dot{u}_i = \dot{u}_{i-1} + \frac{1}{c} (u_{i-1} - u_i + \alpha_i(t)k_{\text{SO}}\text{sign}(\sigma_{\text{d},i} - \beta\sigma_{\text{M},\text{d},i})) , \quad (6.26)$$

leading to the dynamics

$$\ddot{\sigma}_{\text{d},i} = -\alpha_i(t)k_{\text{SO}}\text{sign}(\sigma_{\text{d},i} - \beta\sigma_{\text{M},\text{d},i}) . \quad (6.27)$$

Alternatively, the parameter $\alpha_i(t)$ has to grow with the vehicle number as in Section 6.2. The results using control law (6.26) are shown in Figures 6.5 and 6.6. Note the smoother convergence of the position errors, which makes this approach more comfortable than the relative degree two sliding surface (6.12). Moreover, the acceleration with sliding variable (6.22) is continuous. However, only asymptotic convergence of the position error (5.3) is achieved, more information needs to be communicated and again, accelerations increase with the position in the string.

In the case that only the acceleration u_{i-1} is communicated, a sliding mode controller for relative degree one sliding surfaces has to be used, which is discussed in the next section.

First Order Sliding Mode Controller

A first order sliding mode controller (FOSMC) drives the relative degree one sliding variable σ to zero without undershoot as presented, e.g., in [154]. In order to analyze the convergence of the sliding variable σ using the FOSMC, a Lyapunov function V and its derivative [154] have to satisfy

$$V = \frac{1}{2}\sigma^2 > 0, \quad \dot{V} = \sigma\dot{\sigma} < 0, \quad \forall \sigma \neq 0 . \quad (6.28)$$

Then, the sliding surface $\sigma = 0$ can be reached in finite time T and sliding can be maintained for all times $t \geq T$. The control law using first order sliding with the sliding variable (6.22) is given by

$$u_i = k\text{sign}(\sigma_{\text{d},i}) . \quad (6.29)$$

6.2. Constant Distance Sliding Variable

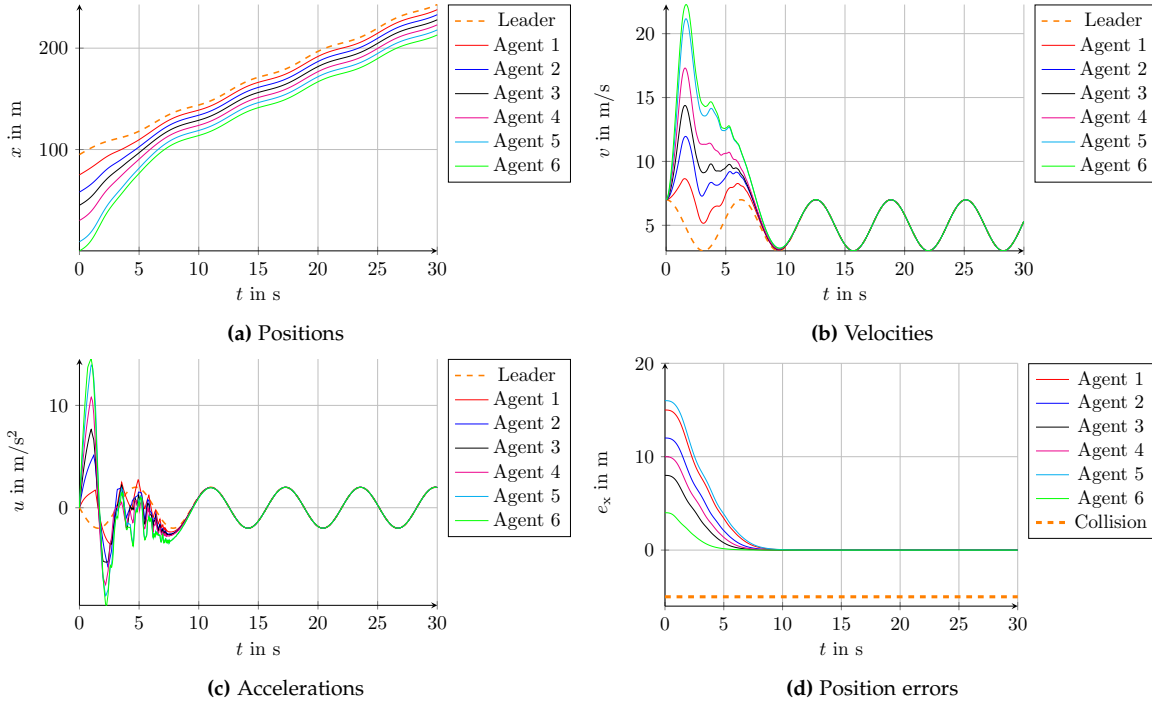


Figure 6.5.: Results using the SOC with constant distance sliding surface of relative degree one and feed-forward

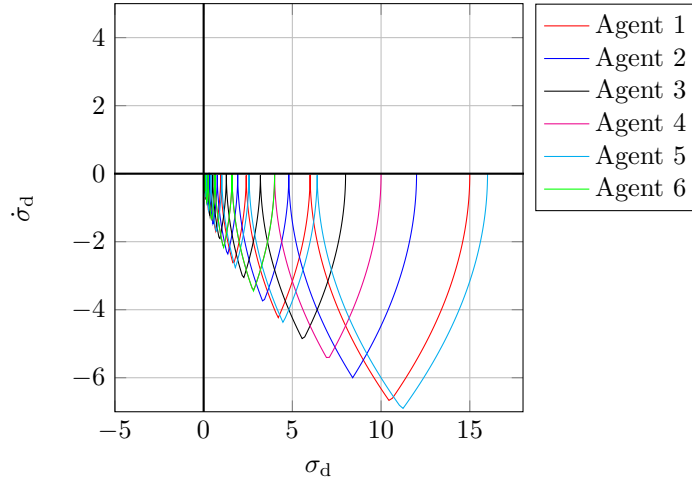


Figure 6.6.: Phase plane of the sliding variable using the SOC with feed-forward

Then with the error dynamics (6.8), the dynamics of the sliding variable (6.22) are governed by

$$\dot{\sigma}_{d,i} = e_{v,i} + c(u_{i-1} - k\text{sign}(\sigma_{d,i})) . \quad (6.30)$$

In the remainder of this section, the parameter $c = 1$ has been chosen for easier readability, without loss of generality. Computing the Lyapunov derivative in relation (6.28) yields

$$\dot{V} = \sigma_{d,i}(e_{v,i} + u_{i-1} - k\text{sign}(\sigma_{d,i})) . \quad (6.31)$$

6. String Stability in the Context of Sliding Mode Control

If the initial errors $e_{x,i}(t_0), e_{v,i}(t_0)$ of all agents are zero, then the reaching condition (6.28) is satisfied for $\sigma_{d,i}(t_0) = 0$ and thus $\dot{\sigma}_{d,i}(t) = 0$ for all i and for all times t if $k > |u_{i-1}|$. Note that for $\sigma_{d,i} = 0$, $\dot{V} \leq 0$ is sufficient to maintain sliding, which results in $k \geq |u_{i-1}|$. With the constant distance spacing sliding variable (6.22) in sliding, the control inputs fulfill

$$u_i = u_{i-1} = u_r \quad \forall i = 1, \dots, N, \quad (6.32)$$

and due to Assumption 1, $k > |u_r|$ can be satisfied and sliding is maintained for all times for zero initial errors. This means that the formation with a constant distance spacing can be maintained when starting in formation, although the acceleration of the predecessor u_{i-1} is not communicated.

Remark 8. Note that it is not possible to use the well-known super-twisting algorithm [100] to compensate the disturbance u_{i-1} , since the super-twisting algorithm can only reject disturbances δ which are Lipschitz continuous, i.e., which satisfy

$$|\dot{\delta}(t)| \leq L \quad \forall t \quad (6.33)$$

for some L . Here, the disturbance in (6.30) is given by $\delta = e_{v,i} + cu_{i-1}$, which is discontinuous due to the sign function in u_{i-1} .

However, if initial spacing errors are non-zero, it is not possible to guarantee that no position error undershoot occurs, since

$$k > |e_{v,i} + u_{i-1}| \quad (6.34)$$

has to hold to guarantee reaching. However, this condition cannot be satisfied for arbitrarily large non-zero initial errors and thus the control design goals are violated.

The results using (6.29) for non-zero initial errors are shown in Figure 6.7 and 6.8. It can be seen that position error undershoot cannot be avoided and collisions occur. In the phase plane of the sliding variable in Figure 6.8, it can be seen for all agents that sliding along the surface cannot be maintained when the surface is reached. Eventually, consensus (6.6) is achieved nonetheless, but this approach is clearly not applicable to platooning in practice.

Another first order sliding mode controller can be used that also incorporates information about the preceding vehicle's acceleration. It is given by

$$u_i = (v_{i-1} - v_i) + u_{i-1} + k\text{sign}(\sigma_{d,i}), \quad (6.35)$$

which yields the sliding dynamics

$$\dot{\sigma}_{d,i} = -k\text{sign}(\sigma_{d,i}). \quad (6.36)$$

Note that the relative velocity $(v_{i-1} - v_i)$ is already known from the sliding surface design (6.22). Then, the position error converges to zero asymptotically and position undershoot is avoided as shown in Figure 6.9. However, the velocities and control inputs along the string again increase with the position in the string. Due to (6.36), sliding can be reached and maintained for all times as can be seen in Figure 6.10.

6.2. Constant Distance Sliding Variable

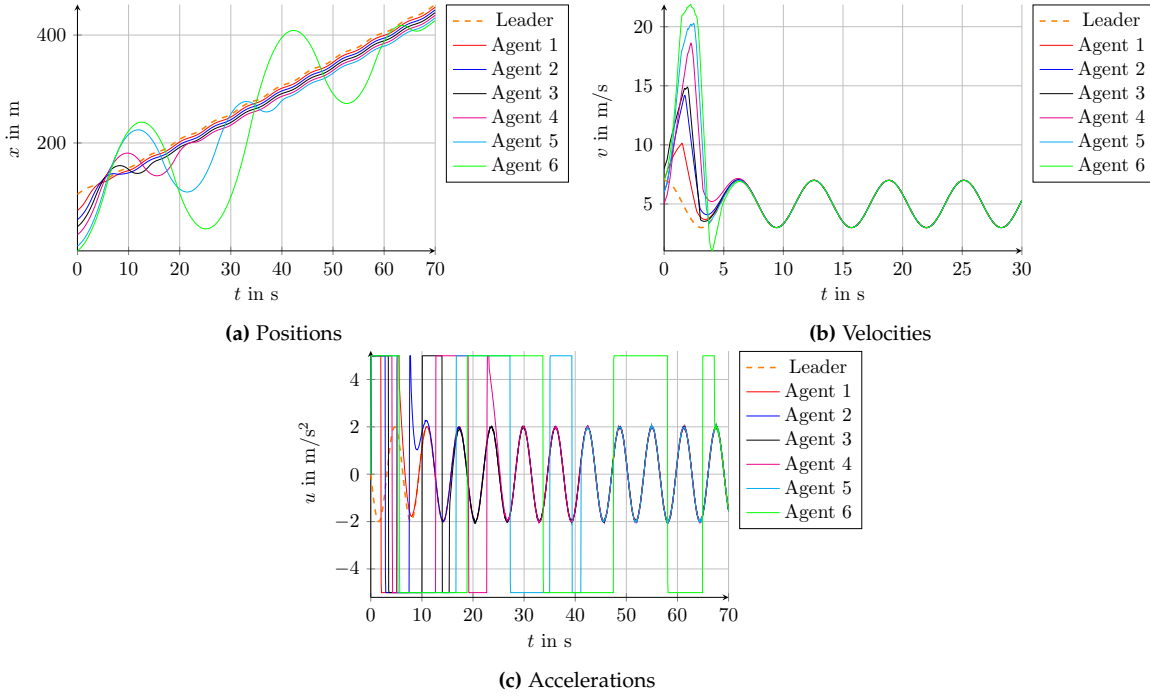


Figure 6.7.: Results using the FOSMC with constant distance sliding surface of relative degree one

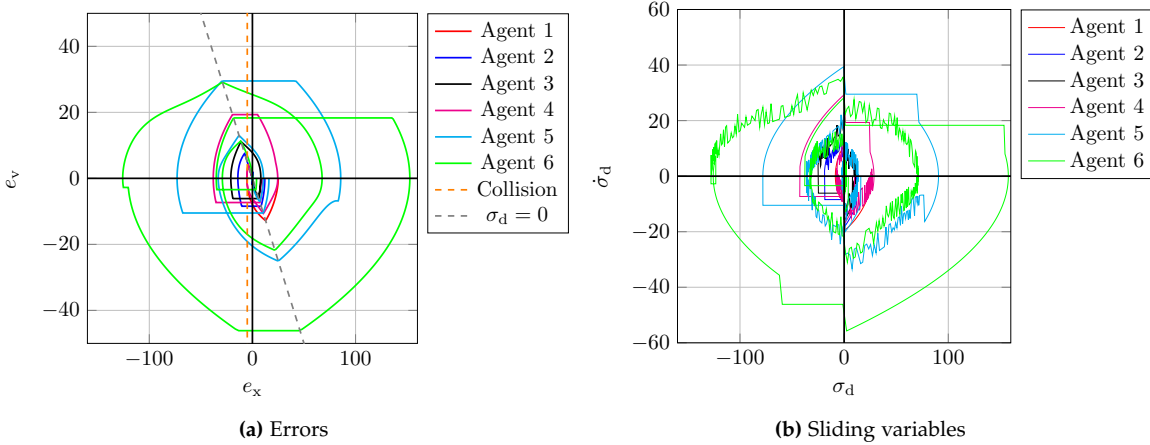


Figure 6.8.: Phase planes using the FOSMC with constant distance sliding surface of relative degree one

However, note that sliding is only possible in the case of perfect communication; if the communicated acceleration of the preceding vehicle is affected by a time-delay denoted τ , then the control law (6.35) becomes

$$u_i(t) = (v_{i-1}(t) - v_i(t)) + u_{i-1}(t - \tau) + k \text{sign}(\sigma_{d,i}(t)), \quad (6.37)$$

which yields the dynamics

$$\dot{\sigma}_{d,i} = u_{i-1}(t) - u_{i-1}(t - \tau) - k \text{sign}(\sigma_{d,i}). \quad (6.38)$$

Note, however, that due to the sign function in u_{i-1} the difference $\bar{u}_{i-1} = u_{i-1}(t) - u_{i-1}(t - \tau)$ may be large even for arbitrarily small $\tau > 0$ and is thus not bounded by $|\bar{u}_{i-1}| \leq k$.

6. String Stability in the Context of Sliding Mode Control

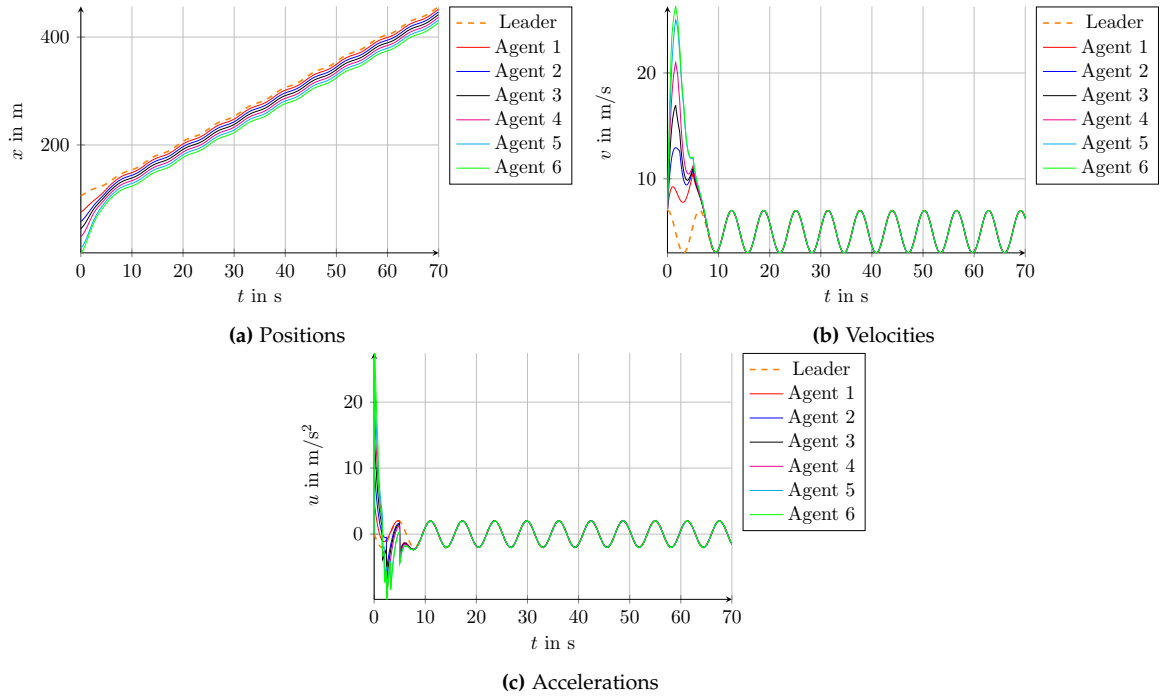


Figure 6.9.: Results using the FOSMC with constant distance sliding surface of relative degree one with feed-forward

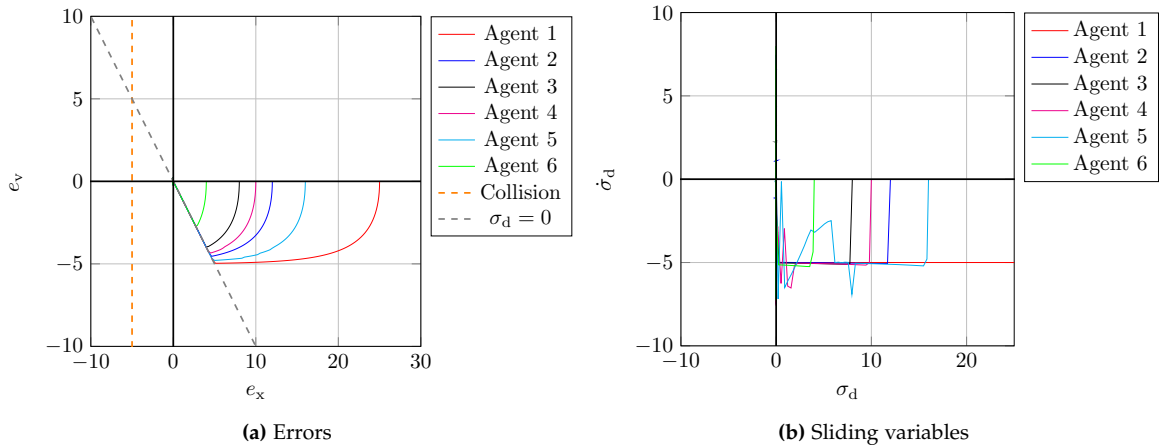


Figure 6.10.: Phase plane using the FOSMC with constant distance sliding surface of relative degree one with feed-forward

Therefore, sliding cannot be guaranteed, although more information than in (6.29) is incorporated. Results are shown in Figures 6.11, 6.12, where position overshoots and thus collisions can be observed due to the time-delay in the feed-forward part (cf. Figure 6.9, 6.10 for corresponding results without time-delay).

Hence, time-delays can seriously degrade the performance of a sliding mode controller. In order to circumvent these problems that arise with a constant distance sliding surface, the constant time-headway spacing is investigated in the following section.

6.2. Constant Distance Sliding Variable

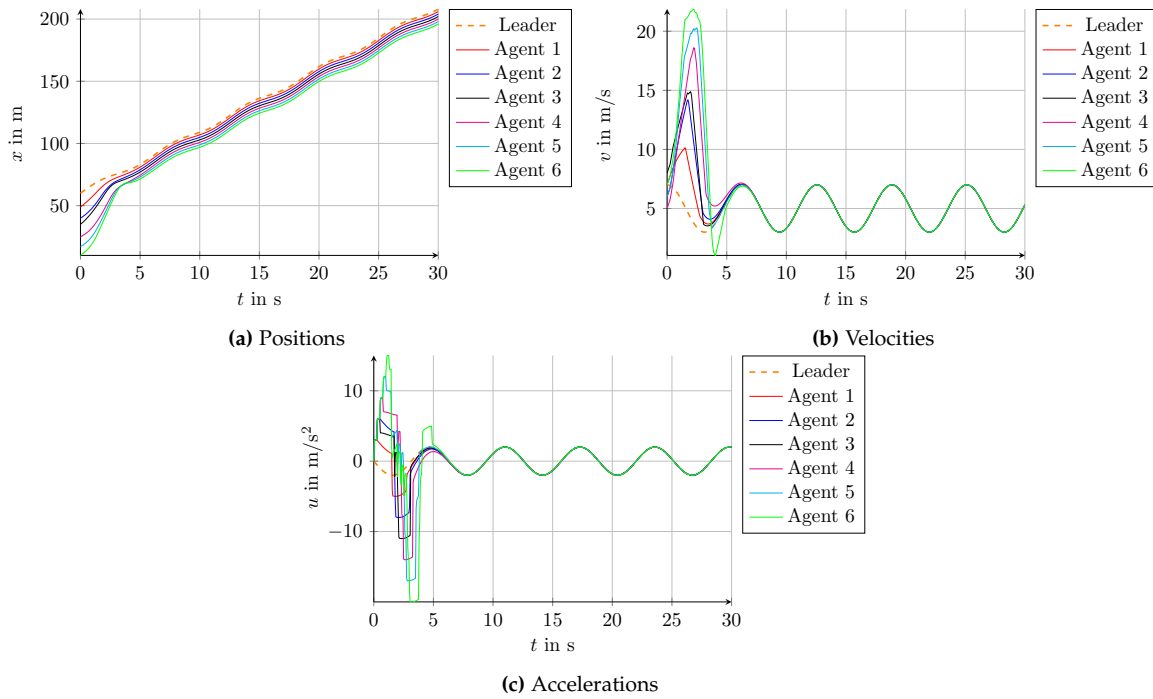


Figure 6.11.: Results using the FOSMC with the constant distance sliding surface of relative degree one and delayed feed-forward

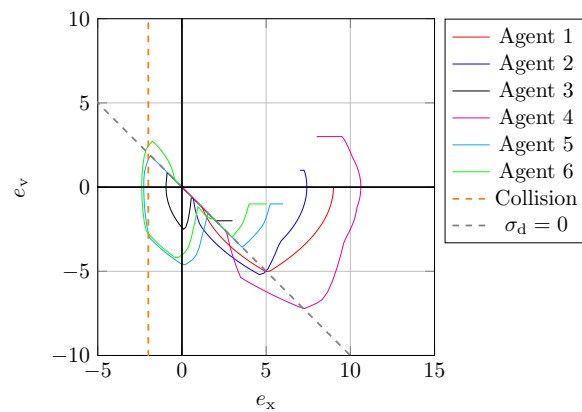


Figure 6.12.: Phase plane of the errors using the FOSMC with the constant distance sliding surface of relative degree one and delayed feed-forward

6.3. Constant Time-Headway Sliding Variable

The constant time-headway spacing has frequently been used for platooning, since it is possible to guarantee string stability with an appropriate controller choice. The sliding variable is defined by the position error (6.7) with an additional velocity-dependent distance as

$$\sigma_{t,i} = e_{x,i} - t_h v_i . \quad (6.39)$$

In sliding, i.e., for $\sigma_{t,i} = 0$, the equilibrium of the position error (6.7) is shifted to $e_{x,i} = t_h v_i$ as shown in Figure 6.13. The positions and velocities are shown in Figure 6.14, where the oscillating behavior of the leading vehicle is damped along the string in both states.

The sliding variable has relative degree one with respect to the control input u_i , and different sliding mode controllers can be used. In the following, a FOSMC, the super-twisting algorithm, and the SOC are discussed.

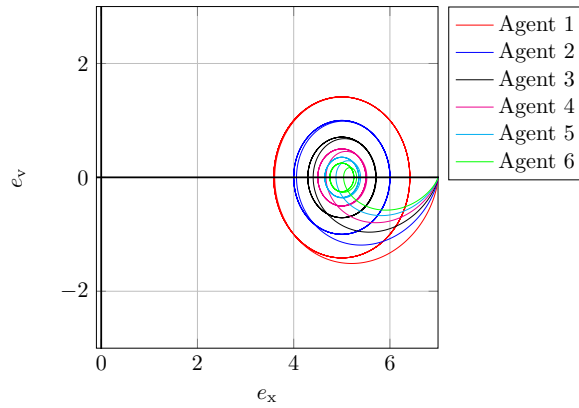


Figure 6.13.: Phase plane of the errors when using a constant time-headway spacing

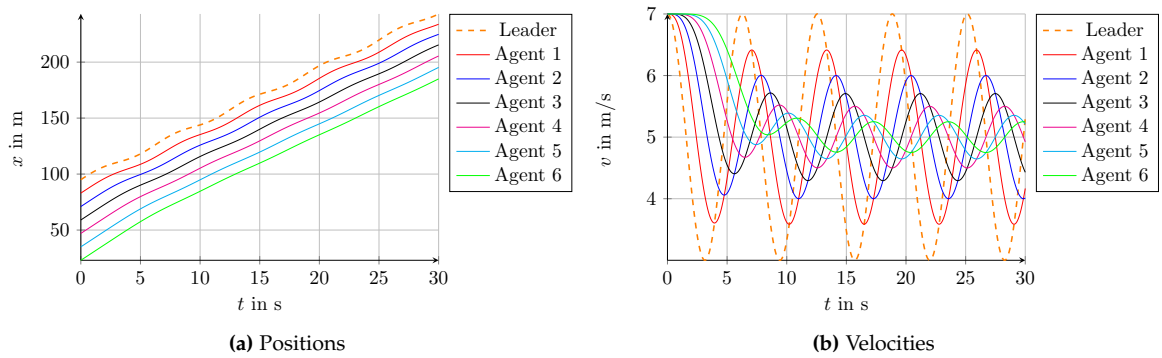


Figure 6.14.: Results using the constant time-headway spacing in sliding ($\sigma_{t,i} = 0$)

First Order Sliding Mode Controller

The first order sliding mode control law reads with (6.39) as

$$u_i = k \text{sign}(\sigma_{t,i}) , \quad (6.40)$$

and the sliding variable dynamics results in

$$\dot{\sigma}_{t,i} = e_{v,i} - t_h k \text{sign}(\sigma_{t,i}) . \quad (6.41)$$

Then, the derivative of the Lyapunov function according to (6.28) is

$$\dot{V}_i = \sigma_{t,i} \dot{\sigma}_{t,i} = \sigma_{t,i} (e_{v,i} - t_h k \text{sign}(\sigma_{t,i})) . \quad (6.42)$$

The control law (6.40) is capable of driving $\sigma_{t,i}$ to zero and can thus maintain sliding if

$$t_h k > |e_{v,i}| \quad (6.43)$$

holds. Then, the discontinuous term $t_h k \text{sign}(\sigma_{t,i})$ dominates $e_{v,i}$, and reaching in finite time can be guaranteed with

$$\dot{V}_i = -\tilde{k} \sigma_{t,i} \text{sign}(\sigma_{t,i}) = -\tilde{k} |\sigma_{t,i}| < 0 , \quad (6.44)$$

with $\tilde{k} = t_h k - e_{v,i}$. For a given upper bound of $|e_{v,i}|$, this can always be guaranteed by appropriate choice of t_h , even though $k > 0$ is bounded by u_{\max} . However, large time-headways are required for large initial spacing errors, which is undesired.

The results using the FOSMC with constant time-headway spacing (6.40) are shown in Figure 6.15, with $k = 3$, $t_h = 1$ and $\Delta_i = 5$ and non-zero initial spacing errors. Agent 1 has a large initial spacing error, whereas the other following agents have small initial spacing errors. Note that the accelerations in the plot have been filtered by a low-pass filter with transfer function

$$F(s) = u_{f,i}(s) / u_i(s) = 1 / (0.01s + 1) . \quad (6.45)$$

One can see that the accelerations along the string are damped in sliding. However, in the reaching phase, the damping effect due to the time-headway is not given, and a large velocity error arises due to the large position error of Agent 1. If overshoot in the sliding variable $\sigma_{t,i}$ could be prevented, then also the position error $e_{x,i}$ would never be negative and hence, collisions could be avoided, as shown by the remaining following agents.

The phase planes of the sliding variable (6.39) and the position error (6.7) are shown in Figure 6.16. For large initial spacing errors, reaching cannot be guaranteed due to violation of (6.43) and hence, a collision occurs for Agent 1 due to the overshoot in $\sigma_{t,1}$. String stability cannot be guaranteed in the reaching phase, while in sliding, the formation is maintained as desired. Also note that the circular trajectories in sliding are caused by the sinusoidal reference acceleration of the leader and the dependence on the velocity using the time-headway (6.39). Due to the damping behavior of the time-headway, the radii of these circles are also decreased with increasing position i in the string.

6. String Stability in the Context of Sliding Mode Control

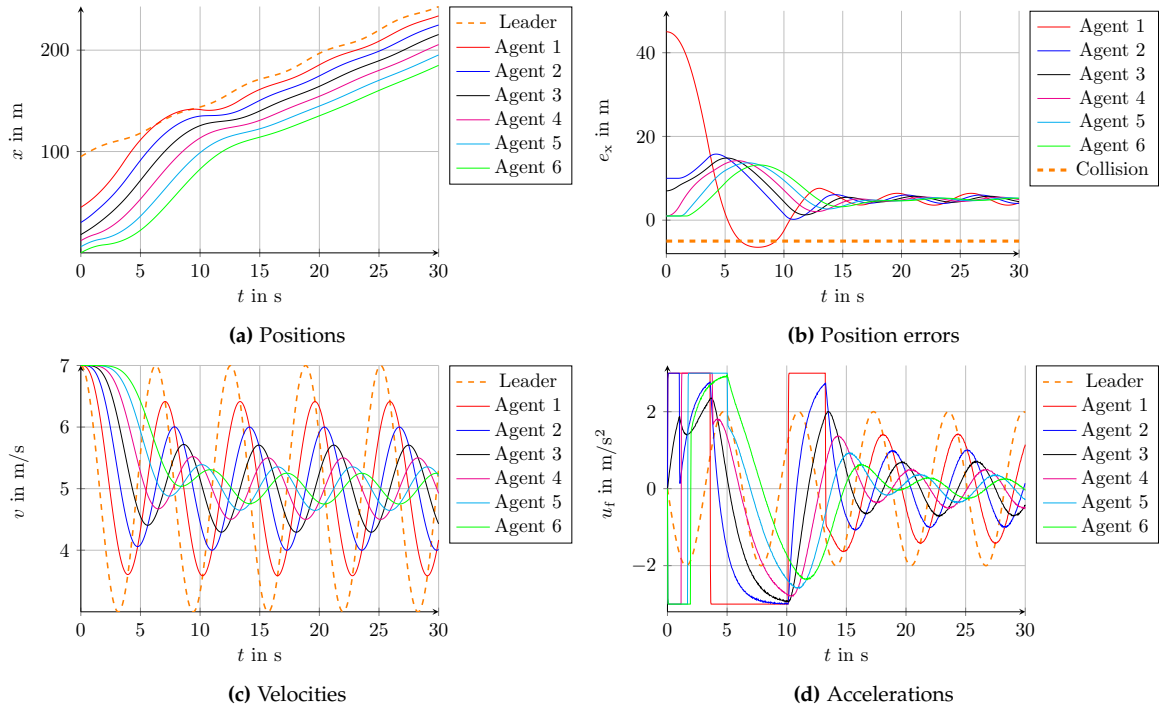


Figure 6.15.: Results using the FOSMC with constant time-headway spacing

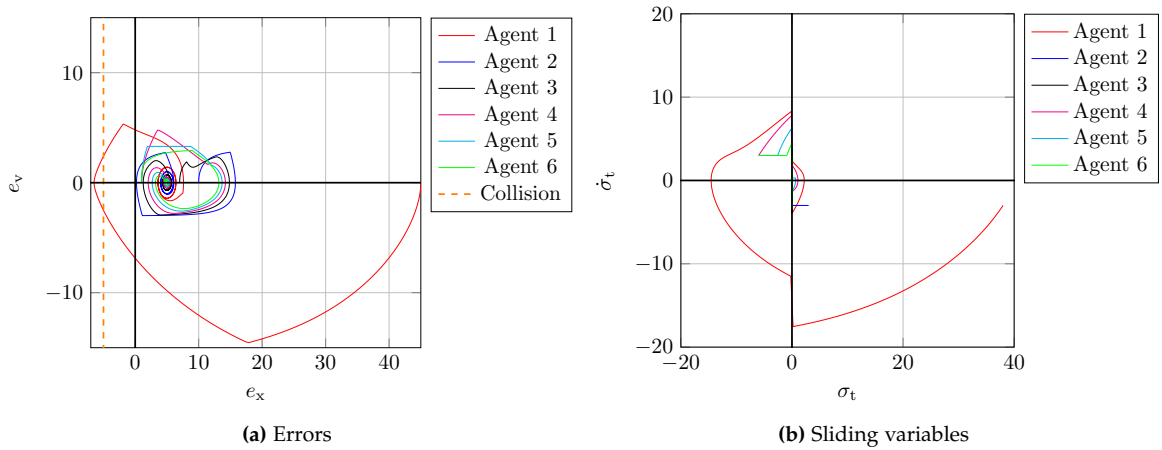


Figure 6.16.: Phase planes using the FOSMC with constant time-headway spacing

Super-Twisting Algorithm

For relative degree one systems, the super-twisting algorithm (STA) is a popular controller due to its continuous control input [71]. The control input is given by

$$u_i(t) = k_1 |\sigma_{t,i}(t)|^{\frac{1}{2}} \text{sign}(\sigma_{t,i}(t)) + \int_0^t k_2 \text{sign}(\sigma_{t,i}(\tau)) d\tau, \quad (6.46)$$

6.3. Constant Time-Headway Sliding Variable

with the constant time-headway sliding variable $\sigma_{t,i}$ defined in (6.39) and an appropriate choice of parameters k_1, k_2 . This control law yields the sliding dynamics

$$\dot{\sigma}_{t,i} = e_{v,i} - t_h \left(k_1 |\sigma_{t,i}|^{\frac{1}{2}} \text{sign}(\sigma_{t,i}) + \int_0^t k_2 \text{sign}(\sigma_{t,i}) d\tau \right). \quad (6.47)$$

Similar to the first order sliding mode controller, $e_{v,i}$ has to be compensated by appropriate choice of the parameters t_h, k_1, k_2 . The results in Figure 6.17 have been achieved with $k_1 = 15, k_2 = 10$, and the phase planes are shown in Figure 6.18.

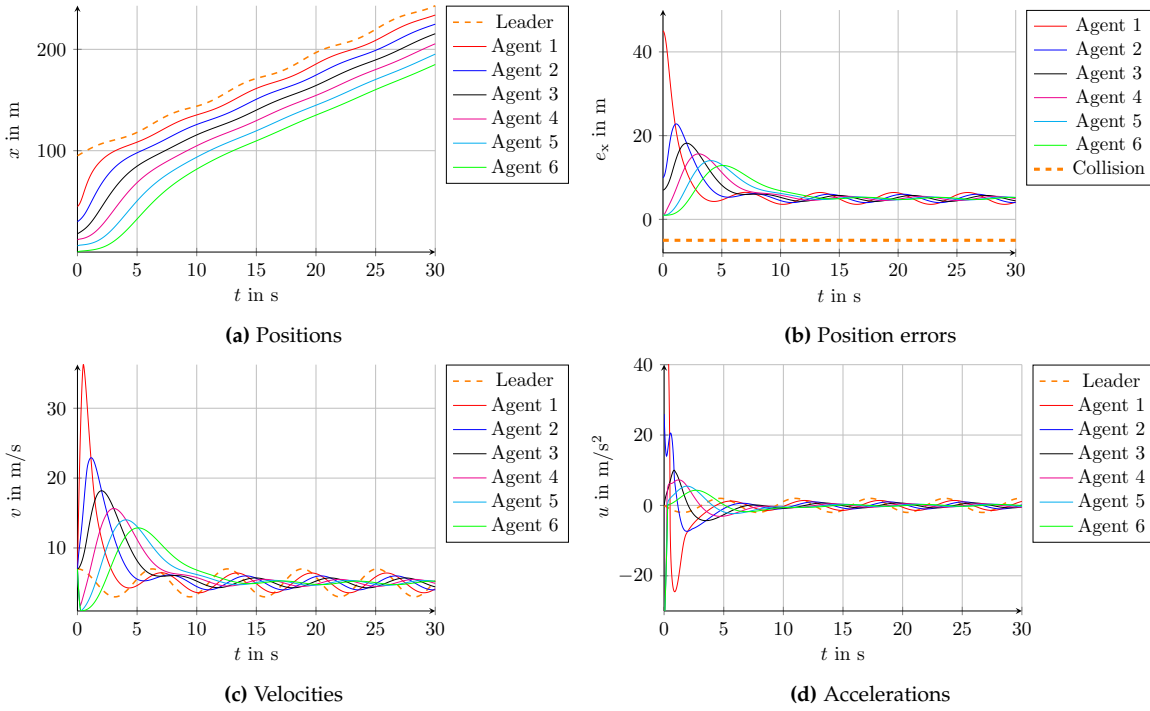


Figure 6.17.: Results using the STA with constant time-headway spacing with $t_h = 1$

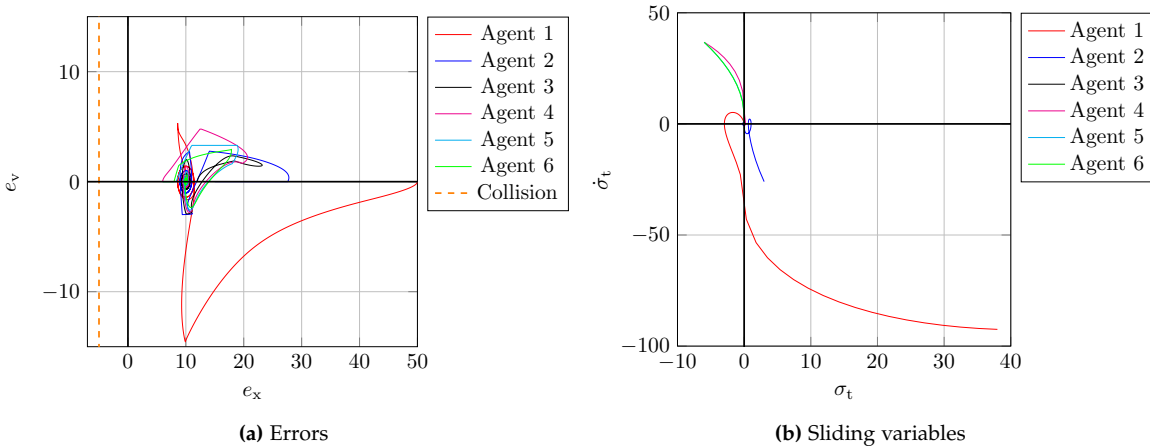


Figure 6.18.: Phase planes using the STA with constant time-headway spacing with $t_h = 1$

6. String Stability in the Context of Sliding Mode Control

The accelerations using the STA are continuous, but depend on the initial errors; the acceleration of the first agent is large due to large initial spacing errors. The accelerations of the remaining agents are amplified along the string. Again, string stability is only achieved in sliding, and non-zero initial spacing errors can lead to collisions due to undershoot in the position errors. Furthermore, attenuation of the relevant states cannot be guaranteed in the reaching phase. Nevertheless, if the time-headway is large enough, the safety distance is not violated.

The distance must always be larger than or equal to the desired distance Δ_i . This works for a certain critical time-headway similar to the linear controllers, $t_h \geq t_{crit}$, for given parameters k_1, k_2 and given initial spacing errors $e_{x,i}(t_0), e_{v,i}(t_0)$. For different initial errors, the critical value of t_h changes and string stability cannot be guaranteed for arbitrary t_h as in Figure 6.19, which shows undesired platooning behavior for $t_h = 0.4$. Note that the time-headways cannot be made arbitrary small, even if the initial errors are zero. However, in contrast to linear control, frequency domain analysis to find proper parameters is not possible and the parameter choice based on initial spacing errors is not practical.

Due to the fact that the magnitude of the position error undershoot is difficult to compute or restrict, the requirement as stated in (6.10) is formulated: in order to guarantee string stability similar to the linear case, the position error (6.7) has to be non-negative for all times in the case of non-negative initial spacing errors $e_{x,i}(t_0) \geq 0$.

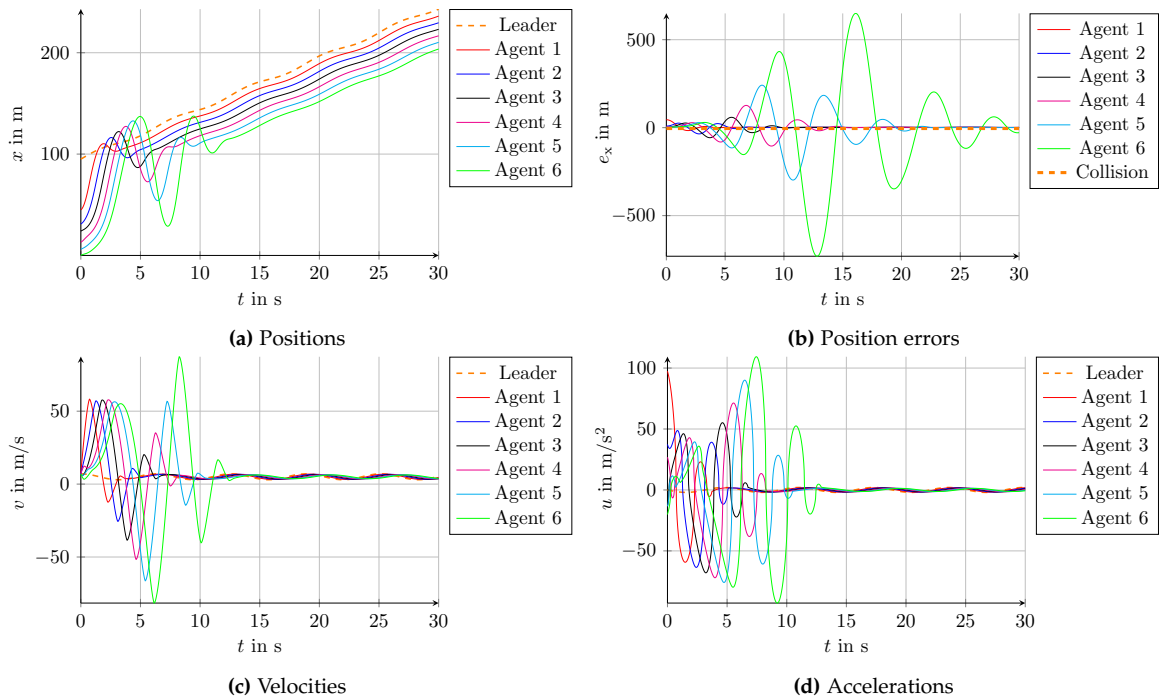


Figure 6.19.: Results using the STA with constant time-headway spacing with $t_h = 0.4$

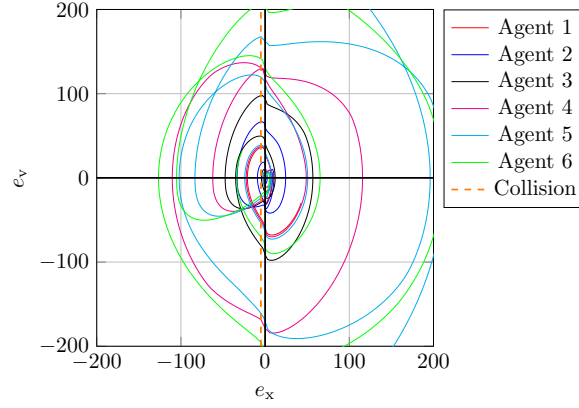


Figure 6.20.: Phase plane of the errors using the STA with constant time-headway spacing with $t_h = 0.4$

Suboptimal Controller

In order to avoid overshoot in the sliding variable, the SOC for the constant time-headway spacing (6.39) is used, which is given by

$$\dot{u}_i = \alpha_i(t)k_{SO}\text{sign}(\sigma_{t,i} - \beta\sigma_{M,t,i}) . \quad (6.48)$$

Note that due to the fact the the sliding variable $\sigma_{t,i}$ is of relative degree one and the SOC is for relative degree two systems, the time-derivative \dot{u}_i of the acceleration is used as auxiliary control input. The acceleration for the agents (6.4) is then computed by integration of the auxiliary control input \dot{u}_i .

As previously discussed for the constant distance spacing using the SOC in (6.16), the parameter $\alpha_i \geq 1$ can be adapted such that monotonic convergence for the sliding variable can be guaranteed. The parameter k_{SO} has to dominate the disturbances and has to be chosen larger than the disturbances' maximum amplitude, and $0.5 \leq \beta < 1$ is a convergence parameter. $\sigma_{M,t,i}$ is the value of the sliding variable at the last instance when the derivative $\dot{\sigma}_i$ was zero (details on the suboptimal algorithm can be found in [32], [33]).

The sliding dynamics with the constant time-headway sliding variable (6.39) and the control input (6.48) are governed by

$$\ddot{\sigma}_{t,i} = u_{i-1} - u_i - t_h \left(\alpha_i(t)k_{SO}\text{sign}(\sigma_{t,i} - \beta\sigma_{M,t,i}) \right) . \quad (6.49)$$

If the condition

$$|u_{i-1} - u_i| < t_h k_{SO} \quad (6.50)$$

is fulfilled, then reaching and sliding can be guaranteed. Simulation results using the SOC (6.48) are shown in Figures 6.21 and 6.22 with $k = 5$, $\beta = 0.7$, $\alpha_i = 1$, $t_h = 1$, where the initial spacings have been chosen small so that reaching can be achieved.

Note that monotonic convergence cannot be guaranteed with the constant α_i , and there is a small undershoot in the sliding variables of Agents 1 and 2. However, since the initial

6. String Stability in the Context of Sliding Mode Control

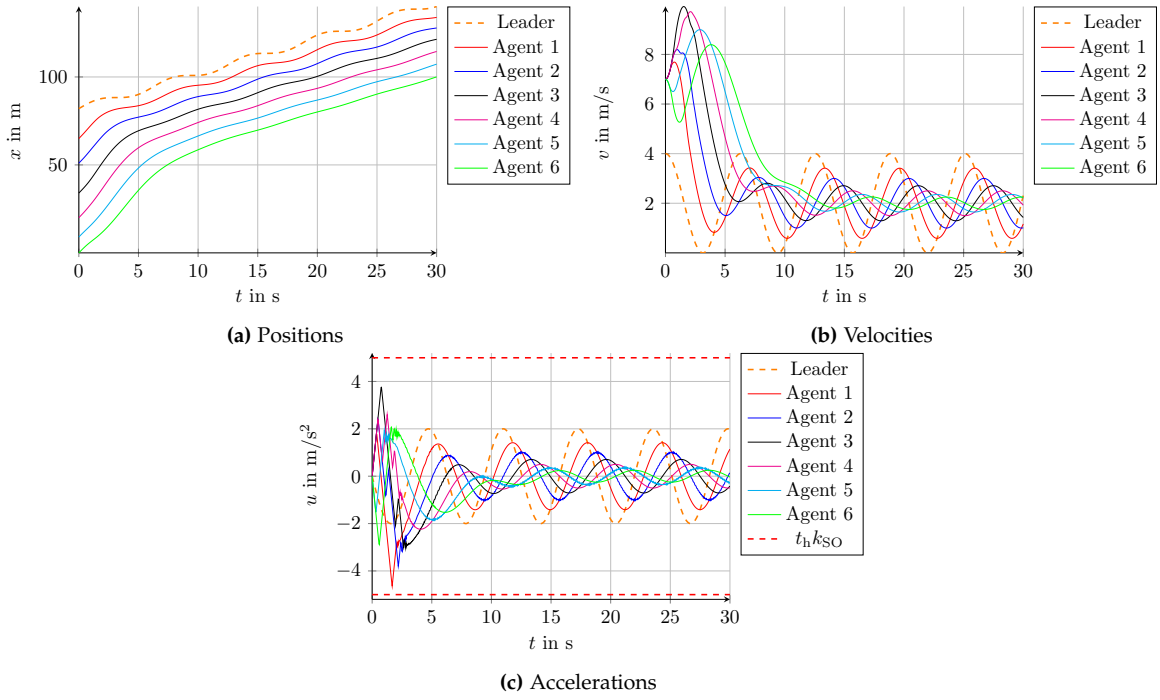


Figure 6.21.: Results using the SOC with constant time-headway spacing and small initial spacing errors

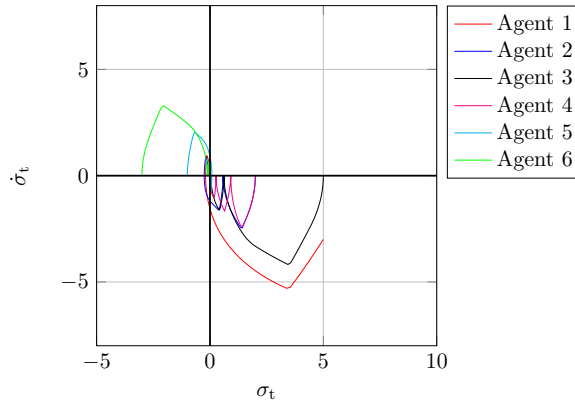


Figure 6.22.: Phase plane of the sliding variables using the SOC with constant time-headway spacing and small initial spacing errors

spacing errors are small, no collisions occur and eventually, sliding can be reached and maintained.

Then in sliding, string stability can be guaranteed, as has been presented in [66]. Therein, it is assumed that the system is in sliding for all times, i.e., the initial errors are zero. The position errors, velocities and accelerations are thus attenuated due to the time-headway in the sliding variable (6.39), and sliding results in filtering behavior with

$$0 = x_{i-1} - x_i - \Delta_i - t_h v_i, \quad (6.51)$$

and differentiation yields

$$0 = v_{i-1} - v_i - t_h u_i, \quad (6.52)$$

which can be written in frequency domain as

$$V_{i-1} = V_i(1 + t_h s). \quad (6.53)$$

If sliding is maintained for all times, then the transfer function

$$G(s) = \frac{V_i(s)}{V_{i-1}(s)} = \frac{1}{t_h s + 1} \quad (6.54)$$

can be established as in Remark 7 of the previous Chapter. The platoon is string stable since

$$\|G\|_\infty = \max_{\omega} |G(j\omega)| \leq 1 \quad (6.55)$$

holds. Note that this is valid only in sliding.

However, there are several issues that arise when dealing with large non-zero initial spacing errors that will be discussed subsequently. The effect of a large spacing error e_1 is shown in Figures 6.23, 6.24. Reaching cannot be guaranteed for Agent 1 since the accelerations violate (6.50) as shown in Figure 6.23c, and eventually collisions occur. The other vehicles, however, are able to reach and maintain sliding. Hence, the vehicles in sliding can avoid amplification of the velocities and accelerations, but no statement on the reaching phase can be made.

The adaptive parameter $\alpha_i(t)$ can be tuned such that monotonic convergence of $\sigma_{t,i}$ is guaranteed, i.e., there is no overshoot in the sliding variable and condition (6.50) can be achieved. In Figures 6.25, 6.26, the parameter α_i is time-varying and adapted as proposed in [32], [33]. However, increasing alpha will also increase the control input \dot{u}_i , yielding an amplification in accelerations during the reaching phase.

For the problem at hand, it has been assumed in [66] that the accelerations u_{i-1} and u_i are bounded, which is true in sliding due to the attenuation of the relevant states according to (6.54). With non-zero initial spacing errors and the requirement that monotonic convergence has to be guaranteed, the auxiliary control inputs \dot{u}_i may increase, which results in arbitrarily large accelerations u_i in the reaching phase. If a saturation is introduced, i.e., the agents' accelerations are limited to $|u_i| \leq u_{\max}$, windup effects may occur due to the integration of the auxiliary variable \dot{u}_i . In Figures 6.27, 6.28 the windup effects are shown for accelerations bounded by $u_{\max} = 2$. Note that condition (6.50) is fulfilled with $|u_{i-1} - u_i| \leq 2u_{\max} \leq t_h k = 5$, but due to windup of the integrator, sliding along the surface cannot be maintained.

Thus, an anti-windup (AW) measure is required. One possibility is to use back-calculation in the same way as proposed for PID controllers, see [30]. The control law in this case is given by

$$\dot{u}_i = -\gamma(u_i - u_{\max}) + \alpha_i(t)k_{\text{SO}}\text{sign}(\sigma_{t,i} - \beta\sigma_{M,t,i}), \quad (6.56)$$

6. String Stability in the Context of Sliding Mode Control

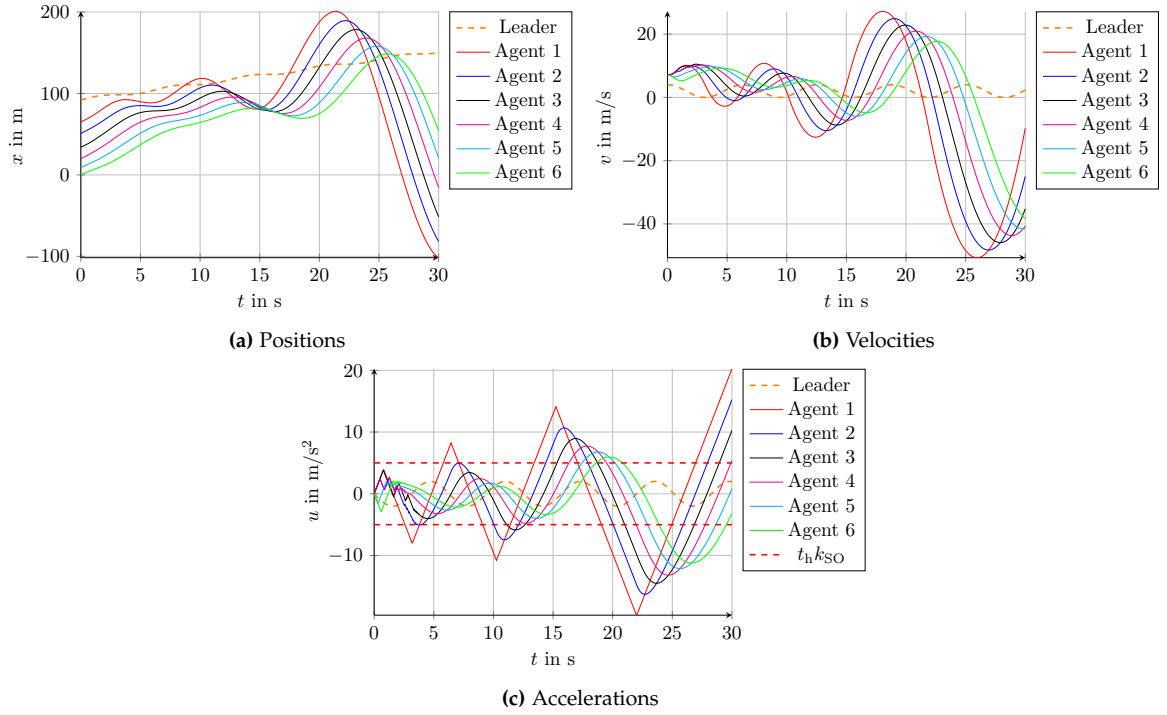


Figure 6.23.: Results using the SOC with constant time-headway spacing and large initial spacing errors

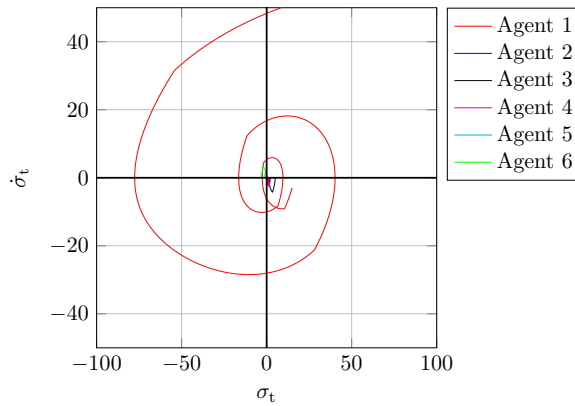


Figure 6.24.: Phase plane of the sliding variables using the SOC with constant time-headway spacing and large initial spacing errors

which results in the sliding variable dynamics

$$\ddot{\sigma}_{t,i} = u_{i-1} - u_i + t_h \gamma u_i - t_h \gamma u_{\max,i} - t_h \left(\alpha_i(t) k_{SO} \text{sign}(\sigma_{t,i} - \beta \sigma_{M,t,i}) \right). \quad (6.57)$$

Note that additional terms appear in this dynamics. Results with back-calculation are shown in Figure 6.29 for $u_{\max} = 2$. The parameter $k = 2.5u_{\max}$ is chosen in order to satisfy condition (6.50), and $\gamma = k_{SO}$ has been found in simulations. The AW measure affects the convergence of the sliding variable, which can be observed in the phase plane of the sliding variable in Figure 6.30. The parameter choice of γ , however, is not clear, and further analysis is required in future work for the platooning application.

6.3. Constant Time-Headway Sliding Variable

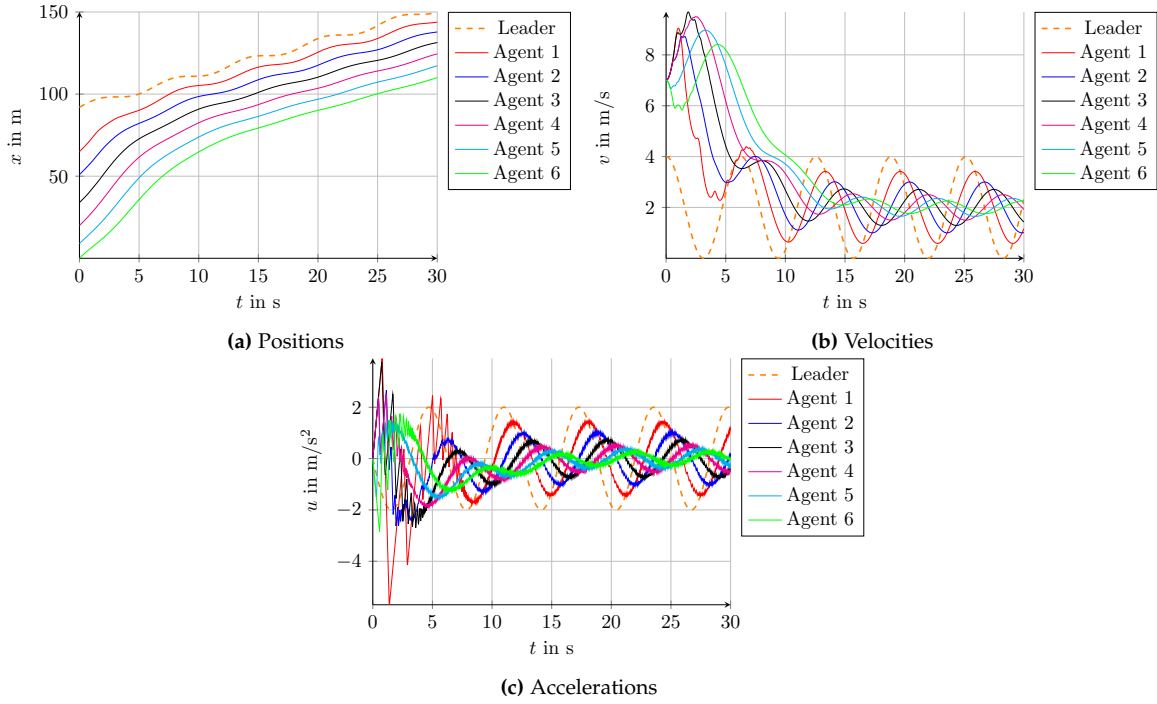


Figure 6.25.: Results using the SOC with constant time-headway spacing and adaptive α_i for monotonic convergence

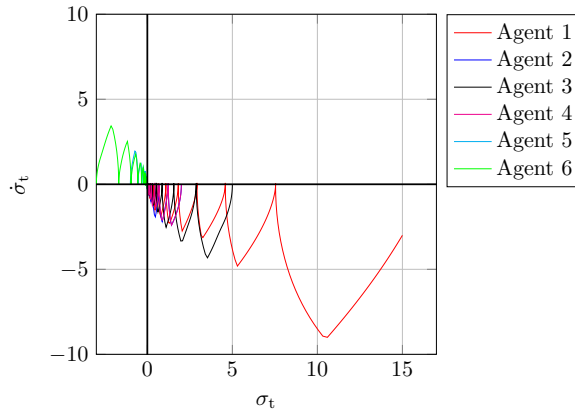


Figure 6.26.: Phase plane of the sliding variables using the SOC with constant time-headway spacing and monotonic convergence

It can thus be concluded that the SOC with the constant time-headway spacing without anti-windup measure (6.48) is not suitable for string stable vehicle-following with arbitrary non-zero initial spacing errors.

If the acceleration of the preceding vehicle is communicated, the control law can be designed as

$$\dot{u}_i = \frac{1}{t_h} (u_{i-1} - u_i) + \alpha_i(t) k_{SO} \text{sign}(\sigma_{t,i} - \beta \sigma_{M,t,i}), \quad (6.58)$$

6. String Stability in the Context of Sliding Mode Control

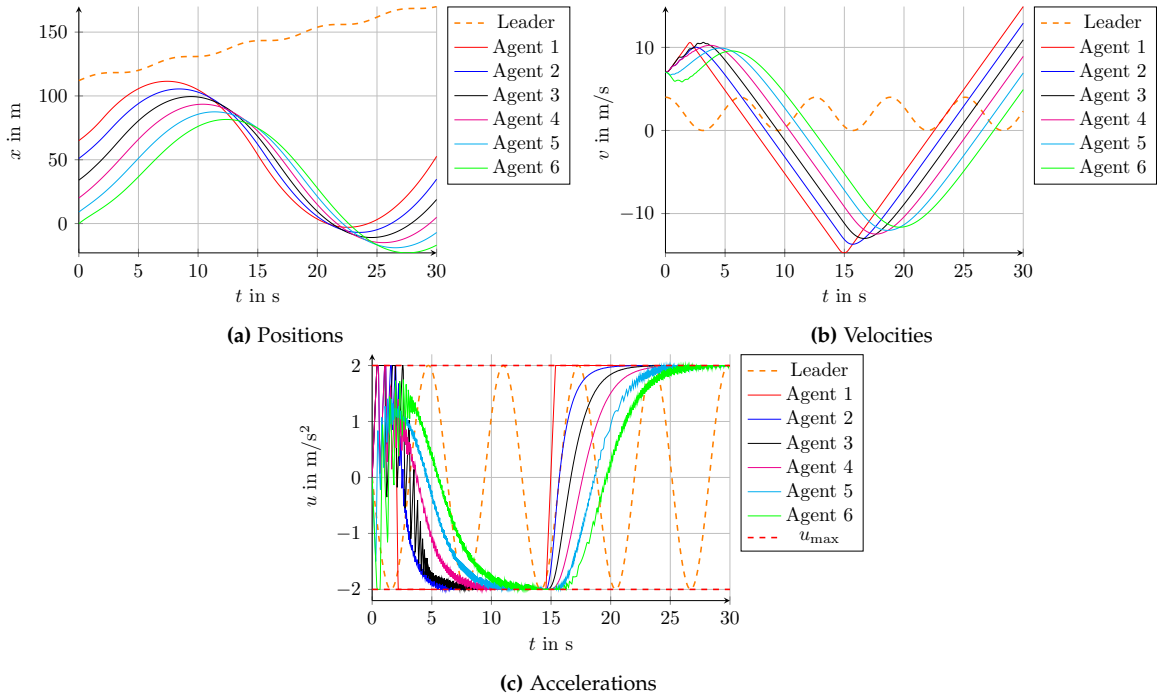


Figure 6.27.: Results using the SOC with constant time-headway spacing and adaptive α_i for monotonic convergence with actuator limits. Windup effects occur due to the additional integrator and an input saturation.

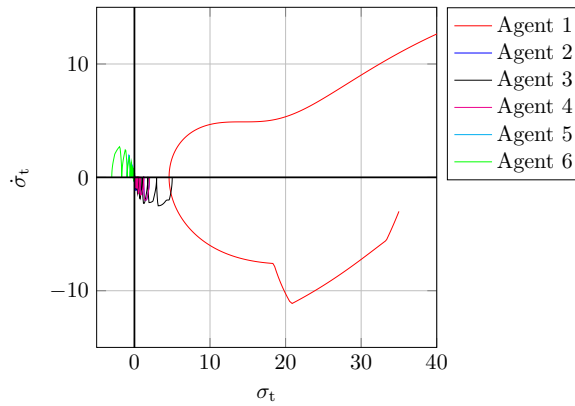


Figure 6.28.: Phase plane of the sliding variables using the SOC with constant time-headway spacing and adaptive α_i for monotonic convergence with actuator limits

which yields the sliding dynamics

$$\ddot{\sigma}_{t,i} = -t_h \alpha_i(t) k_{\text{SO}} \text{sign}(\sigma_{t,i} - \beta \sigma_{M,t,i}) . \quad (6.59)$$

Then, reaching and sliding can be guaranteed without using an anti-windup measure . The results with communicated acceleration are shown in Figures 6.31, 6.32 with $u_{\text{max}} = 2$. Note that the windup problem has vanished due to incorporation of the accelerations in (6.58).

6.3. Constant Time-Headway Sliding Variable

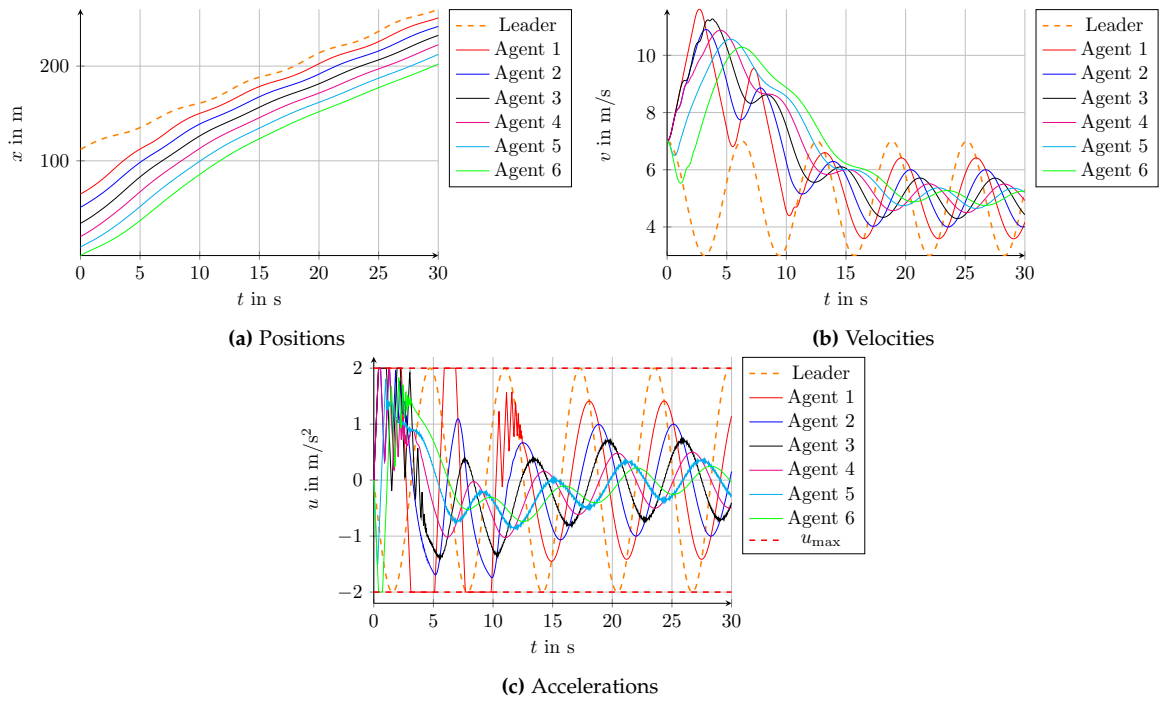


Figure 6.29.: Results using the SOC with constant time-headway spacing and adaptive α_i for monotonic convergence with actuator constraints and AW measure

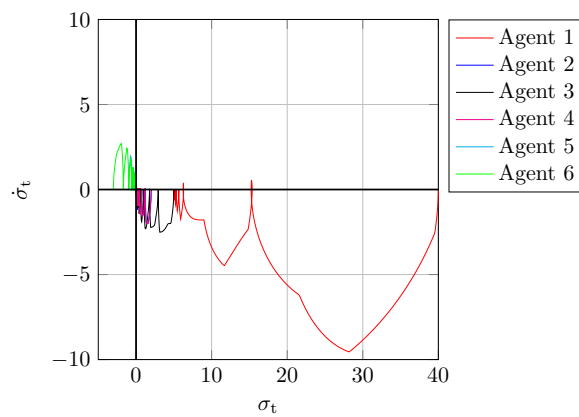


Figure 6.30.: Phase plane of the sliding variables using the SOC with AW measure

6. String Stability in the Context of Sliding Mode Control

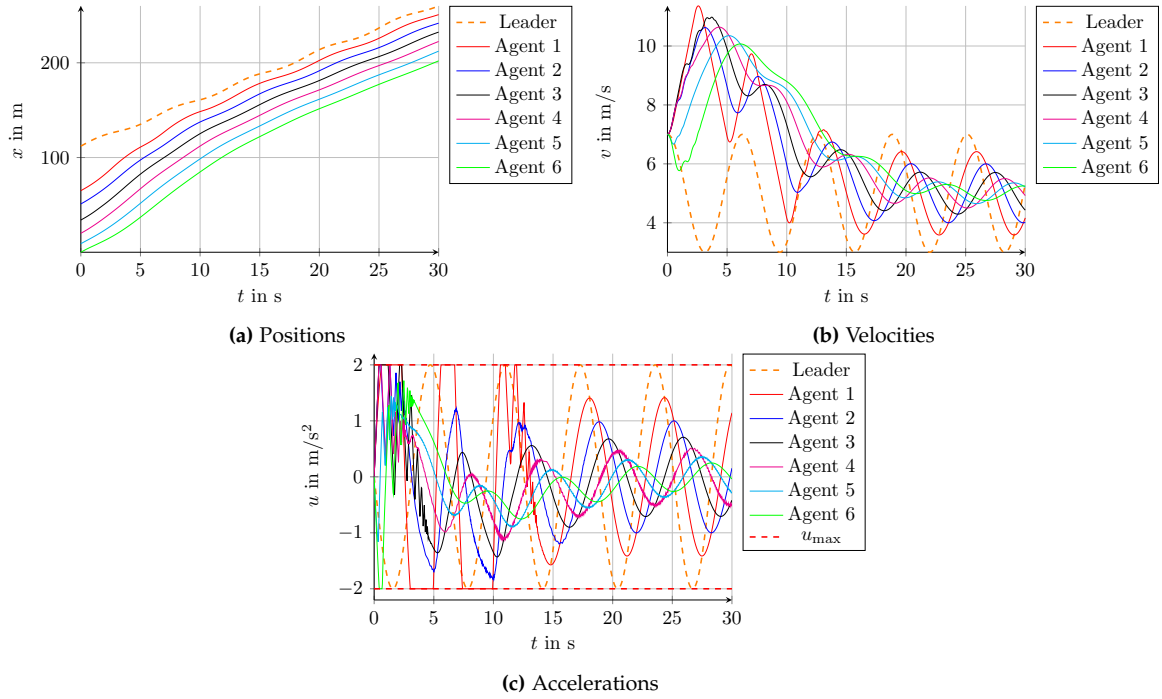


Figure 6.31.: Results using the SOC with constant time-headway spacing and adaptive α_i for monotonic convergence with (perfect) communication

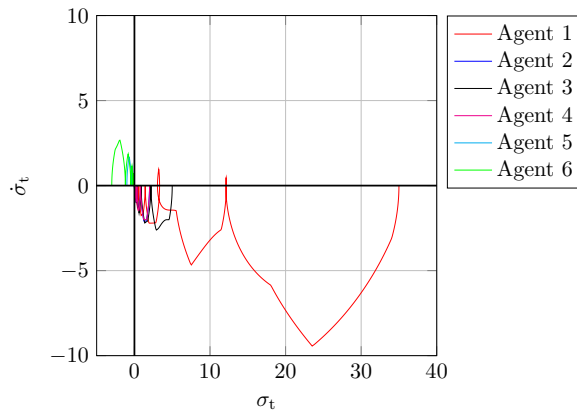


Figure 6.32.: Phase plane of the sliding variables using the SOC with constant time-headway spacing and adaptive α_i for monotonic convergence with (perfect) communication

Example using a FOSMC

To conclude the discussion of the different sliding mode controllers for non-zero initial spacing errors, an example for the constant distance spacing and the constant time-headway spacing with the first order controllers (6.29) resp. (6.40) of agent i in the string is investigated. The leader's reference acceleration is given by $u_r = 2 \sin(t)$.

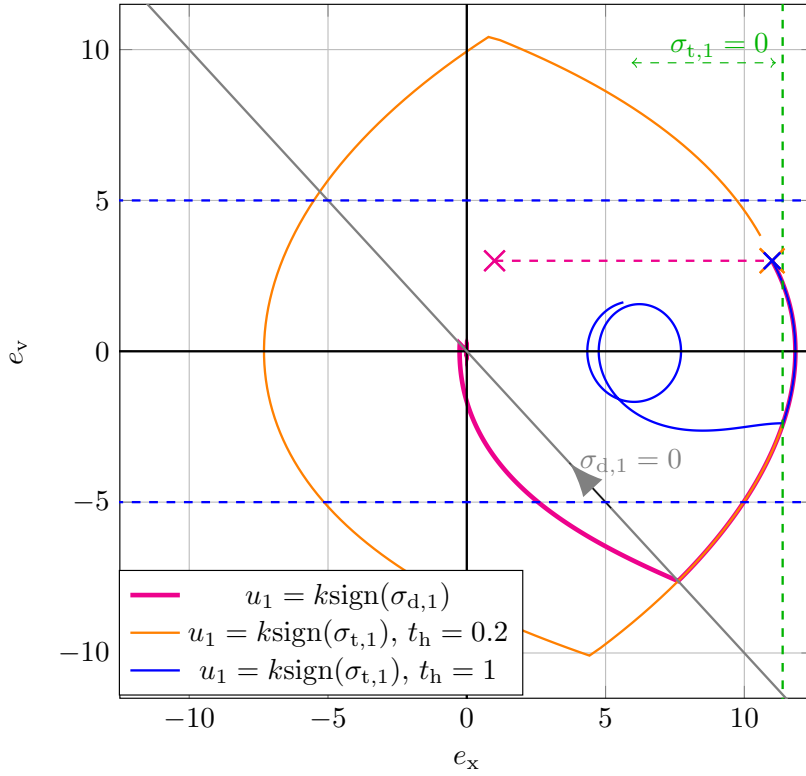


Figure 6.33.: Phase plane of the errors with different sliding surfaces using a FOSMC with $k = 5$

Constant Distance Sliding Variable

First, the constant distance spacing with relative degree one (6.22) with the controller (6.29) is considered. The initial position and velocity errors are small and positive, $e_{x,i}(t_0) = 1$, $e_{v,i}(t_0) = 3$, denoted by the magenta cross in Figure 6.33, which shows the phase plane of the errors for Agent 1, i.e., the first following vehicle. Then, if both the predecessor $i - 1$ and the agent i apply the maximum control input $u_{i-1} = u_i = k = 5$, the velocity error (6.8) remains constant due to $\dot{e}_{v,i} = 0$. Then, the position error increases with $\dot{e}_{x,i} = e_{v,i} > 0$, as indicated by the magenta dashed line. The constant distance sliding variable (6.22) increases until the predecessor stops to apply the maximum acceleration at time t_s , which yields the error values $e_{x,i}(t_s) = 11$, $e_{v,i}(t_s) = 3$ in Figure 6.33.

Then, the system is steered (magenta solid line) towards the sliding surface $\sigma_{d,i} = 0$ (gray line). However, since the condition to maintain sliding (6.34) is not satisfied when the

6. String Stability in the Context of Sliding Mode Control

sliding surface is reached at $e_{x,i}(t_s) = 7.6$, $e_{v,i}(t_s) = -7.6$, sliding cannot be maintained and a small position overshoot occurs. Eventually, the desired distance Δ_i can be reached.

Note that small initial spacing errors can be increased unboundedly due to a non-zero velocity error, which can result in collisions. Bounded initial position errors are thus not sufficient to guarantee collision-free platooning.

Constant Time-Headway Sliding Variable

If the constant time-headway spacing (6.39) is used for initial errors $e_{x,i}(t_s) = 11$, $e_{v,i}(t_s) = 3$ as indicated by the blue cross, position overshoot can be avoided in reaching (solid blue line). Condition (6.43) is satisfied for all times, where the bounds are indicated by the blue dashed lines for $t_h = 1$. Since the time-headway is large enough, reaching is guaranteed and sliding can be maintained. In sliding, the equilibrium of the position error is shifted with the velocity of the vehicle. The constant time-headway sliding surface $\sigma_{t,i} = 0$ is depicted by a green dashed vertical line in Figure 6.33 that moves within a certain range of the phase plane; the velocities of the vehicles change due to a sinusoidal reference acceleration.

However, for the same initial spacing errors, a small time-headway $t_h = 0.2$ violates the reaching condition (6.43), and reaching and sliding cannot be guaranteed as shown by the orange line in Figure 6.33. The small time-headway then results in a negative position error, which can lead to collisions, and the performance may be even worse than with the constant distance spacing. The time-headway thus has to be chosen large, which also means that the distances in sliding are large. A trade-off between safety and efficiency has to be found, but it is not clear how t_h should be chosen such that reaching can be guaranteed for arbitrary non-zero initial errors.

In summary, it can be concluded that all spacing policies work well in sliding, but the reaching phase is problematic regarding amplification of accelerations and position error overshoots. String stability cannot be guaranteed for non-zero initial spacing errors using the standard approaches. Other sliding mode based techniques, such as the controller proposed by Bhat and Bernstein [42] with discontinuous control extension as in [52], integral sliding mode [127] or a sliding sector approach [123], lead to similar problems and no approach can guarantee both monotonic convergence and limited accelerations. Although communication improves the reaching performance and guarantees monotonic convergence, bounded accelerations cannot be considered at the same time. To remedy these problems, a sliding surface is developed in the following section that combines the benefits of constant distance and constant time-headway spacing: a small inter-vehicle distance is maintained collision-free.

6.4. Adaptive Time-Headway Sliding Variable

A sliding mode based controller that fulfills all requirements presented in Section 6.1.1 has been proposed by the author of this thesis in [143] for platooning with an adaptive spacing policy, which is called adaptive time-headway formation controller (ATFC) subsequently.

6.4.1. Adaptive Time-Headway Formation Controller - ATFC

A novel combination of two sliding surfaces with decreasing time-headway is used to finally reach a constant distance spacing without collisions for non-zero initial errors. It is assumed that only the predecessor's information is available. In order to maintain small constant inter-vehicle distances to improve traffic throughput, the aim of the proposed approach is to decrease the time-headway in (6.39). However, note that $t_h \rightarrow 0$ means loss of influence on the sliding dynamics (6.41). Thus, a new sliding variable is proposed, combining the constant distance (6.22) and constant time-headway spacing (6.39). With this approach, the time-headway $t_{h,i}(t)$ in the new sliding variable can be decreased to zero asymptotically and a constant distance formation can be achieved. Specifically, the behavior is eventually approximated by the desired error dynamics

$$\dot{e}_{x,i} = -\frac{1}{t_h^*} e_{x,i}, \quad t_h^* > 0, \quad (6.60)$$

where t_h^* is a desired convergence time constant. For this purpose, the following assumption is made.

Assumption 2. *The initial spacing errors are bounded and depend on the velocity of the vehicle such that $0 \leq e_{x,i}(t_0) \leq t_{\max,i} v_i(t_0)$, where $t_{\max,i} \geq 0$ is a parameter that can be specified. The maximum velocity error is bounded such that $|e_{v,i}(t_0)| < t_{h,i}(t_0) u_{\max}$ with*

$$t_{h,i}(t_0) = \begin{cases} e_{x,i}(t_0)/v_i(t_0) & \text{if } e_{x,i}(t_0) > t_h^* v_i(t_0) \text{ (Phase I)} \\ (e_{x,i}(t_0) + t_h^* e_{v,i}(t_0))/v_{i-1}(t_0) & \text{otherwise (Phase II)}. \end{cases} \quad (6.61)$$

Remark 9. *Assumption 2 on the one hand excludes vehicles that are either far behind or too slow to follow up with the platoon, which is hence no restriction. On the other hand, vehicles may not be close to the predecessor and/or too fast relative to preceding vehicles so that collisions occur regardless of the controller used. In practice, these situations must be avoided by a higher-level unit, e.g., the behavioral planning level that switches the reference generation on and off. It is hence reasonable to assume these conditions for platooning.*

Remark 10. *The condition in (6.61) is based on the position error due to the fact that two different phases and thus two different sliding variables are proposed. The initialization is performed so that the sliding variable of the active phase is zero, and hence the time-headway is computed according to the current phase. The phases depend on the parameter t_h^* in (6.60), as will be described in the following theorem.*

Theorem 1. [143] *Consider the interconnected system (6.3), (6.4) and the errors (6.7) with Assumptions 1 and 2. Let the control law u_i consists of two phases with a specified switching condition. In Phase I, the control law*

$$u_i = k \text{sign}(\sigma_{t,i}), \quad (6.62)$$

with the sliding variable

$$\sigma_{t,i} = e_{x,i} - t_{h,i} v_i, \quad (6.63)$$

6. String Stability in the Context of Sliding Mode Control

and an adaptive law

$$\dot{t}_{h,i} = -\mu_1 \frac{t_h^* k}{v_i}, \quad (6.64)$$

$$\mu_1 = \begin{cases} 1 & -\frac{t_h^* k}{2} \leq e_{v,i} < 0 \text{ and } t_{h,i} > t_h^* \\ 0 & \text{otherwise,} \end{cases} \quad (6.65)$$

is applied, with t_h^* defined in (6.60). The initial time-headway is computed as

$$t_{h,i}(t_0) = \frac{e_{x,i}(t_0)}{v_i(t_0)}, \quad (6.66)$$

and satisfies $t_{h,i}(t_0) < t_{\max,i}$ according to Assumptions 1 and 2.

In Phase II, control law

$$u_i = k \text{sign}(\tilde{\sigma}_i), \quad (6.67)$$

with the combined sliding variable

$$\tilde{\sigma}_i = e_{x,i} - t_{h,i} v_i + (t_h^* - t_{h,i}) e_{v,i}, \quad (6.68)$$

and an adaptive law

$$\dot{t}_{h,i} = -\mu_2 \frac{k}{v_{i-1}} t_{h,i}, \quad \mu_2 = \begin{cases} 1 & -\frac{t_{h,i} k}{2} \leq e_{v,i} < 0 \\ 0 & \text{otherwise} \end{cases} \quad (6.69)$$

is used. The switching time $t_{s,i}$ for agent i from Phase I to Phase II is given by the time instant for which

$$\tilde{\sigma}_i(t_{s,i}) = 0, \quad \sigma_{t,i}(t_{s,i}) = 0, \quad t_{h,i}(t_{s,i}) \leq t_h^* \quad (6.70)$$

holds. Then, the interconnected system fulfills all design goals presented in Section 6.1.1.

Proof of Phase I. The dynamics of the sliding variable (6.63) with errors defined in (6.7), and control input (6.62) with (6.64), reads as

$$\dot{\sigma}_{t,i} = e_{v,i} - t_{h,i} k \text{sign}(\sigma_{t,i}) + \mu_1 t_h^* k. \quad (6.71)$$

With the proposed initialization of the time-headway (6.66), starting on the sliding surface is guaranteed with $\sigma_{t,i}(t_0) = 0$. The derivative of the Lyapunov function as in (6.28) with (6.71) results in

$$\dot{V}_i = \sigma_{t,i} \dot{\sigma}_{t,i} = \sigma_{t,i} (e_{v,i} - t_{h,i} k \text{sign}(\sigma_{t,i}) + \mu_1 t_h^* k). \quad (6.72)$$

For a negative derivative of the Lyapunov function, the following conditions can be stated: for a positive sliding variable $\sigma_{t,i} > 0$, it has to be guaranteed that $\dot{\sigma}_{t,i} < 0$, and vice versa. For a positive sliding variable, one can distinguish between two cases for the velocity error: first, if the velocity error is positive, then $\mu_1 = 0$ according to (6.65) and hence

$$e_{v,i} - t_{h,i} k \text{sign}(\sigma_{t,i}) < 0 \quad (6.73)$$

must be valid, which results in the same condition as in the constant time-headway spacing (6.43), i.e.,

$$|e_{v,i}| < t_{h,i}k, \quad (6.74)$$

which is satisfied at the initial time due to Assumption 2. Moreover, if the velocity error is negative and large so that $\mu_1 = 0$, then (6.74) has to hold. Second, if the velocity error is so that $\mu_1 = 1$, then the condition

$$e_{v,i} - t_{h,i}k + \mu_1 t_h^* k < 0 \quad (6.75)$$

has to be satisfied. Then, since $e_{v,i} < 0$ for $\mu_1 = 1$ and $t_{h,i} > t_h^*$ in Phase 1, this is automatically fulfilled.

If the sliding variable is negative $\sigma_{t,i} < 0$, again two cases for the velocity error must be investigated in order to state that $\dot{\sigma}_{t,i} > 0$ is to be satisfied. First, if $\mu_1 = 0$ due to a positive velocity error $e_{v,i} > 0$, then

$$e_{v,i} + t_{h,i}k > 0 \quad (6.76)$$

is fulfilled without restrictions, since both terms are positive. Second, a negative velocity error leads to the condition that depends on μ_1 , i.e.,

$$e_{v,i} + t_{h,i}k + \mu_1 t_h^* k > 0 \quad (6.77)$$

must hold. These conditions for sliding (6.74), (6.77) can be summarized as

$$-t_{h,i}k - \mu_1 t_h^* k < e_{v,i} < t_{h,i}k. \quad (6.78)$$

The dynamics of the errors (6.8) in sliding are

$$\begin{aligned} \dot{e}_{x,i} &= e_{v,i}, \\ \dot{e}_{v,i} &= \underbrace{u_{i-1} - \frac{\mu_1}{t_{h,i}} t_h^* k}_{\hat{u}_i} - \frac{e_{v,i}}{t_{h,i}}, \end{aligned} \quad (6.79)$$

where \hat{u}_i are bounded inputs,

$$\hat{u}_i \in [-k - \mu_1 t_h^* k, k], \quad (6.80)$$

to a stable linear system. When starting on the sliding surface $\sigma_{t,i} = 0$ with a proper choice of the time-headway as in (6.66), the sliding variable can be maintained at zero although the time-headway is changed. Hence, it has to be guaranteed that the velocity error stays bounded in sliding, so that conditions (6.78) are fulfilled for all times.

Note that the dynamics of the velocity error (6.79) are described by a scalar linear time-varying (LTV) system with bounded input, which means that the velocity error cannot change abruptly. For a short summary on stability analysis of the given linear LTV system, the reader is referred to Appendix B. With $\mu_1 = 1$ as defined in (6.65), the velocity errors are per definition in a subinterval within the bounds (6.78). For $\mu_1 = 0$, the time-headway

6. String Stability in the Context of Sliding Mode Control

is held constant and the velocity errors converge according to (6.79) asymptotically, i.e., $\lim_{t \rightarrow \infty} e_{v,i} = t_{h,i} u_{i-1}$, which is within the bounds (6.78) for all finite times with the initial errors from Assumption 2 and an appropriate choice of the time-headway (6.66). Note, moreover, that the preceding vehicle will not apply maximum acceleration for infinite times due to the fact that $u_r < k$ as discussed in Appendix A and due to the damping behavior of the time-headway as in (6.54).

Then, $\sigma_{v,i} = 0$ can be maintained, which yields positive position errors $e_{x,i} = t_{h,i} v_i$, and negative position errors are avoided as demanded in the design goals in Section 6.1.1. The position and velocity errors (6.79) are bounded and independent of the position in the platoon and independent of the length of the platoon, satisfying (6.11). The accelerations are bounded by design (6.62) with $u_{\max} = k$. \square

Hence all requirements in Section 6.1.1 except the constant distance spacing are satisfied. As soon as the time-headway has been decreased to $t_{h,i} = t_h^*$, Phase II is used to decrease the time-headway to zero asymptotically. However, in the case that the initial time-headway (6.66) is less than the desired value t_h^* , the time-headway is kept constant (6.65). Then, Phase II can be activated as soon as (6.70) is fulfilled, which is the case when $e_{v,i} = 0$.

Proof of Phase II. Due to the design of the switching condition (6.70), the sliding variable of Phase II is also zero at the switching time $t_{s,i}$. With the error dynamics (6.8), the first order sliding mode control law (6.67) and the adaptation law (6.69), the dynamics of the new sliding variables (6.68) are governed by

$$\dot{\tilde{\sigma}}_i = e_{v,i} + (t_h^* - t_{h,i})u_{i-1} - t_h^* k \text{sign}(\tilde{\sigma}_i) + \mu_2 k t_{h,i}. \quad (6.81)$$

Analogously to Phase I, it has to be shown as in (6.28) that $\dot{V}_i = \tilde{\sigma}_i \dot{\tilde{\sigma}}_i < 0$. Thus, it has to be guaranteed that the sliding variable and its derivate have opposite signs, then $\tilde{\sigma}_i = 0$ can be maintained. Again, the sliding surface and its derivative are investigated, where different conditions for the velocity errors can be stated.

For a positive sliding variable $\tilde{\sigma}_i > 0$, the derivative in (6.81) has to be negative for both positive and negative velocity errors, i.e.,

$$e_{v,i} + (t_h^* - t_{h,i})u_{i-1} - t_h^* k + \mu_2 k t_{h,i} < 0, \quad (6.82)$$

where $u_{i-1} = k$ is applied in the worst case, yielding the condition

$$e_{v,i} - t_{h,i} k + \mu_2 t_{h,i} k < 0. \quad (6.83)$$

If $\mu_2 = 0$, then the standard condition for time-headway spacing (6.43), i.e. $|e_{v,i}| < t_{h,i} k$, must hold. In the case that $\mu_2 = 1$, the velocity error is negative by definition (6.69) and hence the derivative of the Lyapunov function is negative.

If the sliding variable is negative, $\tilde{\sigma}_i < 0$, the derivative of the sliding variable must be positive, and the worst case is $u_{i-1} = -k$ in (6.81), which yields the condition

$$e_{v,i} + t_{h,i} k + \mu_2 k t_{h,i} > 0. \quad (6.84)$$

Then, for $\mu_2 = 1$, it has to hold that $|e_{v,i}| < t_{h,i}(1 + \mu_2)k$, which is always satisfied for the given definition of μ_2 in (6.69). For $\mu_2 = 0$, if the velocity error is positive, then

$$e_{v,i} + t_{h,i}k > 0 \quad (6.85)$$

is always satisfied. If the velocity error is negative, but $\mu_2 = 0$ due to the magnitude of the velocity error with $|e_{v,i}| > \frac{1}{2}t_{h,i}k$, then $|e_{v,i}| < t_{h,i}k$ has to hold.

Summarizing the conditions on the velocity errors (6.83), (6.84), (6.85) of Phase II, it can be concluded that

$$-(1 + \mu_2)t_{h,i}k < e_{v,i} < t_{h,i}k \quad (6.86)$$

must hold. In order to complete the proof, it has to be shown that the velocity error stays bounded in sliding, i.e., the error dynamics are so that the velocity error does not violate (6.86). The error dynamics are governed by

$$\begin{aligned} \dot{e}_{x,i} &= e_{v,i} , \\ \dot{e}_{v,i} &= -\frac{1}{t_h^*}e_{v,i} + \underbrace{\frac{t_{h,i}}{t_h^*}(u_{i-1} - \mu_2k)}_{\tilde{u}_i} . \end{aligned} \quad (6.87)$$

Similar to Phase 1, the dynamics of the velocity errors in (6.87) are affected by the inputs \tilde{u}_i , which are bounded by

$$\tilde{u}_i \in \left[-\frac{(1 + \mu_2)kt_{h,i}}{t_h^*}, \frac{kt_{h,i}}{t_h^*} \right] . \quad (6.88)$$

Since (6.87) is a linear time-invariant stable system with bounded inputs \tilde{u}_i and due to the switching condition for μ_2 in (6.69), it can be concluded that the velocity errors fulfill the bounds (6.86):

As long as $\mu_2 = 1$, the velocity errors are within the bounds by definition (6.69). If the velocity errors are so that $\mu_2 = 0$, the time-headway is held constant and the velocity errors converge asymptotically as in Phase 1 to $\lim_{t \rightarrow \infty} |e_{v,i}| = |t_{h,i}u_{i-1}|$, which is less than $t_{h,i}k$ as soon as $\tilde{\sigma}_i = 0$ is maintained, since $|u_{i-1}| < k$ for $t \rightarrow \infty$ due to the fact that the leader's acceleration is bounded by $u_{r,\max} < u_{\max} = k$ (see Assumption 1). This fact has been discussed, e.g., in [69] for a general consensus problem, which is summarized in Appendix A, and it also holds for the platooning application. Thus, the bounds in (6.86) hold and sliding can be maintained.

Finally, the time-headway approaches zero with (6.69), then (6.87) approximates (6.60) and a constant distance spacing (6.6) is achieved. If the time-headway cannot be decreased due to $e_{v,i} \geq 0$, then the preceding vehicle cannot be approached. There is no position error undershoot for all times in Phase II, which can be concluded from the position error dynamics in (6.87) and Assumption 1. Moreover, the errors are bounded and independent of the position i in platoon (6.87), the accelerations are bounded by design (6.67) with $u_{\max} = k$, hence all requirements in Section 6.1.1 are fulfilled. \square

6. String Stability in the Context of Sliding Mode Control

Remark 11. Note that large values for t_h^* in Theorem 1 mean slow convergence (6.60) in Phase II, while small values lead to loss of influence on sliding dynamics (6.71) in Phase I. Hence, a trade-off has been chosen with $t_h^* = 1$ in all simulations and experiments presented in this thesis.

Remark 12. In order to limit the relative velocities between the vehicles, the bounds for μ_1 in (6.65) and μ_2 (6.69) have been chosen conservatively. Note, however, that it has to be guaranteed that the error of the LTV system in (6.79) does not exceed the value $t_{h,i}k$ for constant $t_{h,i}$. An example is given in Appendix B.

Remark 13. The proposed approach yields bounded position errors, bounded velocity errors and bounded accelerations. However, the absolute velocities of the vehicles are not bounded; the velocities of the agents increase in order to speed up to reach the preceding vehicle. Hence, the absolute velocities along the string can increase along the string. However, the increase in velocities is expected, otherwise the vehicles will not be able to catch up with the platoon. Constraints on the absolute velocity may be handled by an upper-level decision unit, i.e., if too large velocities are necessary, the preceding vehicle is no longer tracked.

Remark 14. Note that so far, the leader's acceleration u_r is considered as the only disturbance to the platoon. Additional external disturbances will be discussed in the next chapter.

6.4.2. Simulation Results

In the following simulations, a platoon with 7 agents has been considered, i.e. 6 followers and node 0 as reference node (leader). The vectors of the agents' initial positions and velocities are given by

$$\begin{aligned} x(t_0) &= [190 \ 157 \ 120 \ 95 \ 75 \ 57 \ 0]^T \text{ in m,} \\ v(t_0) &= [15 \ 12 \ 14 \ 13 \ 16 \ 14 \ 15]^T \text{ in m/s.} \end{aligned} \quad (6.89)$$

The reference acceleration is time-varying and chosen as

$$u_r(t) = \begin{cases} -4 & t \in [12, 14) \\ 0.5 & t \in [14, 16) \\ 2 \sin(t) & \text{otherwise} \end{cases} \text{ in m/s}^2. \quad (6.90)$$

The desired constant distance in (6.12) is $\Delta_i = \Delta = 2$ m for all agents $i = 1, 2, \dots, 6$. The controller parameter (6.62), (6.67) has been chosen as $k = 5$ m/s² and the time-headway in (6.60) as $t_h^* = 1$ s.

The positions of the agents using the proposed approach from Theorem 1 are shown in Figure 6.34. The distances between the agents are reduced without collisions with the preceding vehicles, since position error undershoot is avoided as shown in Figure 6.35. The constant distance spacing is eventually reached in an approximated way, since the position errors and the time-headways approach zero. The phase plane is shown in Figure 6.36, where the bounds are depicted by dashed lines, which are not violated and hence sliding can be maintained.

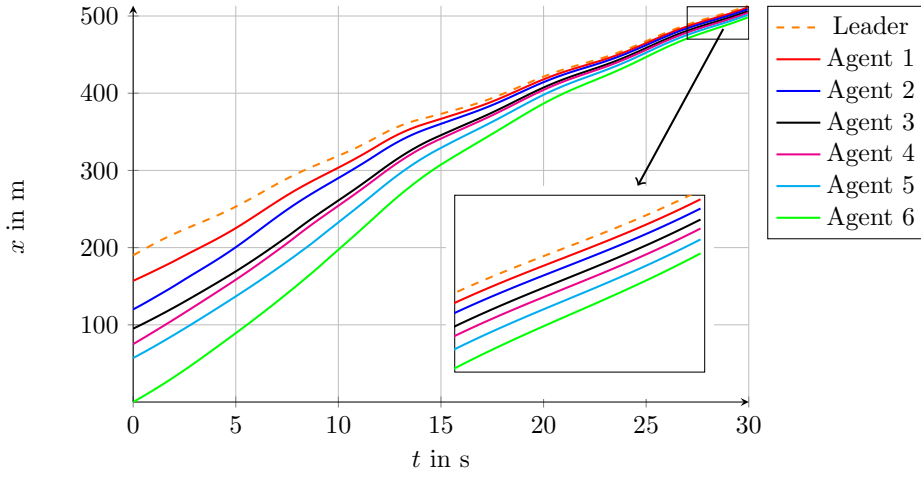


Figure 6.34.: Positions of all agents using the ATFC with a time-varying reference acceleration

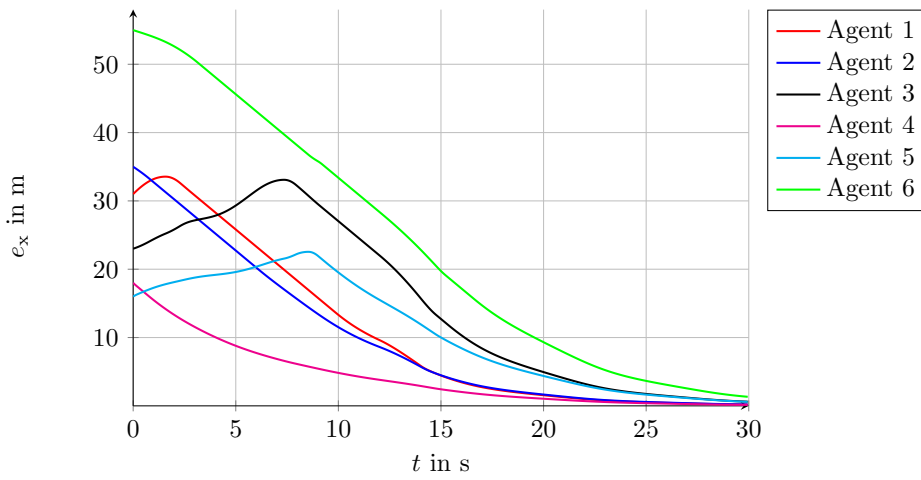


Figure 6.35.: The position errors of the agents using the ATFC are non-negative for all times and converge to zero, resulting in a constant distance spacing.

In Figure 6.37, the adapted time-headways are depicted. The maximum time-headway $t_{\max} = 5$ s has not been exceeded in the given example. The convergence of the time-headways depend on the velocity errors displayed in Figure 6.38; if the velocity errors are in the range given by (6.65), (6.69), the time-headways are decreased, otherwise held constant. Note also the different adaptation laws (6.64), (6.69): in the first phase, a constant slope that is scaled by the velocity is used in order to reach the defined time-headway t_h^* in finite time. Then in the second phase, the time-headway converges asymptotically to zero, and the convergence depends again on the velocity of the vehicles, which leads to different adaptation laws for the following agents.

Both the position errors and the velocity errors do not depend on the position i in the string. Figure 6.39 represents the filtered control inputs and it can be seen that the filtered accelerations are not amplified along the platoon. Note that if a predecessor applies a large deceleration (for example, starting from $t = 12$ s in (6.90)), the time-headway is held constant to avoid large velocity errors that would otherwise result in a position error

6. String Stability in the Context of Sliding Mode Control

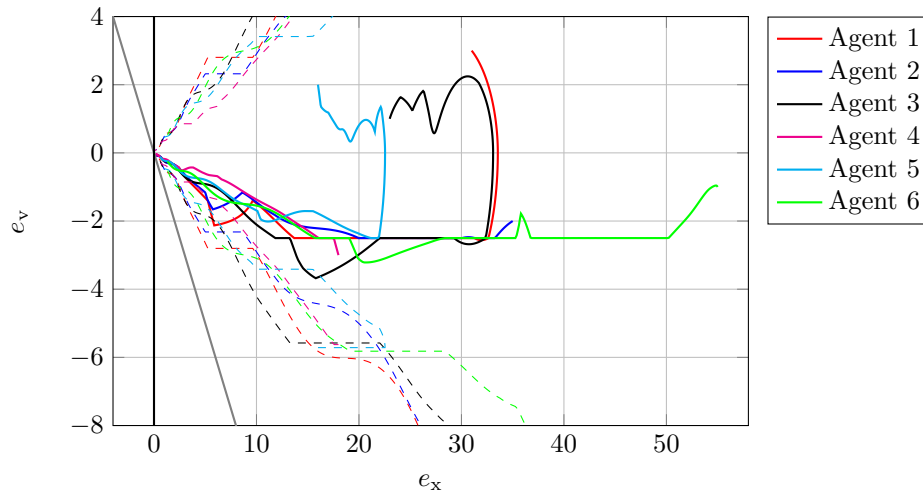


Figure 6.36.: Phase plane of the errors using the ATFC with bounds on the errors (dashed).

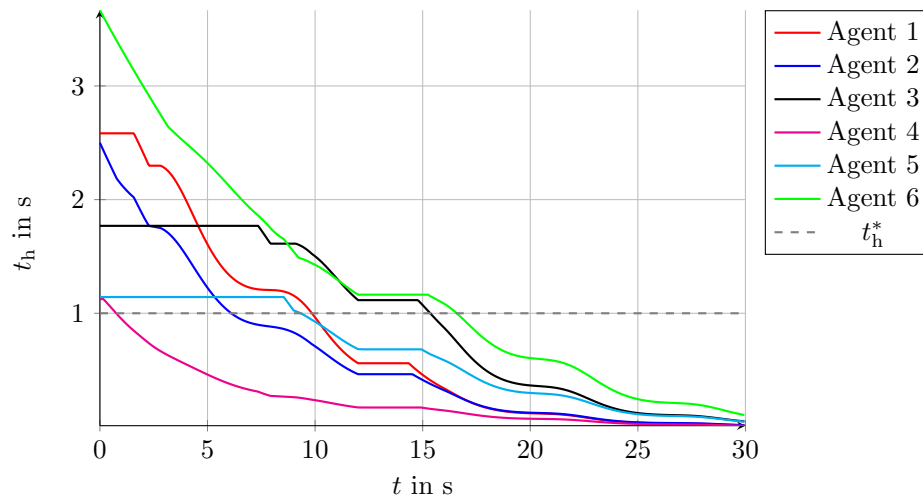


Figure 6.37.: Time-headways using the ATFC with different decays in the first phase and in the second phase

undershoot.

The simulations show that the proposed approach can still exhibit a damping behavior in severe situations due to the time-headway that is crucial for string stability. Eventually, a constant distance spacing is reached in an approximated way without communication between the vehicles. The leader's acceleration can be arbitrary within its bounds, and the sliding mode controllers are capable of compensating this disturbance.

Remark 15 (Integral Sliding Mode Control). *Note that the idea of the ATFC is similar to an integral sliding mode control approach: the sliding variable is chosen so that it is zero at initial time, $\sigma(t_0) = 0$. However, a "classical" integral sliding mode controller as discussed in [79],*

$$\begin{aligned} \sigma_i &= e_{x,i} + e_{v,i} + z_i, \\ \dot{z}_i &= -e_{v,i}, \quad z_i(t_0) = -e_{x,i}(t_0) - e_{v,i}(t_0), \end{aligned} \quad (6.91)$$

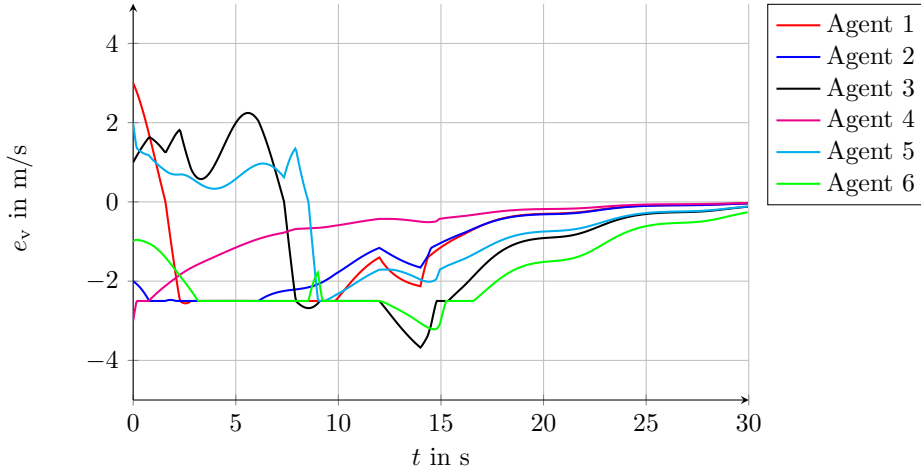


Figure 6.38.: The velocity errors of all agents using the ATFC are bounded and do not depend on the position in the string i . Large velocity errors are avoided by the proposed approach.

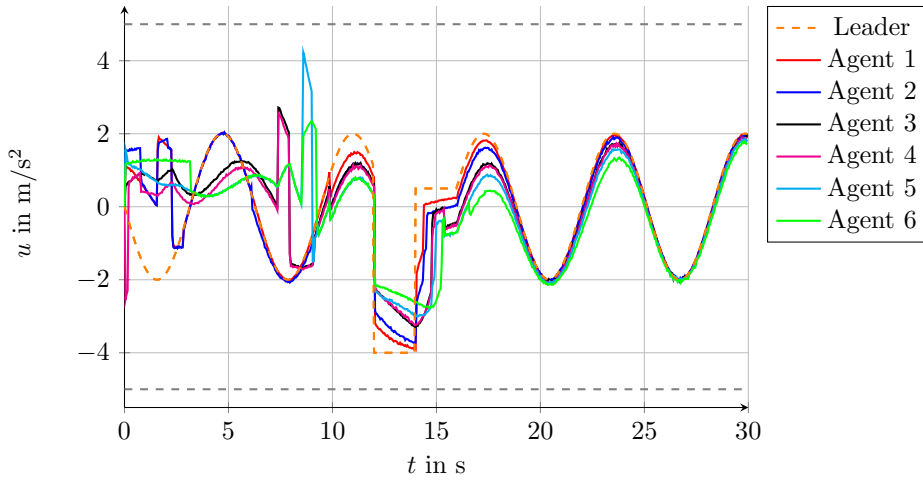


Figure 6.39.: The filtered control inputs using the ATFC are not amplified along the platoon.

cannot maintain sliding when a constant distance spacing is used. The sliding dynamics reads as

$$\dot{\sigma}_i = e_{v,i} + (u_{i-1} - u_i) - e_{v,i} , \quad (6.92)$$

and it is assumed that there is a nominal control input $u_{\text{nom},i}$ that stabilizes the system in the disturbance-free case, with the control input

$$u_i = u_{\text{nom},i} + u_{\text{smc},i} . \quad (6.93)$$

Then, the integral sliding mode control input $u_{\text{smc},i}$ compensates the disturbances, while the nominal controller $u_{\text{nom},i}$ stabilizes the system. However, in the given platooning application, all control inputs u_i have to be bounded by the same value, $|u_i| \leq u_{\text{max}}$. With a stabilizing nominal control input $u_{\text{nom},i} = F \cdot [e_{x,i} \ e_{v,i}]^T$, the control inputs are not bounded. Moreover, $u_{\text{smc},i}$ has to compensate $|u_{i-1}| \leq u_{\text{max}}$, and with $u_{\text{smc},i} = u_{\text{max}}$ the nominal part is inactive, $u_{\text{nom},i} = 0$ due to

6. String Stability in the Context of Sliding Mode Control

$|u_{\text{nom},i} + u_{\text{smc},i}| \leq u_{\text{max}}$. Then, with $u_{\text{nom}} = 0$ and $u_{\text{smc},i} = u_{i-1}$, the error dynamics is governed by

$$\begin{aligned} \dot{e}_{x,i} &= e_{v,i} , \\ \dot{e}_{v,i} &= 0 , \end{aligned} \tag{6.94}$$

and the errors do not converge to zero. Thus, this ‘‘classical’’ integral sliding mode controller cannot guarantee collision-free platooning.

Integral sliding mode control has been used for consensus problems in [127] for single-integrator systems $\dot{x}_i = u_i$, with the control input

$$\begin{aligned} u_i &= e_{x,i} - k \text{sign}(x_i + z_i) , \\ \dot{z}_i &= -e_{x,i} , \quad z_i(t_0) = -x_i(t_0) . \end{aligned} \tag{6.95}$$

Note, however, that a nominal part is used for the proposed consensus control law; this approach cannot be extended for double-integrator dynamics with control inputs bounded by the same maximum value.

Remark 16 (Sequential Activation). *Sequential activation of the controllers can be used to deal with bounded accelerations; as soon as the predecessor has reached the formation and applies $u_i = u_r$, a nominal control input can be computed so that $u_{\text{nom},i} \leq u_{\text{max}} - u_r$ is fulfilled. However, reaching a small inter-vehicle distance may take a long time, and the sequential activation is not focus of this work.*

In order to apply the proposed ATFC approach on highways, several scenarios have been analyzed that are the focus of the next chapter.

7. Formation Control in Highway Driving

The Motorway Chauffeur of Part 1 reaches its performance limits in merging scenarios on highways in case of high traffic volume as discussed in Chapter 3. Hence, this chapter focuses on a cooperative merging assist at lane reductions that extends the Motorway Chauffeur of Part 1 by the robust longitudinal formation controller from Part 2.

The merging assist is implemented as follows: instead of using the velocity profiles of the trajectory planning (TP) level from Section 2.2, the formation controller with adaptive time-headway (ATFC) proposed in Section 6.4 is used for longitudinal control. When the vehicles are in formation, i.e., keeping a safe distance to the preceding vehicle, the lane change is triggered and executed by the TP level. In order to guarantee safety, the evaluation of the TP level is still active, but the parameters of the ellipses are adapted according to the desired distance Δ_i that shall be maintained by the formation controller. Figure 7.1 depicts the closed loop system including the new formation control. While in Part I the reference

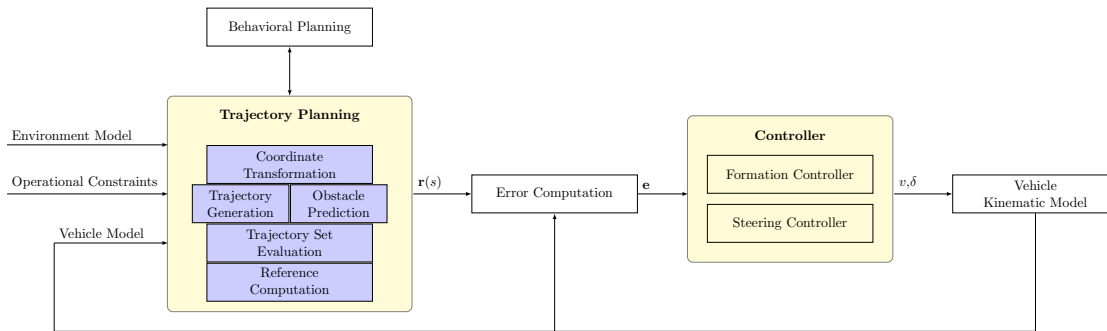


Figure 7.1.: Architecture of the merging assist on highways

path $\mathbf{r}(t)$ for the lane change was a function of time due to the predefined velocity profiles in the TP, the new reference path $\mathbf{r}(s)$ is a function of the way-length s . This way-length depends on the velocity of the vehicle that is computed by the formation controller. Then in the TP, a constant velocity is assumed over the entire prediction horizon. Since the velocity computed by the formation controller might not be constant, the reference lateral deviation has to be provided as function of way-length as well.

In the following discussions, the focus lies on longitudinal control in merging scenarios. The problem statement is similar to the platooning application. Once the longitudinal control is performed in a satisfactory way, the lateral guidance is activated; hence, the longitudinal and lateral dynamics are considered separately. The longitudinal dynamics of the controlled vehicle are simplified and assumed to be governed by double integrator dynamics. However, when attempting to apply the ATFC from Chapter 6 to highway driving, three additional problems arise: first, the network topology of the agents, i.e., the predecessor, needs to

7. Formation Control in Highway Driving

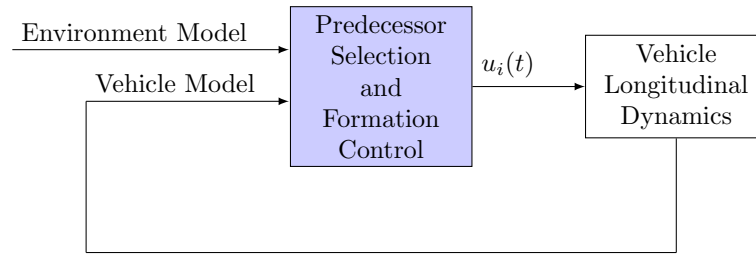


Figure 7.2.: Structure for longitudinal control of vehicle i .

be defined for all agents before the formation controller can be applied. In a platooning scenario, i.e. when considering only one lane, this procedure is very simple: the vehicle in front is automatically the predecessor. However, when two or more lanes are considered, such as is the case for merging scenarios at lane reductions, the choice of the vehicle that merges in front of the ego vehicle might not be clear. Hence, the merging sequence has to be defined before the longitudinal control is activated. This procedure is called “predecessor selection”, as depicted in Figure 7.2, which shows the structure of the longitudinal control loop. This first task of topology selection is discussed for the proposed formation controller in Section 7.1.

Second, if the assumptions for the ATFC are not fulfilled, which may apply to scenarios with two or more lanes, a strategy to reach the assumptions has to be considered separately. For example, suppose two vehicles are next to each other on adjacent lanes; then, depending on the predecessor selection, one vehicle has to yield to the other one. The initial position errors of the yielding agent are negative and the ATFC cannot be applied. Since the vehicle first has to “reach” the assumptions for applicability of the ATFC, this phase will subsequently be called “reaching phase” and is discussed in detail in Section 7.2.

Third, the curvature of the road and the lane change trajectory are disturbances to the longitudinal dynamics that have to be compensated by the velocity controller; this is described in Section 7.3. In Section 7.4, finally, the effects of actuator dynamics on the performance of platooning and the ATFC are investigated.

7.1. Topology Selection

The topology of the vehicles, i.e., the choice of a preceding or involved vehicle in a maneuver, is very important for merging scenarios. In a platooning scenario on a single lane, the preceding vehicle is the vehicle in front and the topology is clearly defined as in Chapters 5 and 6. However, when vehicles on multiple lanes are considered, the choice of the preceding vehicle for a merging maneuver, also called merging sequence, has to be investigated in more detail.

With a reasonable predecessor choice, the position errors with respect to vehicles on the same lane have no undershoot and thus collisions can be avoided, and safety can be achieved. Moreover, by avoiding low velocities that propagate upstream, traffic jams can be reduced and thus efficiency is increased. Finally, comfortable merging scenarios will be achieved, if high accelerations of the vehicles and unnecessary braking maneuvers are avoided. With these goals in mind, an appropriate choice of the preceding vehicle is proposed based on relative velocities and relative positions of the vehicles that are involved in the merging scenario.

As soon as multiple lanes are considered, the topology is not predefined. This is highlighted in the example in Figure 7.3 with two lanes and two possible positions A, B of the green vehicle for a merging maneuver to the right lane. The basic considerations for selecting the preceding vehicle are as follows. First of all, the relative positions are considered; if another vehicle is in front, then it is a potential predecessor, e.g., the red vehicle has two potential predecessors. However, based on relative positions only, it is not clear whether the green vehicle will pass the blue one in order to reach position B , or slow down for position A . For this reason, the relative velocities are considered, with the aim to avoid decelerations or stand-still before the merging point.

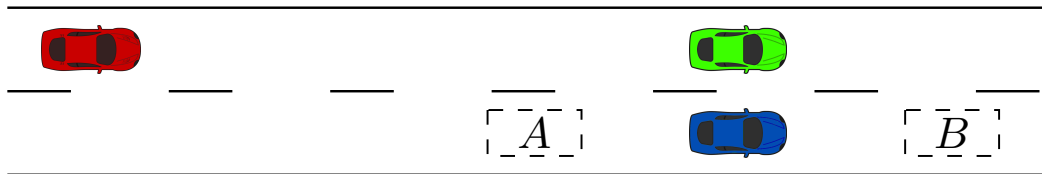


Figure 7.3.: Merging scenario with two possible positions A, B for the green vehicle. In this scenario, the choice of the predecessor for all three vehicles is unclear, and a proper topology selection algorithm is necessary.

Consider again Figure 7.3, where the relative position between green and blue vehicle is zero and thus not sufficient to determine the preceding vehicle; the following three considerations based on the relative velocity are taken into account in the predecessor selection: First, if the green vehicle is faster than the blue vehicle, merging behind the blue vehicle is not reasonable due to undesired decelerations. The green vehicle is thus the preceding vehicle and merges at position B . Second, if the blue vehicle is faster, then the blue vehicle is the predecessor of the green vehicle, which merges at position A . Third, if the green vehicle maintains the same speed as the blue vehicle, then it is not clear if the green or the blue vehicle will be the preceding vehicle. In this case, the preceding vehicle cannot be determined from relative position and velocity alone, and the merging sequence

7. Formation Control in Highway Driving

is determined by the current lane of the vehicles. For example, if the left lane has priority in Figure 7.3, then for zero relative velocity the green vehicle on the left is the predecessor of the blue vehicle. The red vehicle then chooses either the green vehicle (green vehicle in position A) or the blue vehicle (green vehicle in position B) as predecessor. Summarizing these considerations, the relative positions, relative velocities and lanes of the vehicles have to be considered for predecessor selection.

It is assumed in the remainder of this chapter that all considered vehicles apply the same selection algorithm. If no position error undershoots occur, it is sufficient to apply the algorithm once. Otherwise it has to be applied at every sampling step in order to avoid collisions with vehicles on the own lane. Consider the example in Figure 7.4, where the red vehicle follows the green one, and the green agent follows the blue one. In the case of a position error undershoot of the green vehicle, where the maximum undershoot is indicated by the green ellipse, the red vehicle tries to follow the green vehicle and collides with the blue one (green rectangle). However, if the predecessor of the red agent is changed to the blue agent, collisions can be avoided as indicated by the blue rectangle.

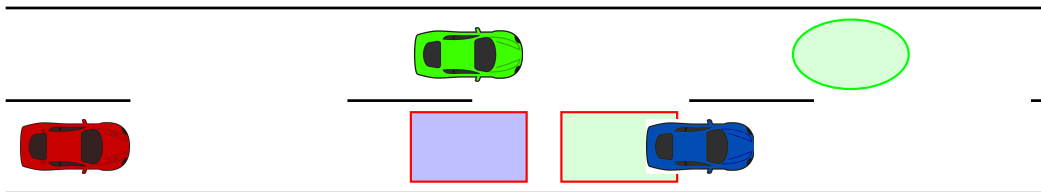


Figure 7.4.: Example for a changing predecessor with an active predecessor selection algorithm

Hence, at a given time instant, e.g., when a merging flag is set by the behavioral planning level, the merging procedure is initiated. Then, the predecessor selection is executed and the longitudinal formation controller is used. An algorithm for the predecessor selection has been proposed in [146], which is presented in detail subsequently. First, it has to be determined which vehicles in the surroundings are taken into account by the ego vehicle. It is reasonable to consider only the closest vehicles as depicted in Figure 7.5 for the green vehicle. For this purpose, two parameters are defined: the maximum distance rear Δ_{\min} and the maximum distance in front of the ego vehicle Δ_{\max} , considering all lanes. These distances can depend on the velocity and are typically limited by the range of the sensors. The gray vehicles are out of range, while the colored vehicles are added to the list of possible preceding vehicles.

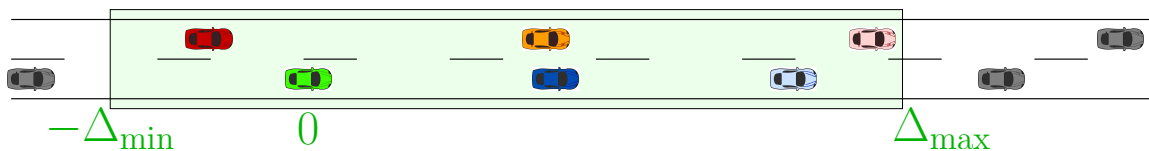


Figure 7.5.: First step of predecessor selection: vehicles that are not in a reasonable range of the green vehicle are excluded, where the range is defined by Δ_{\min} and Δ_{\max} and depicted by the green rectangle.

As in the previous chapters, the agents are modeled by double integrator dynamics,

$$\begin{aligned}\dot{x}_i &= v_i, \\ \dot{v}_i &= u_i,\end{aligned}\tag{7.1}$$

where x_i and v_i denote position and velocity of vehicle i , respectively, for $i = 0, \dots, N$. Then, in the second step, the relative positions $\tilde{e}_{x,ij} = x_j - x_i$ and relative velocities $e_{v,ij} = v_j - v_i$ of all remaining vehicles are considered; if the relative states fulfill

$$\sigma_{p,ij} = \tilde{e}_{x,ij} + e_{v,ij} \geq 0,\tag{7.2}$$

the corresponding vehicle j is a potential predecessor for i .

This auxiliary variable $\sigma_{p,ij}$ corresponds to the constant distance spacing (6.22) with $c = 1$ in the previous chapter. It is necessary only if multiple lanes are considered, since vehicles can be next to each other with $\tilde{e}_{ij} = \tilde{e}_{ji} = 0$ as discussed for the example in Figure 7.3. In Figure 7.5, potential predecessors of the green vehicle are the dark blue vehicle on the own lane and the dark red and orange vehicles on the adjacent lanes. Note that the red vehicle behind the green ego vehicle is faster, i.e., the relative velocity is positive; thus (7.2) is fulfilled, and it is a potential preceding vehicle. Otherwise, the vehicle would be excluded from the list.

Note that if $\sigma_{p,ij} = 0$, with $\tilde{e}_{x,ij} = e_{v,ij} = 0$, then the preceding vehicle is chosen based on the lane, i.e., the faster vehicles are typically on the left lane, and thus the vehicle on the right lane has to yield. Moreover, if $\sigma_{p,ij} = 0$ and $e_{v,ij} \geq 0$, then the faster vehicle j is the predecessor, since larger velocities are desired in order to increase traffic throughput.

Third, if more than one vehicle remains in the list of predecessors as in the example shown in Figure 7.5, the lateral deviation of the vehicles is taken into account: on the own lane, only one vehicle o can be a predecessor. On the adjacent lane, only the vehicle a with smallest relative position $e_{x,ij}$ that has not been removed yet, i.e., with $\sigma_{p,ij} \geq 0$, needs to be considered. Note that all cars that remain in the list of possible predecessors after the second stage will eventually be in front of the ego; hence it is sufficient to consider only the rearmost vehicle on the adjacent lane with $\sigma_{p,ij} \geq 0$, since all vehicles apply the same selection algorithm.

Fourth, the regions and relations between the two remaining vehicles o, a are taken into account. It is checked which vehicle is the predecessor or the follower with respect to vehicles o, a . For example in Figure 7.5, the orange car in front of the green vehicle will merge in front of the dark blue vehicle. Then, the dark blue in front and the dark red vehicle behind the green ego vehicle remain potential preceding vehicles. The dark red car will follow the dark blue one and is hence the predecessor for the ego vehicle.

Finally, one preceding vehicle is found and a certain strategy to reach the desired distance can be applied, which will be discussed in the following section. This selection algorithm is summarized in pseudo code in Algorithm 1.

Remark 17. *If there is no preceding vehicle for a vehicle, then it acts as a leader and should accelerate to the maximum possible velocity (defined by road limits or the driver) in order to increase traffic throughput. This longitudinal control task can then be handled, e.g., by the TP.*

Algorithm 1 Predecessor selection

```

1: Step 1: Limit Range
2: for  $j = 1, \dots, \text{length}(\text{detected vehicles of agent } i)$  do
3:   compute  $\tilde{e}_{x,ij} = x_j - x_i$  and  $e_{v,ij} = v_j - v_i$ 
4:   if  $-\Delta_{\min} \leq \tilde{e}_{x,ij} \leq \Delta_{\max}$  then
5:     add vehicle  $j$  to vehiclelist
6: Step 2: Check regions:
7: for  $j = 1, \dots, \text{length}(\text{vehiclelist})$  do
8:   compute  $\sigma_{p,ij} = \tilde{e}_{x,ij} + e_{v,ij}$ 
9:   if  $\sigma_{p,ij} == 0$  and  $e_{v,ij} == 0$  and vehicle  $j$  left of  $i$  then
10:    add vehicle  $j$  to predecessorlist ▷ lane priority
11:   else if  $\sigma_{p,ij} \geq 0$  and not ( $\sigma_{p,ij} = 0$  and  $e_{v,ij} < 0$ ) then
12:    add vehicle  $j$  to predecessorlist ▷ velocity priority
13: sort predecessorlist w.r.t.  $\tilde{e}_{x,ij}$ 
14: Step 3: Lane Distinction:
15: for  $j = 1, \dots, \text{length}(\text{predecessorlist})$  do
16:   if vehicle  $i$  and vehicle  $j$  on same lane then
17:     add vehicle  $j$  to ownlanelist
18:   else
19:     add vehicle  $j$  to adjacentlanelist
20: predecessor own lane  $o \leftarrow$  first entry of ownlanelist
21: predecessor adjacent lane  $a \leftarrow$  first entry of adjacentlanelist
22: Step 4: Predecessor Check:
23: if no predecessors  $o, a$  found then
24:   ego vehicle is leader
25: else if no predecessor on one lane then
26:   predecessor  $\leftarrow$  predecessor on other lane
27: else
28:   check regions as in Step 2,  $\sigma_{p,oa} = \tilde{e}_{x,oa} + e_{v,oa}$ 
29:   if  $o$  is immediate predecessor of  $a$  then
30:     predecessor  $\leftarrow a$ 
31:   else
32:     predecessor  $\leftarrow o$ 

```

7.2. Reaching Phase

In the following investigations, the focus lies on the situations where a preceding vehicle has been selected, but the ATFC from Chapter 6 cannot yet be switched on. According to Assumption 2 of the ATFC, the initial position errors have to fulfill

$$0 \leq e_{x,i}(t_0) \leq t_{\max,i} v_i, \quad (7.3)$$

and the velocity errors must be within the bounds

$$|e_{v,i}(t_0)| \leq \min\left(e_{x,i}(t_0)/t_{h,i}(t_0), t_{h,i}(t_0)u_{\max}\right), \quad (7.4)$$

with

$$t_{h,i}(t_0) = \begin{cases} e_{x,i}(t_0)/v_i(t_0) & \text{if } |e_{x,i}(t_0)| > t_h^* v_i(t_0) \\ (e_{x,i}(t_0) + t_h^* e_{v,i}(t_0))/v_{i-1}(t_0) & \text{otherwise.} \end{cases} \quad (7.5)$$

Independent of vehicles on adjacent lanes, vehicles on the same lane have to maintain safety distances for all times also for the merging scenarios, which is stated by the following assumption.

Assumption 3. *All vehicles fulfill Assumption 2 with respect to preceding vehicles on the same lane, but necessarily for predecessors on different lanes.*

In merging scenarios with multiple lanes, however, Assumption 2 is too restrictive. For example, if the initial position error is zero, the initial time-headway $t_{t,i}(t_0)$ is also zero. Then, the velocity error has to be zero as well for the ATFC to be applicable. However, if the following vehicle is on the adjacent lane, non-zero velocity errors are common and have to be taken into account for longitudinal control in merging scenarios. Moreover, from a mathematical point of view, a negative position error may result in a negative initial time-headway, which is not reasonable. Hence, in order to circumvent these restrictions, the ATFC is activated at the time instant t_0 at which the bounds (7.4) are satisfied for the first time. A so-called reaching strategy is implemented in order to reach this state; two different approaches are analyzed for this purpose: first, the constant distance spacing from Chapter 6 is used, and second, the ATFC is extended in order to be applicable for negative position errors.

7.2.1. Reaching Phase with Constant Distance Spacing

In the following, the vehicles are assumed to be sorted such that vehicle $i - 1$ is the predecessor of vehicle i . Consider the position error $e_{x,i}$ and the velocity error $e_{v,i}$ defined by

$$\begin{aligned} e_{x,i} &= x_{i-1} - x_i - \Delta_i, \\ e_{v,i} &= v_{i-1} - v_i, \end{aligned} \quad (7.6)$$

7. Formation Control in Highway Driving

with (7.1), where Δ_i is the desired constant distance for the merging scenario. In a first step, the time-headway is bounded so that

$$t_{h,i}(t_0) \geq 0 \quad (7.7)$$

holds. Thus, negative position errors may result in $t_{h,i}(t_0) = 0$. For control purposes, the constant distance sliding surface is used, which has been defined previously as

$$\sigma_{d,i} = e_{x,i} + ce_{v,i} . \quad (7.8)$$

Keeping in mind the topology selection algorithm in Section 7.1, the sliding variable is a shifted version of the auxiliary variable $\sigma_{p,i} = \sigma_{p,i(i-1)}$ in (7.2) for $c = 1$, i.e.,

$$\sigma_{d,i} = \sigma_{p,i} + \Delta_i . \quad (7.9)$$

Since all predecessors fulfill $\sigma_{p,i} \geq 0$ according to Algorithm 1, the initial values for $\sigma_{d,i}$ are bounded from below by Δ_i . The phase plane of the errors can then be divided into five different regions as shown in Figure 7.6, where the bounds of the ATFC are plotted for $t_{h,i} = 1$ s. Note that initial errors in the yellow (3), orange (4) and brown region (5) correspond to predecessors on the adjacent lane due to Assumption 3.

Then, if the errors do not fulfill the bounds (7.3), (7.4), and the ATFC is not applicable ("ATFC off"), different strategies are required: First, errors in the gray region are excluded from consideration due to $\sigma_{p,i} < 0$, i.e., because the corresponding vehicle cannot be a predecessor. Second, in the white region, the ATFC can be applied.

Otherwise, if the initial errors are in the yellow (3), orange (4) or brown region (5), a first order sliding mode controller with the constant distance spacing (7.9),

$$u_i = k\text{sign}(\sigma_{d,i}) , \quad (7.10)$$

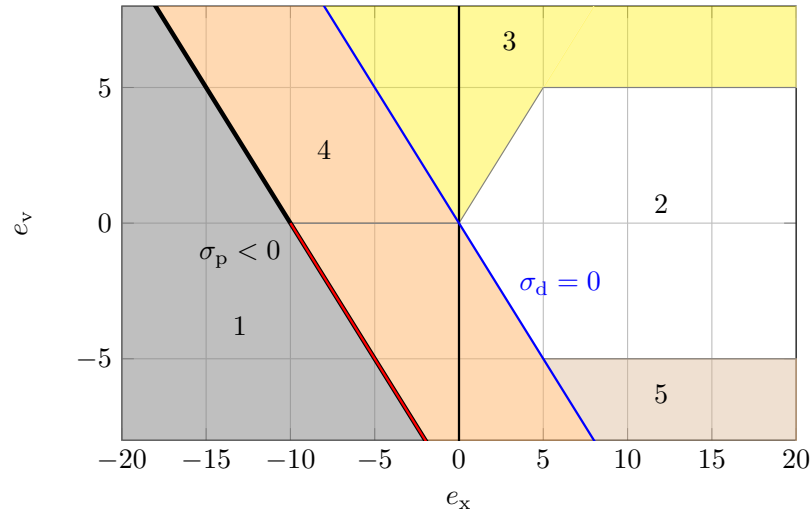


Figure 7.6.: Phase plane of the errors for $t_{h,i} = 1$, $\Delta_i = 10$. The vehicles are excluded from the predecessor list if the corresponding errors are in the gray region.

is used in a first step. However, as discussed in Chapter 6, sliding along the surface $\sigma_{d,i} = 0$ cannot be guaranteed, and either the white region is entered where the ATFC is applied, or the vehicle enters the gray region, in which case it is discarded as predecessor and another vehicle is chosen.

Control input (7.10) can result in different trajectories as shown in Figure 7.7: First, if the following vehicle and the predecessor decelerate with the same value $u_{i-1} = u_i$, the velocity error is constant with $\dot{e}_{v,i} = 0$, and trajectory *I* eventually enters the gray region, where the predecessor is either discarded, or an emergency maneuver is triggered by a higher level planning level as described in Part I of this thesis. Note that vehicles are on adjacent lanes according to Assumption 3, and discarding the vehicle does not result in collisions. In the yellow region, the preceding vehicle may increase the distance until it is no longer in the range of the follower if both vehicles accelerate at maximum, again due to $\dot{e}_{v,i} = 0$, as shown by trajectory *II*. Then, the predecessor will eventually leave the considered range and another preceding vehicle is determined, or the ego vehicle becomes a leader. For small negative position errors and small velocity errors, reaching and sliding can be possible in the left half-plane of Figure 7.7, as depicted by trajectory *IV*. In this case, switching to the ATFC is not required to reach and maintain the desired distance. However, reaching cannot be guaranteed for all errors, and position overshoot occurs as depicted by trajectory *III* and eventually, the ATFC can be applied. Finally, initial errors in the brown region can either enter the white region, or the orange region as depicted by trajectory *V*. Then, the above mentioned cases *I*, *III*, or *IV* apply.

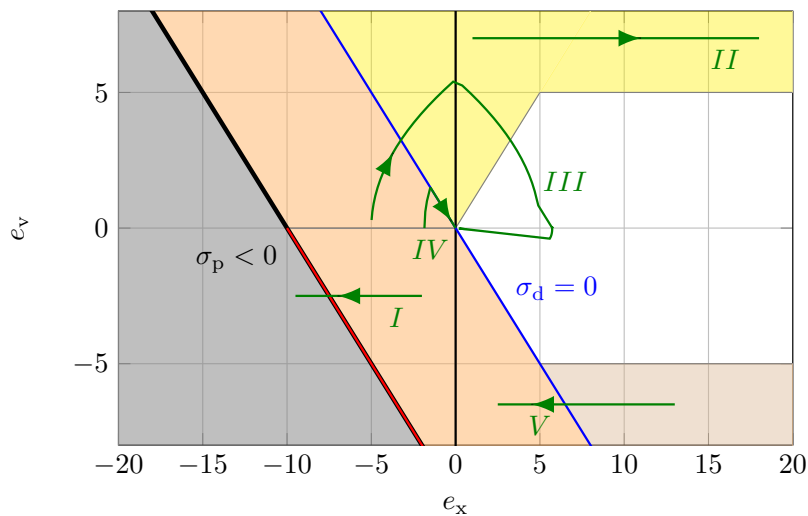


Figure 7.7.: Phase Plane with different trajectories using the constant distance sliding surface if the ATFC is not applicable

Example 1

The following example shows simulation results for three agents with initial positions as depicted in Figure 7.8. The green vehicle (ego vehicle, e) passes the blue one (Agent 2) and merges at gap A behind the red vehicle (Agent 1), which follows the gray one (Agent 0).

7. Formation Control in Highway Driving



Figure 7.8.: Merging example with one leader (gray) and three following vehicles (colored). The green ego vehicle has to reach position A.

The initial errors with respect to the ego vehicle e are given by

$$\begin{aligned} e_{x,0e} &= 60 \text{ m} , & e_{x,1e} &= 19 \text{ m} , & e_{x,2e} &= -10.8 \text{ m} , \\ e_{v,0e} &= -4 \text{ m/s} , & e_{v,1e} &= -7 \text{ m/s} , & e_{v,2e} &= -8 \text{ m/s} . \end{aligned} \quad (7.11)$$

The initial position error with respect to the blue vehicle $e_{x,2e}$ is negative, since the desired distance $\Delta_i = \Delta = 10 \text{ m}$ is initially not maintained between the vehicles. The green vehicle is much faster than the blue one and the variables $\sigma_{d,2e}$ and $\sigma_{p,2e}$ with respect to the blue vehicle are negative and thus in the gray region. The red vehicle speeds up in order to reach the desired position behind the gray leader. It applies the ATFC to reach the gray vehicle, and is the predecessor for the green vehicle, while the blue vehicle follows the green one.

The initial errors of the ego vehicle with respect to the red vehicle are in the brown region (5) in Figure 7.6, with $\sigma_{p,1e} > 0$ and $\sigma_{d,1e} > 0$, and thus the ATFC cannot be applied: the green vehicle's initial velocity is very large, $v_e(t_0) = 38 \text{ m/s}$, thus a small initial time-headway is computed in (7.5) and the velocity error does not fulfill the bounds (7.4) for the ATFC. The green ego vehicle will decrease its distance to the red vehicle according to (7.10). The errors of the blue vehicle with respect to the green vehicle are in the yellow region and the blue vehicle accelerates in order to reach the desired distance behind the green vehicle. However, all disadvantages of the reaching phase with constant distance spacing can be observed: the position error exhibits an undershoot and the ego vehicle first passes the red vehicle due to a large velocity error, as shown in Figure 7.10. Since the vehicles are on adjacent lanes, no collisions occur, but the preceding vehicle for the blue agent changes, and the velocity error and the sliding variable of the blue vehicle change abruptly as shown in Figure 7.9. As soon as the ego vehicle is behind the red vehicle, the predecessor for the blue vehicle is changed back to the green vehicle. Note that the blue vehicle does not collide with the red one, since the preceding vehicle will change as soon as the green vehicle is the predecessor of the red one, as discussed previously for Figure 7.4.

The red vehicle applies the ATFC from the start, while the green and blue vehicle reach the white region at a later time instant. Eventually, the ATFC is applied by all following vehicles, and the time-headways are as depicted in Figure 7.11. Note that the time-headway is zero if the ATFC is not active. However, the performance of the green vehicle is not satisfactory. Although no collisions occur due to different lanes of the green and red vehicle, the large accelerations of the green and blue vehicles and the resulting velocities have a bad impact on the traffic flow.

Thus, the strategy should be chosen depending on the "urgency" of the merging maneuver, as simulated in the next example.

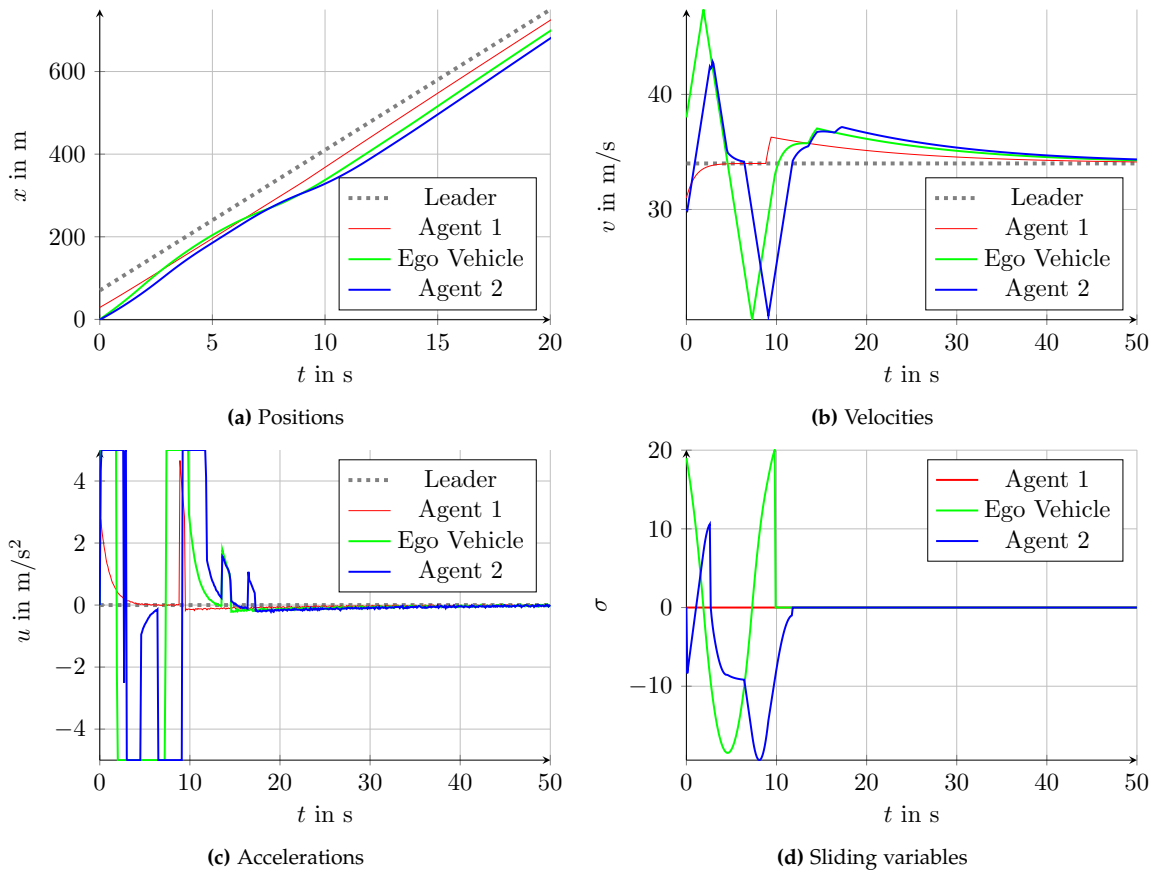


Figure 7.9.: Results of the reaching phase with the constant distance spacing in Example 1

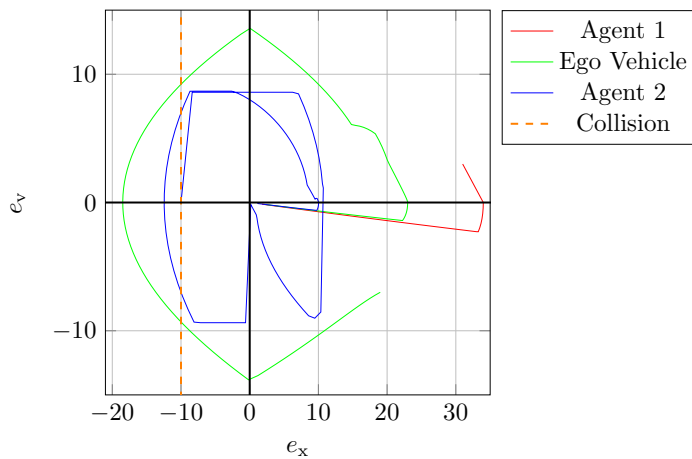


Figure 7.10.: Phase plane of the errors in Example 1, with position undershoot of the ego vehicle and Agent 2 due to a constant distance spacing. The errors of Agent 2 are shown with respect to the ego vehicle; no collisions occur since the ego vehicle is on the adjacent lane. Agent 1 applies the ATFC.

7. Formation Control in Highway Driving

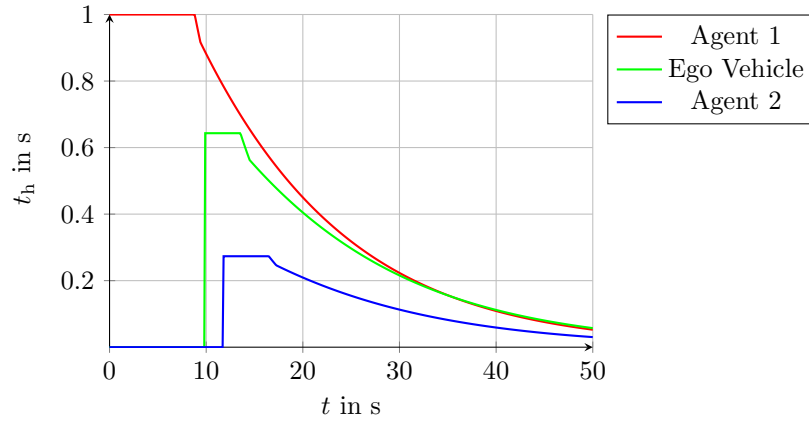


Figure 7.11.: Time-headways in Example 1 using a constant distance spacing for reaching the region where the ATFC is applicable.

Example 2

In the second example, the constant distance sliding variable of the blue vehicle with respect to its green predecessor is negative, i.e., in the orange instead of the yellow region of Figure 7.6. The initial errors are given by

$$\begin{aligned} e_{x,0e} = 60 \text{ m} , \quad e_{x,1e} = 19 \text{ m} , \quad e_{x,2e} = 8 \text{ m} , \\ e_{v,0e} = -2 \text{ m/s} , \quad e_{v,1e} = -5 \text{ m/s} , \quad e_{v,2e} = -6 \text{ m/s} . \end{aligned} \quad (7.12)$$

In order to avoid low velocities, the following vehicle waits for the predecessor on the adjacent lane to speed up without braking. Only if the predecessor does not increase the distance in a predefined time, the following vehicle will yield by decelerating, and the control input is defined by

$$u_i = \begin{cases} k \text{sign}(\sigma_{d,i}) & \text{yield} \\ 0 & \text{wait} . \end{cases} \quad (7.13)$$

Thus, a “yielding point” can be defined, at which the following vehicle has to yield to the preceding car on the adjacent lane.

Simulation results are shown in Figure 7.12. Since the blue agent waits for the green ego vehicle to speed up, its performance is much better than in the first example. Note that the performance of the green vehicle has also changed, since it waits for the red vehicle to speed up. However, the reaching phase takes longer than in the first example. In summary, the performance of the blue vehicle is much better, but the green vehicle’s behavior needs improvement, which can also be concluded from the phase plane in Figure 7.13. In Figure 7.14, the time-headways are shown. Note that although the time-headway of the blue vehicle is larger than in the previous example, the overall performance is better.

Remark 18. Note that if the preceding vehicle brakes during the “waiting” phase of the following vehicle, the velocity error will decrease and eventually become zero. Then, hard deceleration may also be necessary for the following vehicle. However, as soon as the gray region is entered, the predecessor should be discarded.

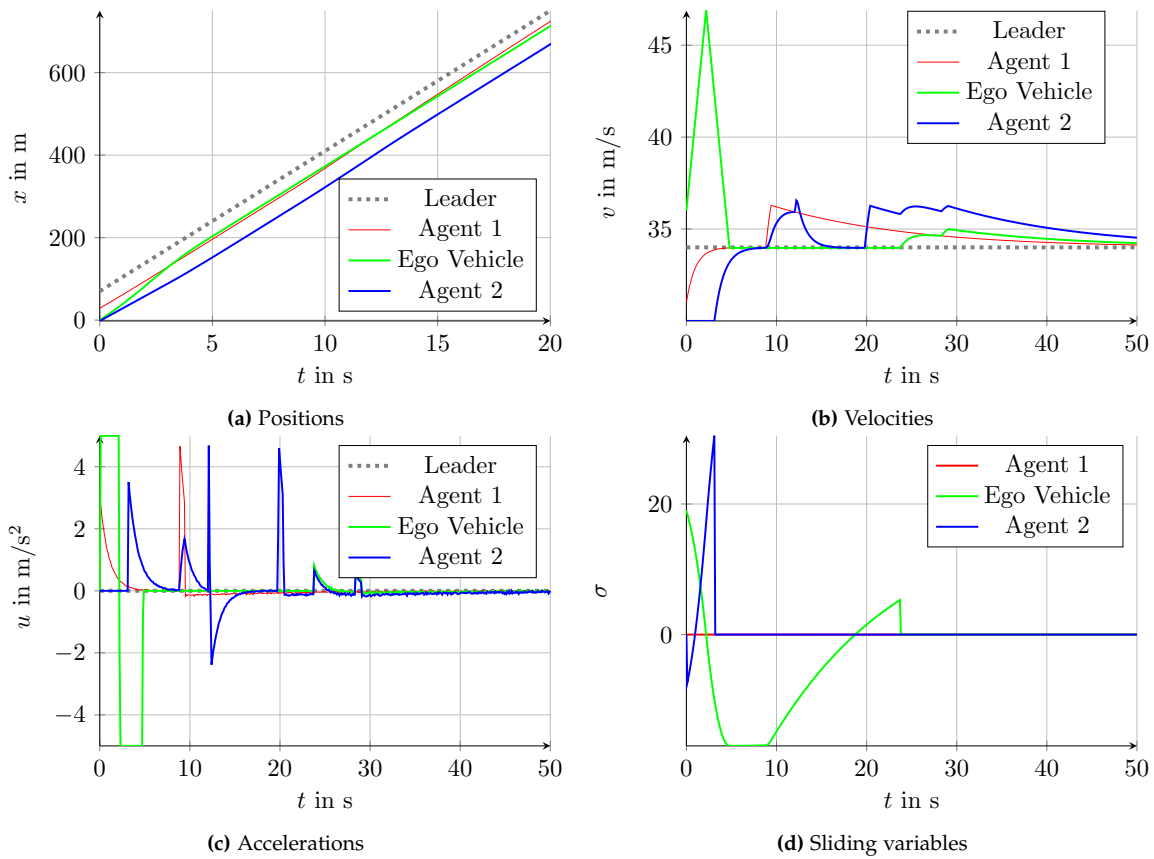


Figure 7.12.: Results of the constant distance spacing with the possibility to wait for the predecessor to speed up in Example 2

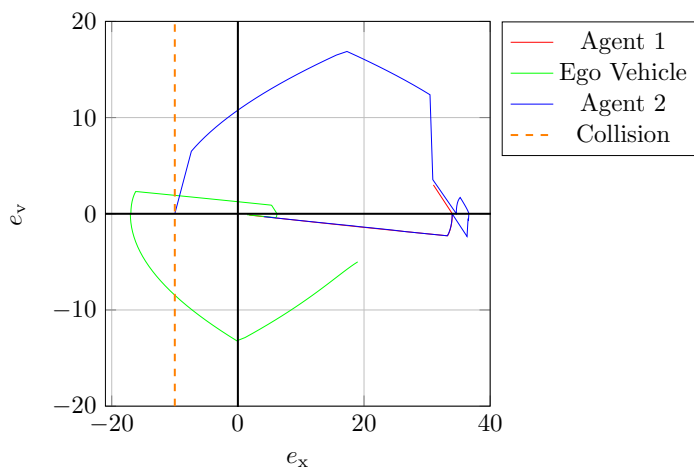


Figure 7.13.: Phase plane of the errors in Example 2

7. Formation Control in Highway Driving

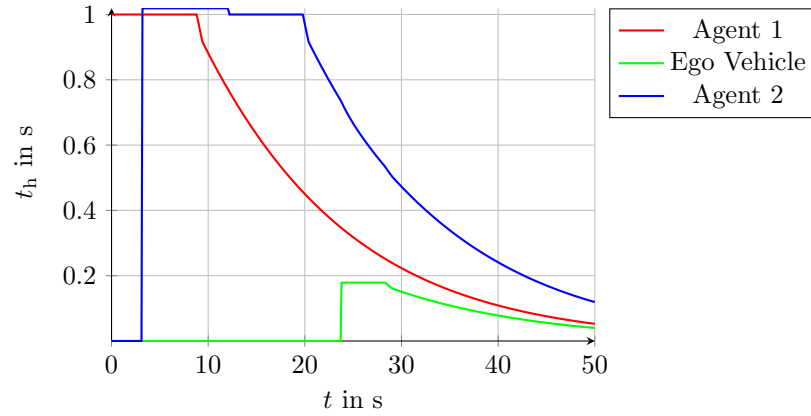


Figure 7.14.: Time-headways in Example 2

Example 3

The third example shows that the constant distance spacing can result in acceptable behavior for small initial errors, since reaching the sliding surface and staying on the surface is possible without switching to the ATFC as previously discussed for trajectory *IV* in Figure 7.7.

The focus of this example lies on the blue vehicle, and the green vehicles' behavior has been changed so that the ATFC is applicable. The initial error are thus given by

$$\begin{aligned} e_{x,0e} &= 60 \text{ m} , & e_{x,1e} &= 39 \text{ m} , & e_{x,2e} &= -10.8 \text{ m} , \\ e_{v,0e} &= -2 \text{ m/s} , & e_{v,1e} &= -5 \text{ m/s} , & e_{v,2e} &= -1 \text{ m/s} . \end{aligned} \quad (7.14)$$

Simulation results are shown in Figure 7.15, with the time-headways in Figure 7.17. Note that the blue vehicle never reaches the white region of the ATFC, since sliding along the constant distance surface is maintained as shown in the phase plane of the errors in Figure 7.16.

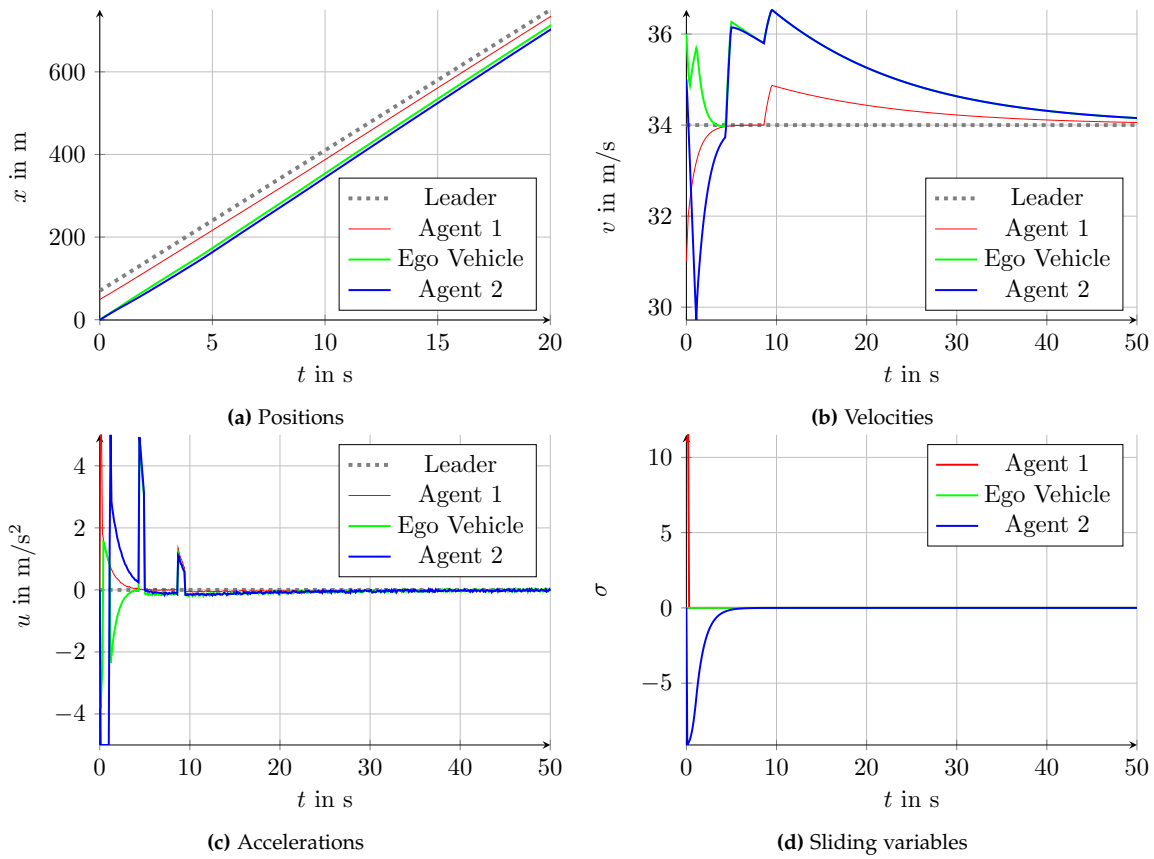


Figure 7.15.: Results of the proposed cooperative merging assist in Example 3

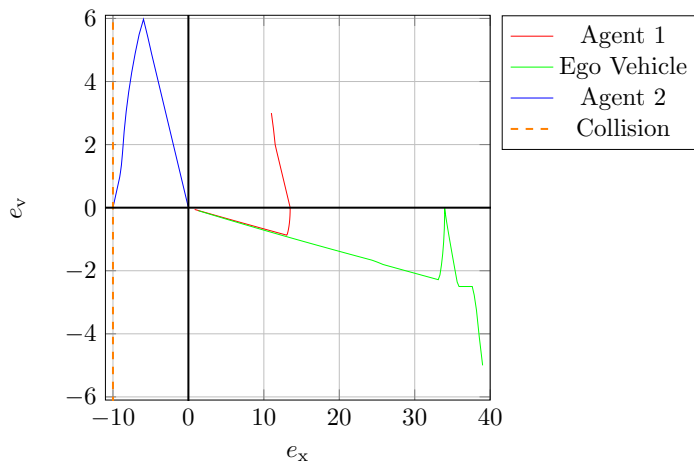


Figure 7.16.: Phase plane of the errors in Example 3

7. Formation Control in Highway Driving

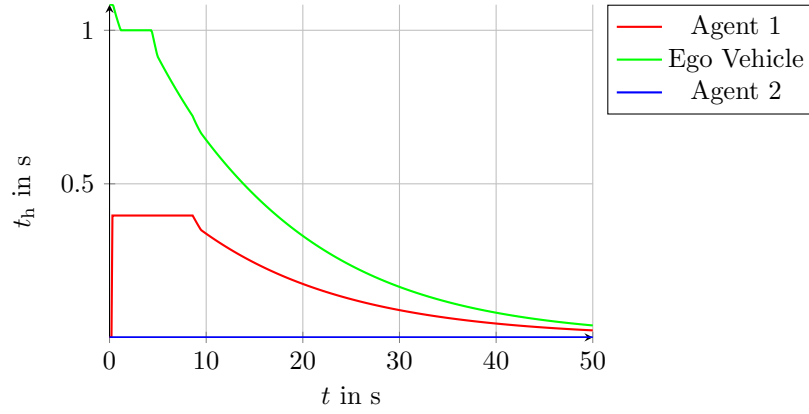


Figure 7.17.: Time-headways in Example 3

Example 4 - SUMO

Simulation results using SUMO with control input (7.10) in the reaching phase are shown in Figure 7.19. The relative distances with respect to the ego vehicle, the lateral deviations and the velocities of the four vehicles in the given example are illustrated therein, and the initial errors are given by

$$\begin{aligned}
 e_{x,0e} &= 110 \text{ m} , & e_{x,1e} &= 35 \text{ m} , & e_{x,2e} &= 8 \text{ m} , \\
 e_{v,0e} &= -2 \text{ m/s} , & e_{v,1e} &= -5 \text{ m/s} , & e_{v,2e} &= -6 \text{ m/s} .
 \end{aligned} \tag{7.15}$$

The sampling time of SUMO is set to $T_{\text{SUMO}} = 0.1 \text{ s}$, while the controller in MATLAB is executed with sampling time $T_{\text{ctrl}} = 0.01 \text{ s}$. This results in discretization effects, which can be observed in the sliding variable, which is shown in Figure 7.18 for the blue vehicle: the sliding variable is not zero, but exhibits a periodic motion within a certain band close to zero, which is also called boundary layer, see, e.g., [35]. Note that the implementation of the ATFC needs to be adapted; in the simulation at hand this is done by decreasing the time-headway as soon as the sliding variable is within its boundary layer. However, these effects are not considered further in this work.

Note that the overall performance of the cooperative merging scenario shown in Figure 7.19 is satisfactory. The relative distances are decreased and the velocities are increased, and the traffic flow can thus be improved during the merging scenario.

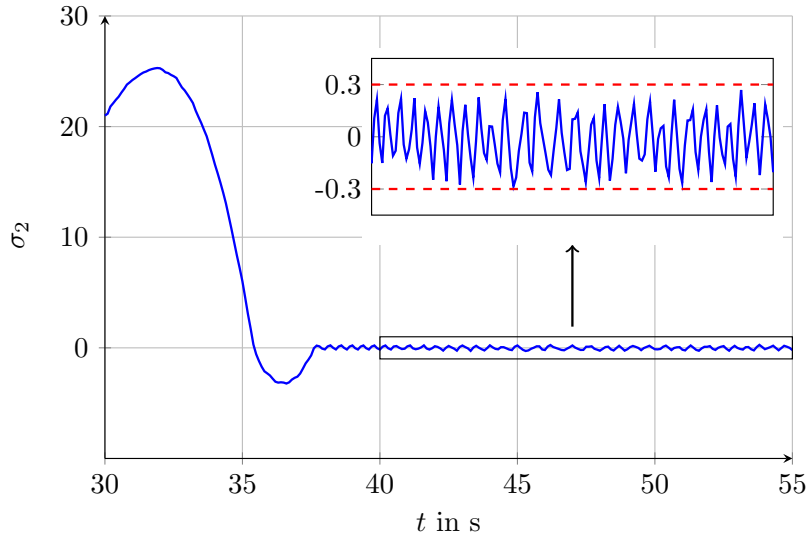


Figure 7.18.: Sliding variable of the blue vehicle

Example 5 - Platoon Merge

A simulation study with 20 vehicles and control law (7.10) in the reaching phase has been performed, where each platoon consists of 10 vehicles. Two identical platoons are driving on adjacent lanes next to each other, where one platoon (blue lines in Figure 7.20) has to brake in front of the lane reduction. Similar to the scenario used in [110], the initial position errors are $e_{x,i}(t_0) = 40$ m to vehicles on the own lane and $v_i(t_0) = 20$ m/s for all $i = 1, \dots, N$, and the platoons are driving next to each other.

The predecessor selection in Algorithm 1 is executed when entering the control region at $x_c = 400$ m. Before entering the control region, constant velocities are maintained. The resulting topology is similar to the zipper's principle: the leader of the overall platoon is the leader of the left platoon, the first follower is the leader of the right platoon, the second follower is the first follower of the left platoon and so on. Then, the reaching strategy (7.10) with the constant distance spacing is switched on, where the desired constant distance is $\Delta_i = \Delta = 10$ m. Note that one platoon has thus a negative position error, and the other platoon has positive position errors. The control parameter is $k = 5$, the parameter of the sliding variable of the ATFC is chosen as $t_h^* = 1$ s as in Remark 11 of the previous chapter, and the convergence parameter of the constant distance sliding variable is $c = 1$ s to match this ATFC parameter. Using a constant distance sliding surface for small negative position errors does not yield satisfactory results as shown in Figure 7.20. Unfortunately, using hard deceleration to reach the ATFC as in (7.10) leads to collisions. Moreover, the vehicles have to stop in the merging zone due to decreasing velocities, which has to be avoided. Although the desired overall platoon length is decreased due to a smaller desired inter-vehicle distance, the platoon length first increases and the traffic flow is decreased significantly.

7. Formation Control in Highway Driving

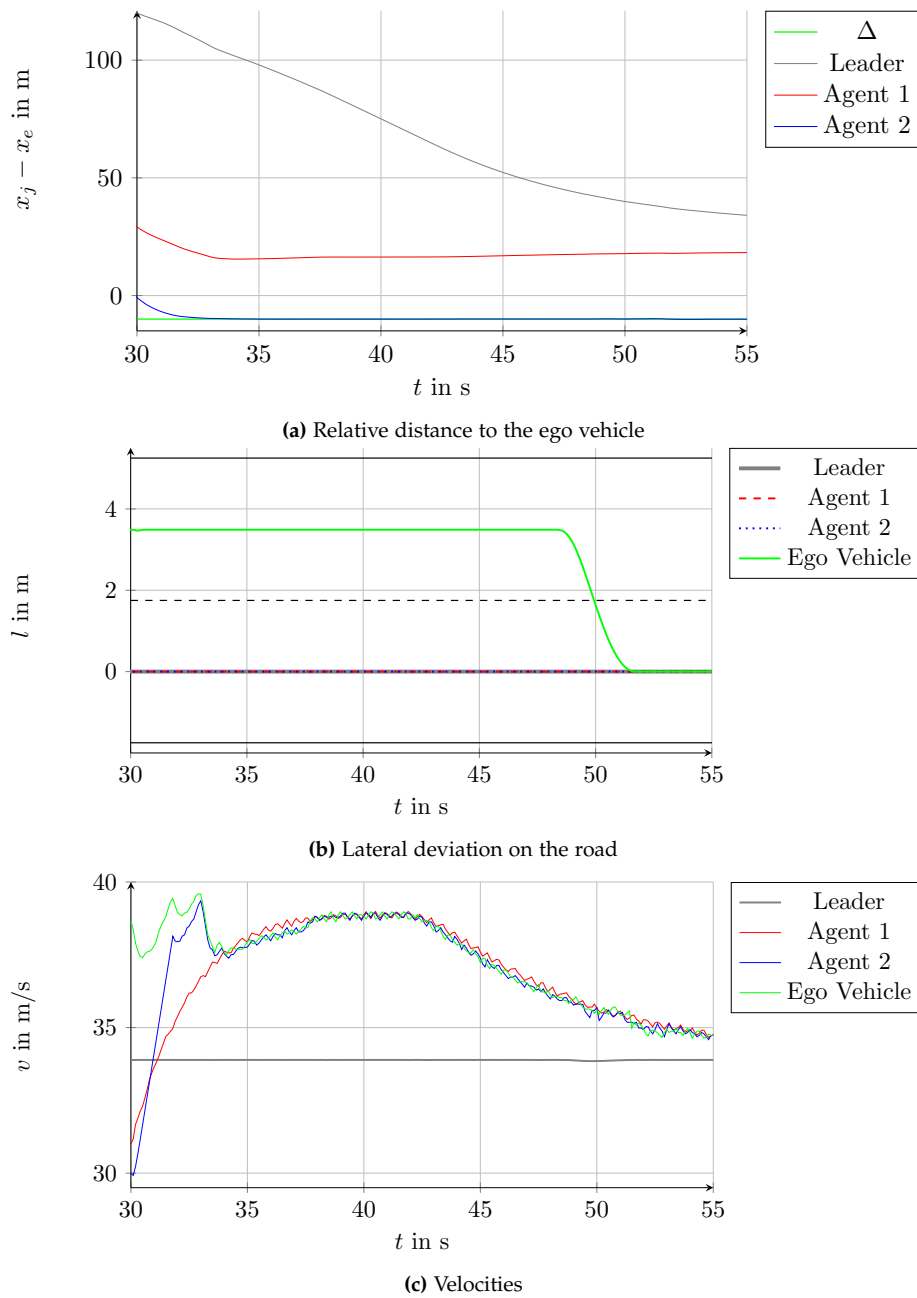


Figure 7.19.: Results of the proposed cooperative merging assist in Example 4 using SUMO

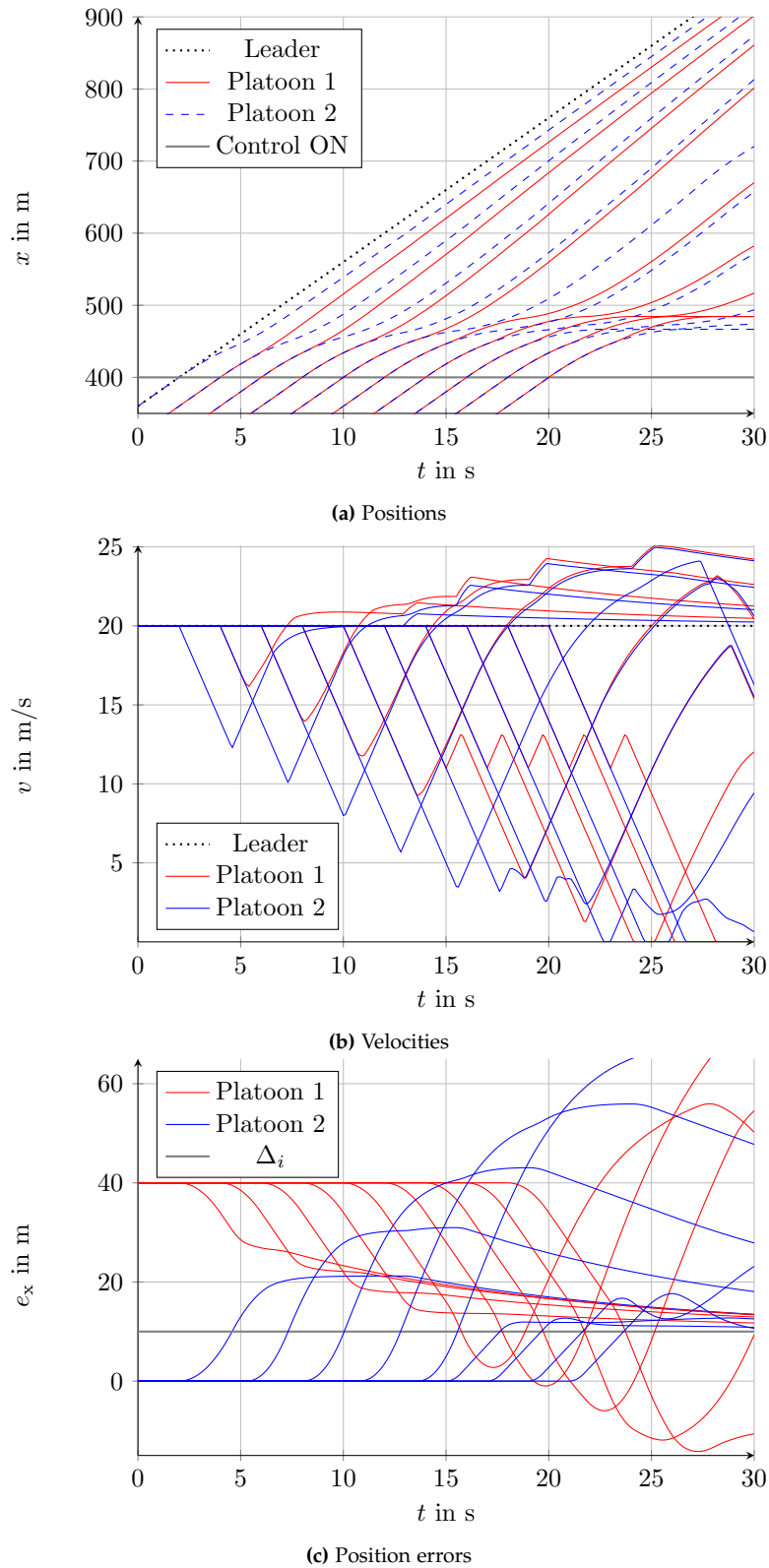


Figure 7.20.: Results of Example 5 using the constant distance sliding surface for negative position errors

7.2.2. Extension of the ATFC for Negative Initial Spacing Errors

In this section, the ATFC proposed in Section 6.4 is extended so that monotonic convergence can also be achieved for negative initial position errors as published in [146]. Since the sliding variable after predecessor selection is bounded according to (7.9) with (7.2), the new bounds on the errors have to satisfy

$$\begin{aligned} -\Delta_i - e_{v,i} &\leq e_{x,i} \leq t_{\max,i} v_i, \\ |e_{v,i}(t_0)| &\leq |t_{h,i}(t_0)| u_{\max}, \end{aligned} \quad (7.16)$$

with the initial time-headway $t_{h,i}(t_0)$ defined by (7.5), i.e., the time-headway parameter can also be negative. While the combined sliding variable of Phase I remains the same, the combined sliding variable of Phase II for $e_{x,i}(t_0) < 0$ is rewritten as

$$\tilde{\sigma}_i^- = e_{x,i} - t_{h,i} v_{i-1} + t_h^* e_{v,i}, \quad (7.17)$$

and the control law is again a first order sliding mode controller with $u_i = k \text{sign}(\tilde{\sigma}_i^-)$. The adaptation laws of the time-headways $t_{h,i}$ are chosen as

$$\dot{t}_{h,i} = \begin{cases} -\mu_1 \frac{k t_h^*}{v_i} & |t_{h,i}| > t_h^* \text{ (Phase I)} \\ -\mu_2 \frac{k}{v_{i-1}} t_{h,i} & \text{otherwise (Phase II)}, \end{cases} \quad (7.18)$$

with

$$\mu_1 = \begin{cases} \text{sign}(t_{h,i}) & -\frac{t_h^* k}{2} \leq e_{v,i} \text{sign}(t_{h,i}) < 0 \\ 0 & \text{otherwise,} \end{cases} \quad (7.19)$$

$$\mu_2 = \begin{cases} 1 & -\frac{t_{h,i} k}{2} \leq e_{v,i} \text{sign}(t_{h,i}) < 0 \\ 0 & \text{otherwise.} \end{cases} \quad (7.20)$$

The proof of Phase I is analogous to the standard ATFC.

Using the ATFC with this extension, it is possible to divide the phase plane of the position errors into four regions as shown in Figure 7.21 for a fixed time-headway $t_{h,i} = 1$ s and $k = 5$. In the white region, the standard ATFC can be applied. In the orange region, the extended ATFC with negative time-headway can be applied. In order to decrease large velocity errors, acceleration of the following vehicle is required in the yellow region, while in the red region, the vehicles have to decelerate. Hence, the following control law is proposed in the yellow/red region,

$$u_i = k \text{sign}(e_{v,i}) \quad \text{if ATFC OFF,} \quad (7.21)$$

with ATFC OFF if the bounds (7.4), (7.16), (7.5) are not fulfilled. With (7.21), the dynamics of the velocity error are governed by

$$\dot{e}_{v,i} = u_{i-1} - k \text{sign}(e_{v,i}), \quad (7.22)$$

which results in a decrease, or at least a non-increase in the velocity error because

$$e_{v,i} \dot{e}_{v,i} = e_{v,i} (u_{i-1} - k \text{sign}(e_{v,i})) \leq 0, \quad (7.23)$$

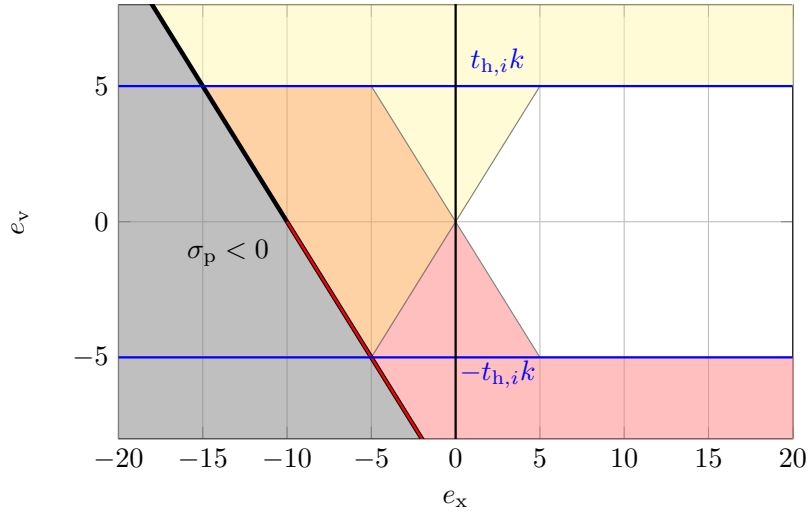


Figure 7.21.: Phase plane of the position errors. The extended ATFC is applied in the orange region and in the white region (shown for $t_{h,i} = 1$ s and $k = 5$). In the yellow and red regions, the vehicles have to decrease the velocity errors.

due to $u_{i-1} \leq k$. The white or orange regions in Figure 7.21 can eventually be reached so that the ATFC is switched on. Then, the constant distance formation is approached, while the accelerations are bounded and the position errors are not amplified. Hence, position error undershoots are avoided and thus safety can be guaranteed due to the string stable behavior of the ATFC. In the case that the ATFC cannot be switched on, the position error will either increase in the yellow region ($e_{v,i} > 0$) and the vehicle will be lost eventually, or decrease in the red region ($e_{v,i} < 0$) so that the gray region is entered, and the vehicle will no longer be considered as preceding agent. These cases only apply if the vehicles are on the adjacent lane according to Assumption 3 and hence no collisions can occur. The resulting trajectories are shown in Figure 7.22.

Remark 19. Note that in the red and yellow regions, maximum accelerations (7.21) are applied; however, compared to the constant distance sliding surface with control input (7.10), the performance is much better with respect to position error overshoot.

Remark 20. An alternative idea is to steer the errors into the white or orange region by using the TP: since the goal is to reach the region where $|e_{x,i}| > |e_{v,i}|$ holds, the new desired velocity for the TP level can, e.g., be set to

$$v_{\text{des}} = v_{i-1} - \rho |e_{x,i}|, \quad (7.24)$$

with $0 \leq \rho < 1$.

To summarize the different stages of the longitudinal control, a state diagram of the proposed cooperative merging assist is shown in Figure 7.23, where additionally an emergency maneuver may be triggered if a critical situation in the reaching phase is encountered. In the remainder of this chapter, the influence of disturbances and unmodeled dynamics on the longitudinal control performance is analyzed.

7. Formation Control in Highway Driving

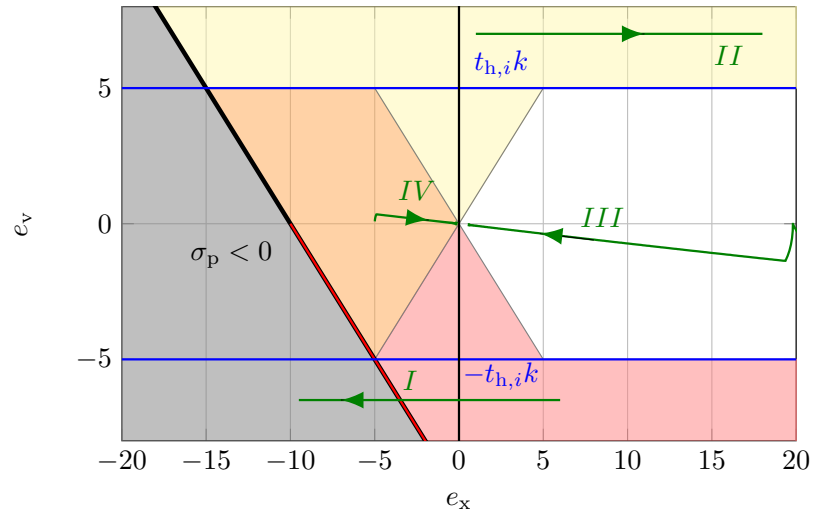


Figure 7.22.: Phase plane of the position errors with different trajectories. The extended ATFC is applied in the orange region and in the white region (shown for $t_{h,i} = 1$ s and $k = 5$). In the yellow and red regions, the vehicles have to decrease the velocity errors.

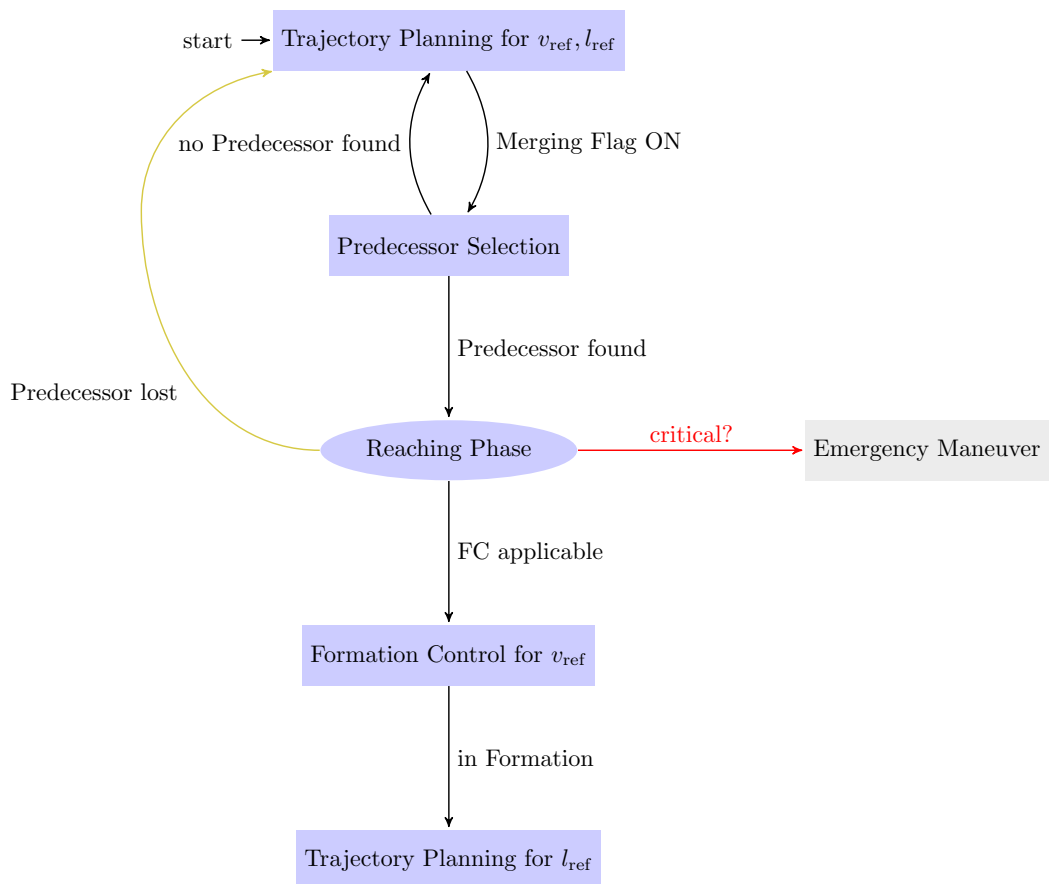


Figure 7.23.: State diagram of the proposed cooperative merging assist

Example 6

The extended ATFC for cooperative merging has been applied to two identical platoons driving next to each other as in Example 5, again with initial position errors $e_{x,i}(t_0) = 40$ m to vehicles on the own lane and $v_i(t_0) = 20$ m/s for all $i = 1, \dots, N$. The same predecessors have been selected, but the reaching strategy has been replaced. Note that with the ATFC, the platoon length can be decreased immediately with decreasing inter-vehicle distance, and the traffic performance is improved significantly as shown in Figure 7.24. There is no position error undershoot or overshoot.

Note, however, that although a certain distance is reached at approximately the same position as in Figure 7.24d, no lower convergence rate bound has been considered so far, which will be subject to future work. Moreover, the velocities of the vehicles increase along the platoon in order to reach the desired small inter-vehicle distance. This increase is expected, since the vehicles have to reach the platoon; in practice, speed limits may have to be considered, which has not been implemented yet.

7. Formation Control in Highway Driving

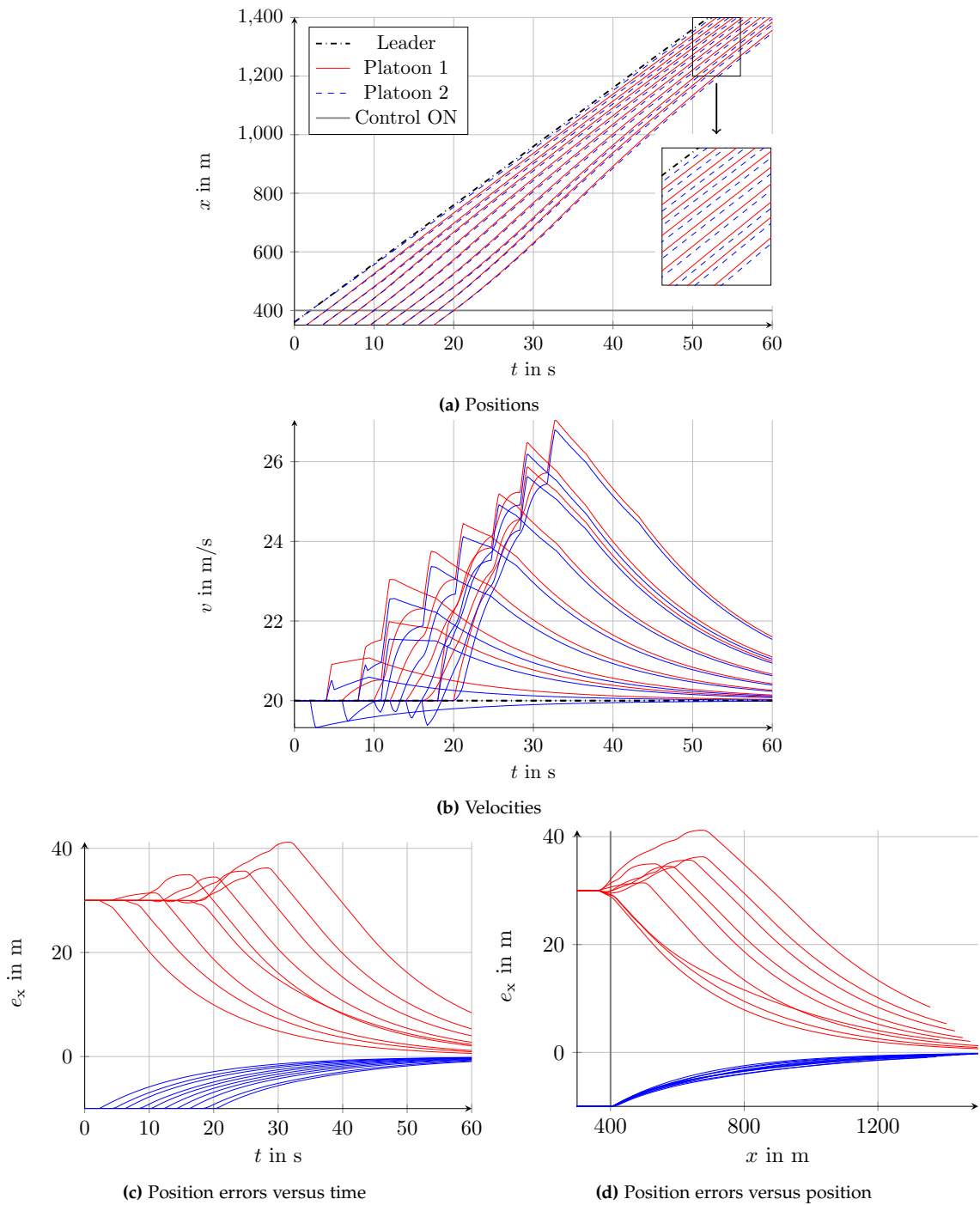


Figure 7.24.: Results of the platoon merging scenario using the proposed cooperative merging control with the ATFC extension for negative position errors

Example 7

In a last experiment, 40 vehicles have been placed randomly, with the condition that collisions can be avoided when the merging zone is entered according to Assumption 3. Then, the predecessor selection algorithm is applied and the vehicles have to merge at the assigned position. A reasonable predecessor choice can be observed in the simulation results in Figure 7.25, since the vehicles on one lane (red or blue) do not collide with each other. The length of the overall platoon is not increased, which improves the traffic flow. Note that the last vehicle has no predecessor due to a large position error upon entering the merging zone. For this last vehicle, the trajectory planning level may be used to speed up to a desired merging velocity. In the last example, the velocities of the involved vehicles are not bounded, which will be considered in future work. The traffic flow of the merging scenario is again significantly improved, and no collisions occur.

7. Formation Control in Highway Driving

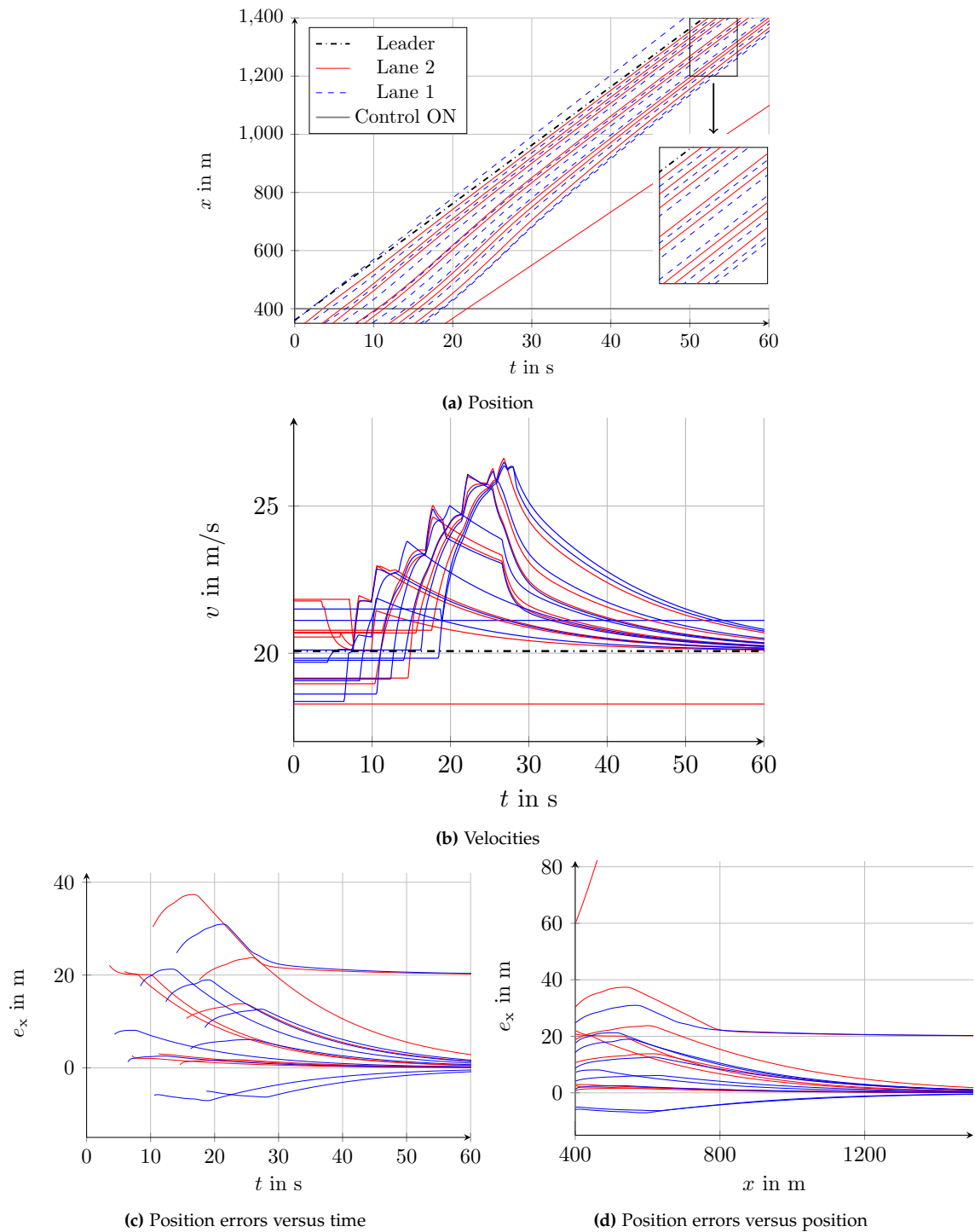


Figure 7.25.: Results using the proposed cooperative merging control for merging with random initial errors and the ATFC extension for negative position errors.

7.3. Robustness with Respect to Curvature

In merging scenarios, a lane change can be performed by the lateral guidance of the vehicles as described in Part 1. In this case, it is important that the longitudinal controller maintains the formation also during a lane change or in road bends. Due to the different curvature of the driven paths, disturbances arise that have to be compensated by the longitudinal controller in order to guarantee safe merging maneuvers.

Consider a road segment depicted in Figure 7.26 with constant curvature $\kappa_1 = 1/R_1$, i.e., a segment of a circle with radius R_1 , where the radius is defined with respect to the center of the inner lane. The curvature on the second lane is then defined by $\kappa_2 = \frac{1}{R_2}$, with $R_2 = R_1 + w_{\text{lane}}$ and lanewidth w_{lane} . If the relative position between the vehicles should

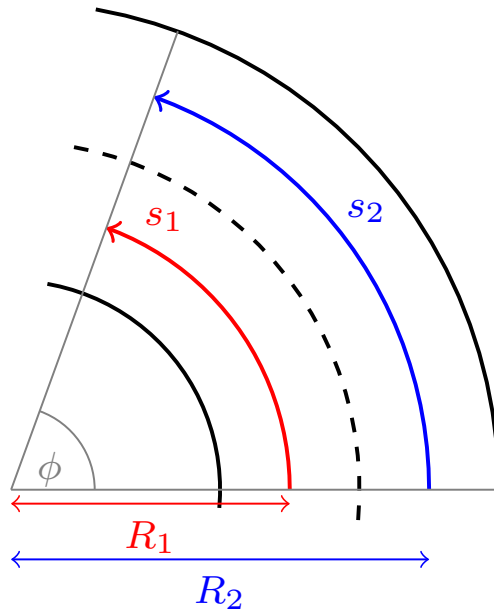


Figure 7.26.: In road bends, the vehicle on the outer lane (blue) has to drive faster than the vehicle on the inner lane (red) in order to keep a constant angular velocity

be maintained, the vehicle on the outer lane has to drive at a faster velocity than the vehicle on the inner lane, due to a larger waylength s_i according to

$$s_i = R_i \phi , \quad (7.25)$$

with angle ϕ . Then for constant curvatures, the requirement for a constant angular velocity results in

$$\dot{\phi} = \frac{\dot{s}_1}{R_1} = \frac{\dot{s}_2}{R_2} , \quad (7.26)$$

and thus, the velocities \dot{s}_1, \dot{s}_2 are related by

$$\dot{s}_1 = \dot{s}_2 \frac{R_1}{R_2} . \quad (7.27)$$

7. Formation Control in Highway Driving

This means that the vehicle on the outer lane has to maintain a faster speed than the vehicle on the inner lane in order to maintain the same angular velocity. Then, the longitudinal distance between the vehicles can be maintained. Hence, the curvature can be interpreted as a disturbance δ_i on the longitudinal velocity and is modeled as an additive disturbance subsequently. The agent's longitudinal dynamics are then given by

$$\begin{aligned}\dot{x}_i &= v_i + \delta_i, \\ \dot{v}_i &= u_i,\end{aligned}\tag{7.28}$$

where x_i, v_i, u_i denote position, velocity and acceleration of vehicle i , respectively. The curvature typically increases linearly on highway roads [105], i.e., they are typically designed as clothoid segments as depicted in Figure 7.27a. A point (x, y) on the clothoid is given by

$$\begin{aligned}x(l) &= A\sqrt{\pi} \int_0^l \cos\left(\frac{\pi t^2}{2}\right) dt, \\ y(l) &= A\sqrt{\pi} \int_0^l \sin\left(\frac{\pi t^2}{2}\right) dt,\end{aligned}\tag{7.29}$$

with parameter l . The length of the path is given by $L = Al\sqrt{\pi}$ and the curvature is $\kappa(l) = \frac{\sqrt{\pi}}{A}l$ for a constant parameter $A > 0$.

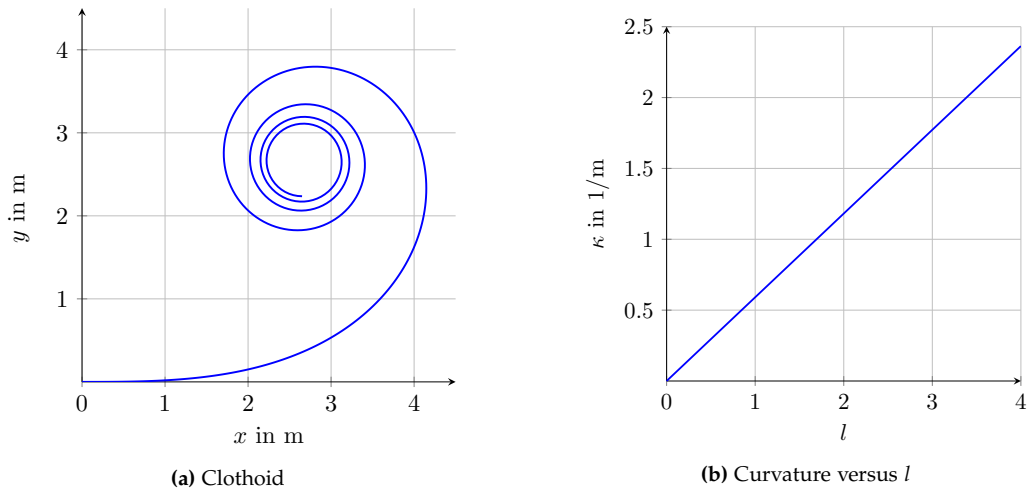


Figure 7.27.: Clothoids and corresponding curvature as a function of the parameter l , which are typically used for design of road segments.

For a road consisting of such clothoid segments, the curvature is differentiable almost everywhere, and the vehicle's longitudinal dynamics are affected by a matched disturbance $\tilde{\delta}_i$,

$$\begin{aligned}\dot{x}_i &= v_i, \\ \dot{v}_i &= u_i + \dot{\delta}_i = u_i + \tilde{\delta}_i,\end{aligned}\tag{7.30}$$

which is bounded by $|\tilde{\delta}_i| < \Gamma_i$. If the acceleration of the preceding vehicle is small such that $|u_{\text{eq},i}(t) + \tilde{\delta}_i| \leq u_{\text{max}}$ holds, then the formation can be maintained even if the vehicles are on different lanes.

Similarly, a lane change can also be considered as a disturbance that arises from different curvatures. During a lane change, the lateral position of the vehicle on the road y_i is changed in a smooth way due to a bounded lateral acceleration as discussed in Part I. Note that the states x_i, v_i in (7.30) are considered with respect to the longitudinal road coordinates as depicted in Figure 7.28. For the longitudinal control in (7.30) using sliding mode techniques

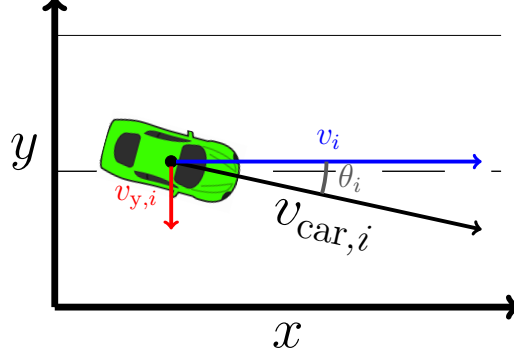


Figure 7.28.: Road coordinate system with corresponding velocities

from Chapter 6, it is assumed that the relative distance to other objects $\tilde{e}_{x,ij} = x_j - x_i$ and velocities v_i, v_{i-1} are available with respect to the road coordinate system.

The lateral velocity of the vehicle $v_{y,i}$ as shown in Figure 7.28 is larger than zero during a lane change, and the velocity of the vehicle $v_{car,i}$ is not equal to the road velocity v_i in (7.30); specifically,

$$v_{car,i} = \sqrt{v_i^2 + v_{y,i}^2}. \quad (7.31)$$

Assuming a point mass, the vehicle can be described in the x, y coordinate system by

$$\begin{aligned} v_i &= \dot{x}_i = v_{car,i} \cos(\theta_i), \\ v_{y,i} &= \dot{y}_i = v_{car,i} \sin(\theta_i), \\ \dot{\theta}_i &= \omega_i, \end{aligned} \quad (7.32)$$

where θ_i is the orientation with respect to the x axis as depicted in Figure 7.28 and ω_i is the angular velocity. The acceleration of the car $\dot{v}_{car,i} = u_{car,i}$ and the acceleration u_i in (7.30) differ as well. By differentiation of (7.32), the dynamics along the x -axis are

$$\begin{aligned} \dot{v}_i &= \dot{v}_{car,i} \cos(\theta_i) - v_{car,i} \sin(\theta_i) \dot{\theta}_i \\ &= u_{car,i} \cos(\theta_i) - \dot{y}_i \dot{\theta}_i \\ &= f(v_i) u_{car,i} - \tilde{d}_i. \end{aligned} \quad (7.33)$$

The orientation θ_i is assumed to be small, which is practicable on highways, thus $\cos(\theta_i) \approx 1$, $\sin(\theta_i) \approx 0$. Then, $0 < f_{\min} \leq |f(v_i)| \leq 1$. Assuming a bounded ω_i , the matched disturbance \tilde{d}_i is bounded, i.e., there exists a D_i such that $|\tilde{d}_i| < D_i$. Since u_i is bounded, one can simplify these disturbances to $\dot{v}_i = u_i + \tilde{d}_i$ and $|\tilde{d}_i| \leq \tilde{D}$, which yields the dynamics

$$\begin{aligned} \dot{x}_i &= v_i, \\ \dot{v}_i &= u_i + \tilde{d}_i. \end{aligned} \quad (7.34)$$

7. Formation Control in Highway Driving

This means that the longitudinal formation can be maintained while driving on a curved road or while merging onto one lane, if the disturbances $\tilde{\delta}_i$ in (7.30) and/or \tilde{d}_i in (7.34) can be rejected by the control input u_i . In the formation with constant distance spacing and a sliding mode based controller as in Chapter 6, the control input is given by $u_{\text{eq},i} = u_r$. Then, if

$$u_r + \Gamma_i + \tilde{D}_i \leq u_{\text{max}} \quad (7.35)$$

holds, both disturbances can be rejected at the same time, if an appropriately designed controller is used, and the formation can be maintained.

Remark 21. *The lateral acceleration of the vehicle during a maneuver consists of the lateral acceleration from the curvature of the road that is bounded by Γ_i , and the lateral acceleration of the lane change that is bounded by \tilde{D}_i . Hence, the curvature of the road is considered in the trajectory generation in Part I, and a desired maximum acceleration is not exceeded. Thus, the disturbance arising from lane change is small if the disturbance from the road curvature is large, which improves the compensation of disturbances.*

A robust longitudinal controller has to be used in order to maintain the formation during a lane change or in bends. The disturbance $\hat{d}_i = \tilde{d}_i + \tilde{\delta}_i$ with (7.30) and (7.34) can be handled by the ATFC in certain scenarios; the sliding dynamics using the ATFC of Chapter 6 with the disturbed dynamics of one agent (7.34) reads as

$$\dot{\sigma}_i = e_{v,i} + (t_h^* - t_{h,i})u_{i-1} + t_h^*(k\text{sign}(\tilde{\sigma}_i) + \hat{d}_i) + \mu_2 k t_{h,i}. \quad (7.36)$$

Then, if condition

$$|e_{v,i} + (t_h^* - t_{h,i})u_{i-1} + t_h^*\hat{d}_i + \mu_2 k t_{h,i}| < t_h^* k \quad (7.37)$$

is fulfilled, sliding can be maintained. Note that (7.37) cannot be guaranteed in general for all times. However, the disturbance compensation is only important for small position errors. Then, the time-headways $t_{h,i}$ and the velocity errors are small. In the case that the longitudinal velocity is much larger than the lateral velocity, the maximum disturbance D_i is very small, especially if compared to the other variables in (7.37). If the vehicles' control is $u_i \approx u_r$ and (7.35) holds, then the disturbance can be compensated. Note that if the disturbance cannot be compensated, the position error to the preceding vehicle increases during the lane change due to a larger way-length, which is not safety critical. Further analysis of the ATFC in presence of disturbances is subject to future work.

Example 8

In order to test the robustness of the algorithm with respect to lane changes or different curvatures, an example is considered. Three vehicles as indicated in Figure 7.29 are simulated, where the green vehicle (Agent 2) has to merge in front of the blue car (Agent 1), to reach position A behind the leader. As a sampling time, $T_s = 0.001$ s is used. The resulting variables of the green vehicle are depicted in Figure 7.30, where the activation of the merging assist is indicated by a gray dashed line. Note that the path of the lane change

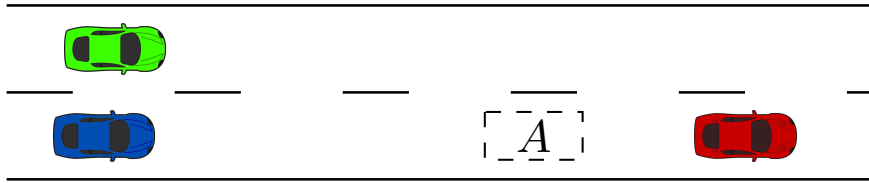


Figure 7.29.: The green vehicle changes lane when in formation (position A). The green and blue vehicle have to compensate the disturbance arising from the lane change.

in Figure 7.30a is chosen to start rather abruptly on purpose in order to investigate the effects of different lateral accelerations.

The velocity of the green car is larger during the lane change in order to maintain a constant velocity along the road as shown in Figure 7.30b. The position errors of the two followers are presented in Figure 7.30c, where no effects caused by the lane change can be detected. Note that the accelerations in Figure 7.30e have been filtered with a first order lag element with time constant $\tau = 0.01$ s. The disturbances are bounded and small as shown in Figure 7.30e. They can thus be rejected, and sliding can be maintained as shown in Figure 7.30d.

7. Formation Control in Highway Driving

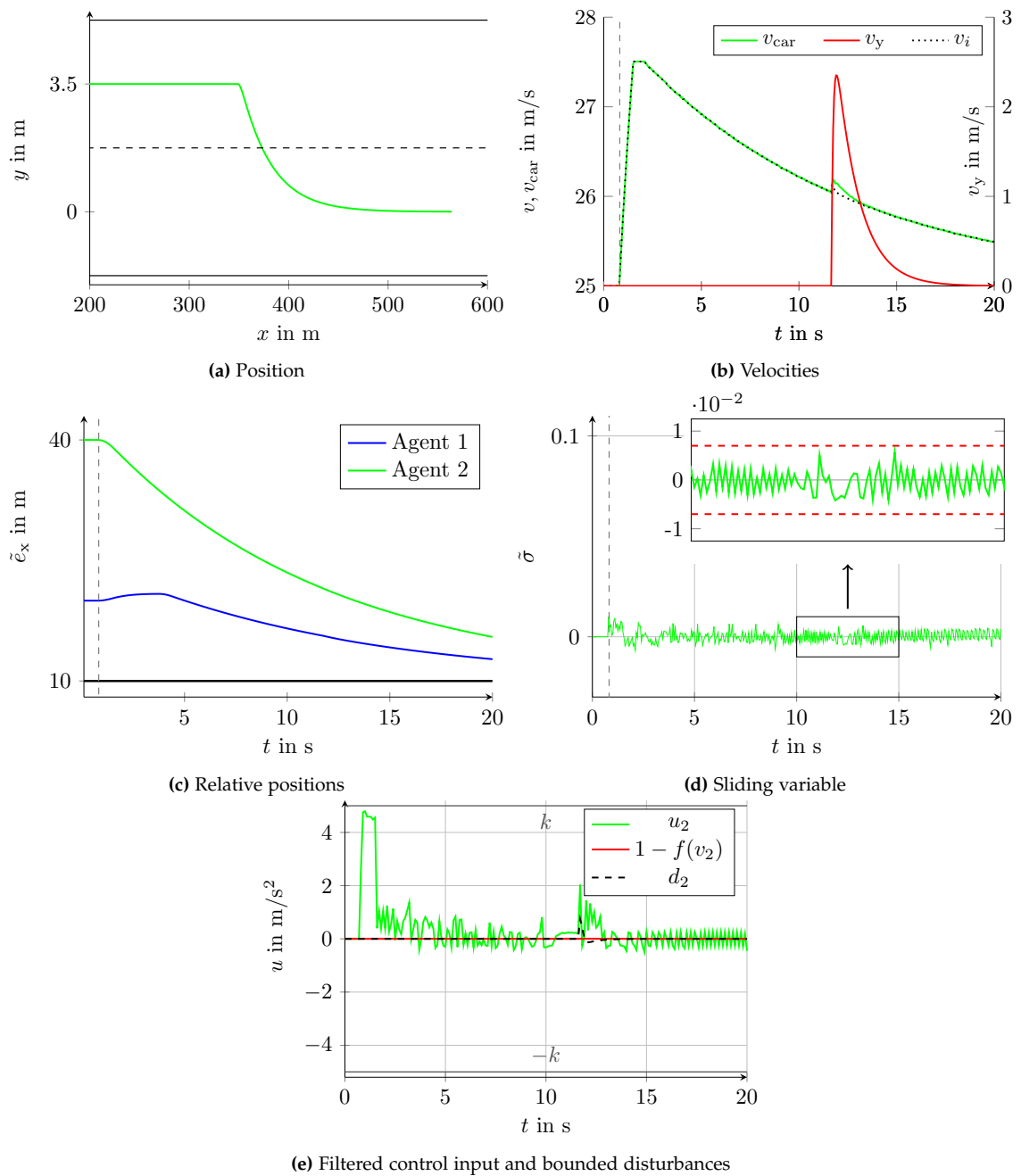


Figure 7.30.: Results of disturbance compensation during a lane change scenario

7.4. Actuator Dynamics

Consider the vehicle's longitudinal dynamics with first order actuator dynamics according to

$$\begin{aligned}\dot{x}_i &= v_i, \\ \dot{v}_i &= a_i, \\ \tau_i \dot{a}_i &= -a_i + u_i,\end{aligned}\tag{7.38}$$

where u_i is the control input, a_i is the actual acceleration of the vehicle, and τ_i is the time constant of the actuator dynamics. First, the constant distance spacing and the constant time-headway spacing are investigated (see also [147]). Then, simulation results using the ATFC are given.

7.4.1. Constant Distance Spacing

Consider a first order sliding mode based controller, denoted by

$$u_i = k \text{sign}(\sigma_{d,i}),\tag{7.39}$$

using the constant distance spacing sliding variable

$$\sigma_{d,i} = e_{x,i} + c e_{v,i}.\tag{7.40}$$

Then, the first derivative of (7.40) considering the dynamics (7.38) becomes

$$\dot{\sigma}_{d,i} = e_{v,i} + c e_{a,i},\tag{7.41}$$

with $e_{v,i} = \xi_{i-1} - \xi_i$, $\xi_{i-1} = x_{i-1}, v_{i-1}, a_{i-1}$, $\xi_i = x_i, v_i, a_i$, and the second derivative of the sliding variable reads as

$$\ddot{\sigma}_{d,i} = e_{a,i} \left(1 - \frac{c}{\tau}\right) + \frac{c}{\tau} u_{i-1} - \frac{c}{\tau} k \text{sign}(\sigma_{d,i}).\tag{7.42}$$

A block diagram of this relation is illustrated in Fig. 7.31. Note that the acceleration a_{i-1} of the preceding vehicle acts as disturbance on agent i . Compared to the previous considerations in this and the previous chapter, the relative degree of the sliding variable with respect to control input u_i is increased to two. Rewriting (7.42) with (7.40), (7.41), the dynamics of the sliding variable are given by

$$\ddot{\sigma}_{d,i} = -\frac{c}{\tau} k \text{sign}(\sigma_{d,i}) + \tilde{c} \dot{\sigma}_{d,i} - \frac{\tilde{c}}{c} \sigma_{d,i} + \underbrace{\frac{\tilde{c}}{c} e_{x,i} + \frac{c}{\tau} u_{i-1}}_f.\tag{7.43}$$

with the abbreviation $\tilde{c} = \frac{1}{c} - \frac{1}{\tau}$. With the state variables $z_1 = \sigma_{d,i}$, $z_2 = \dot{\sigma}_{d,i}$, the sliding variable dynamics can be written as

$$\begin{aligned}\dot{z}_1 &= z_2, \\ \dot{z}_2 &= -\frac{\tilde{c}}{c} z_1 - \tilde{c} z_2 + \frac{c}{\tau} k \text{sign}(z_1) + f.\end{aligned}\tag{7.44}$$

7. Formation Control in Highway Driving

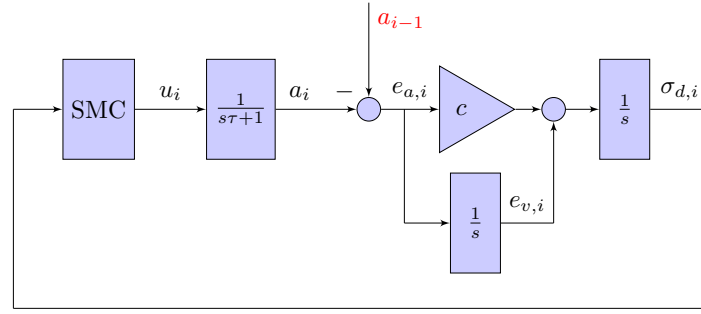


Figure 7.31.: Schematic of the closed-loop system using the constant distance spacing with actuator dynamics for vehicle-following

Then, $\bar{c} < 0$ and $k > 0$ is a necessary condition for asymptotic stability of the equilibrium $z_1 = z_2 = 0$, see [124]. For $f = 0$, the system can be maintained at $\sigma_{d,i} = 0$ and string stability is guaranteed. However, the disturbance f cannot be dominated by the control input (7.39), since all control inputs u_j are bounded by the same value, and no bound for $e_{x,i}$ exists. Note that using the FOSMC, oscillations may be excited for $|f| > \frac{c}{\tau}k$, which are then propagated along the string and can thus result in string unstable behavior.

The inter-connected system can also be formulated using the sliding variable representation

$$\sigma_{d,i} = \sigma_{i-1} - \sigma_i, \quad (7.45)$$

where the auxiliary variables σ_j are defined by

$$\sigma_j = x_j + c v_j. \quad (7.46)$$

The corresponding block diagram is depicted in Figure 7.32. Then, oscillations in one variable σ_{i-1} act as disturbance on the following agent, which cannot be compensated and are thus propagated along the string. Hence, string stability with constant distance spacing cannot be guaranteed in presence of actuator dynamics.

7.4.2. Constant Time-Headway Spacing

Using the constant time-headway spacing (5.16) as sliding variable,

$$\sigma_{t,i} = \ell_{t,i}, \quad (7.47)$$

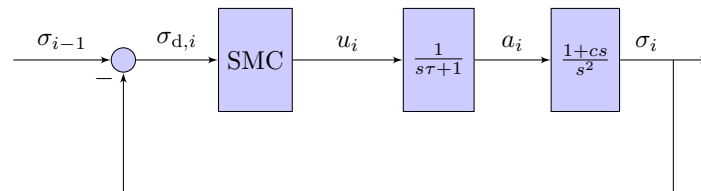


Figure 7.32.: Schematic of the auxiliary system for the sliding variable using a FOSMC with constant distance spacing and actuator dynamics in a platoon

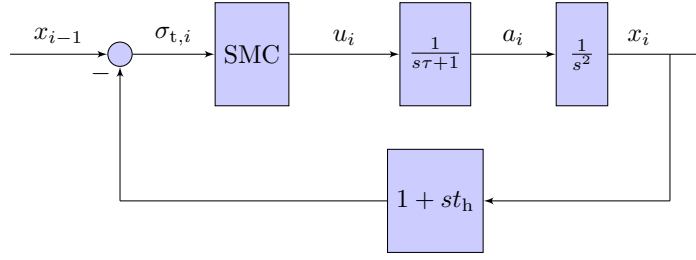


Figure 7.33.: Schematic of the closed-loop system using the constant time-headway spacing for one agent in a platoon

the closed-loop system is depicted in Fig. 7.33. The control input computed by the FOSMC reads as

$$u_i = k \text{sign}(\sigma_{t,i}), \quad (7.48)$$

and the dynamics of the sliding variable is given by

$$\ddot{\sigma}_{t,i} = a_{i-1} + a_i \left(\frac{t_h}{\tau} - 1 \right) - \frac{t_h}{\tau} k \text{sign}(\sigma_{t,i}). \quad (7.49)$$

With $t_h = \tau$, the sliding dynamics results in

$$\ddot{\sigma}_{t,i} = -k \text{sign}(\sigma_{t,i}) + a_{i-1}. \quad (7.50)$$

Note that the disturbance is the acceleration of the preceding vehicle, which is bounded by $|a_{i-1}| < k$ due to the first order lag element in (7.38) and due to control law (7.48). However, for $f \neq 0$, it cannot be guaranteed that sliding can be maintained. Note that (7.50) is a special case of (7.44) with $\tilde{c} = 0$ and $f = a_{i-1} + a_i \left(\frac{t_h}{\tau} - 1 \right)$, and oscillations occur in the case of any deviation from the origin. In addition, it cannot be guaranteed that sliding can be maintained for any $a_{i-1} \neq 0$. The disturbances are then propagated along the string of vehicles, yielding string unstable behavior. Thus, if the sliding variable is deflected from the surface using the FOSMC, string unstable behavior is encountered and the approach (7.39) for dynamics (7.38) is not robust.

Remark 22. *The SOC is robust with respect to these small deviations. However, unmodeled higher order dynamics may result in oscillations also when the SOC is applied. Thus, increasing the order of any sliding mode controller indefinitely is not reasonable. Instead, the parameter choice for t_h has to be investigated in future work.*

7.4.3. Adaptive Time-Headway Spacing - ATFC

Since a deflection from the sliding surface results in string unstable platoons, the effects of actuator dynamics (7.38) on the performance of a platoon with non-zero initial spacing errors using the ATFC have to be investigated. In Phase I of the ATFC, the sliding dynamics under consideration of actuator dynamics reads as

$$\dot{\sigma}_{t,i} = e_{v,i} - t_{h,i} a_i - \dot{t}_{h,i} v_i = e_{v,i} - t_{h,i} a_i + \mu_1 k t_h^*. \quad (7.51)$$

7. Formation Control in Highway Driving

The relative degree of the sliding variable with respect to the control input u_i increases by one, resulting in

$$\ddot{\sigma}_{t,i} = e_{a,i} - \dot{t}_{h,i}a_i - t_{h,i}\frac{1}{\tau}(-a_i + u_i). \quad (7.52)$$

Applying the FOSMC then yields

$$\ddot{\sigma}_{t,i} = -\frac{t_{h,i}}{\tau_i}k\text{sign}(\sigma_{t,i}) + a_{i-1} - a_i \left(1 - \mu_1\frac{kt_{h,i}^*}{v_i} - \frac{t_{h,i}}{\tau}\right), \quad (7.53)$$

which is equivalent to (7.49) for $\mu_1 = 0$, and string stable platooning is possible for a proper choice of the time-headway. For $\mu_1 = 1$, it is difficult to analyze the behavior of the platoon, since sliding cannot be maintained, which is essential in the proof of the ATFC.

The sliding dynamics in Phase II of the ATFC are governed by

$$\dot{\sigma}_i = e_{v,i} + t_h^*(a_{i-1} - a_i) - \dot{t}_{h,i}v_{i-1} - t_{h,i}a_{i-1}, \quad (7.54)$$

and the second derivative of the sliding variable reads

$$\ddot{\sigma}_i = e_{a,i} + \frac{t_h^*}{\tau_i}(u_{i-1} - u_i - a_{i-1} + a_i) + \mu_2k\dot{t}_{h,i} - \dot{t}_{h,i}a_{i-1} + \frac{t_{h,i}}{\tau_i}(a_{i-1} - u_{i-1}), \quad (7.55)$$

which can be rewritten as

$$\ddot{\sigma}_i = e_{a,i} \left(1 - \frac{t_h^*}{\tau_i}\right) + \frac{t_h^*}{\tau_i}(u_{i-1} - u_i) + \underbrace{\frac{\mu_2t_{h,i}(-k^2 + ka_{i-1})}{v_{i-1}} + \frac{t_{h,i}}{\tau_i}(a_{i-1} - u_{i-1})}_{\tilde{f}}, \quad (7.56)$$

which corresponds to (7.42) with an additional disturbance \tilde{f} . Hence, it is difficult to state string stability using the ATFC in Phase II, since the bounds of the disturbance are difficult to obtain. Hence, the behavior of the ATFC is studied in the course of simulation.

Simulation results using the FOSMC in Figures 7.34 and 7.35 for $\tau = 0.1$ show that even though sliding cannot be guaranteed, the ATFC may still yield collision-free results. Note that in the implementation, the time-headway is reduced if the sliding variable is within a band $|\sigma| < \epsilon$, thus the time-headway is decreased although the sliding variable is deflected slightly. However, small oscillations occur, which cannot be compensated and are propagated along the string, limiting the number of vehicles in the platoon.

Since the sliding dynamics (7.42), (7.49) are of the form $\ddot{\sigma} = h(t, x) + g(t, x)u$, a sliding mode controller for relative degree two sliding surfaces can be used. However, similar to the constant distance spacing, the sliding dynamics (7.55) using the proposed adaptation law includes disturbances that cannot be compensated by any controller with bounded acceleration.

Using a suboptimal controller yields the results shown in Figures 7.36 and 7.37 for $\tau = 0.1$. Note that collisions do not occur and the signals exhibit less oscillations, but sliding cannot be maintained. Again, the results are collision-free for the limited number of agents, but no general statement can be made.

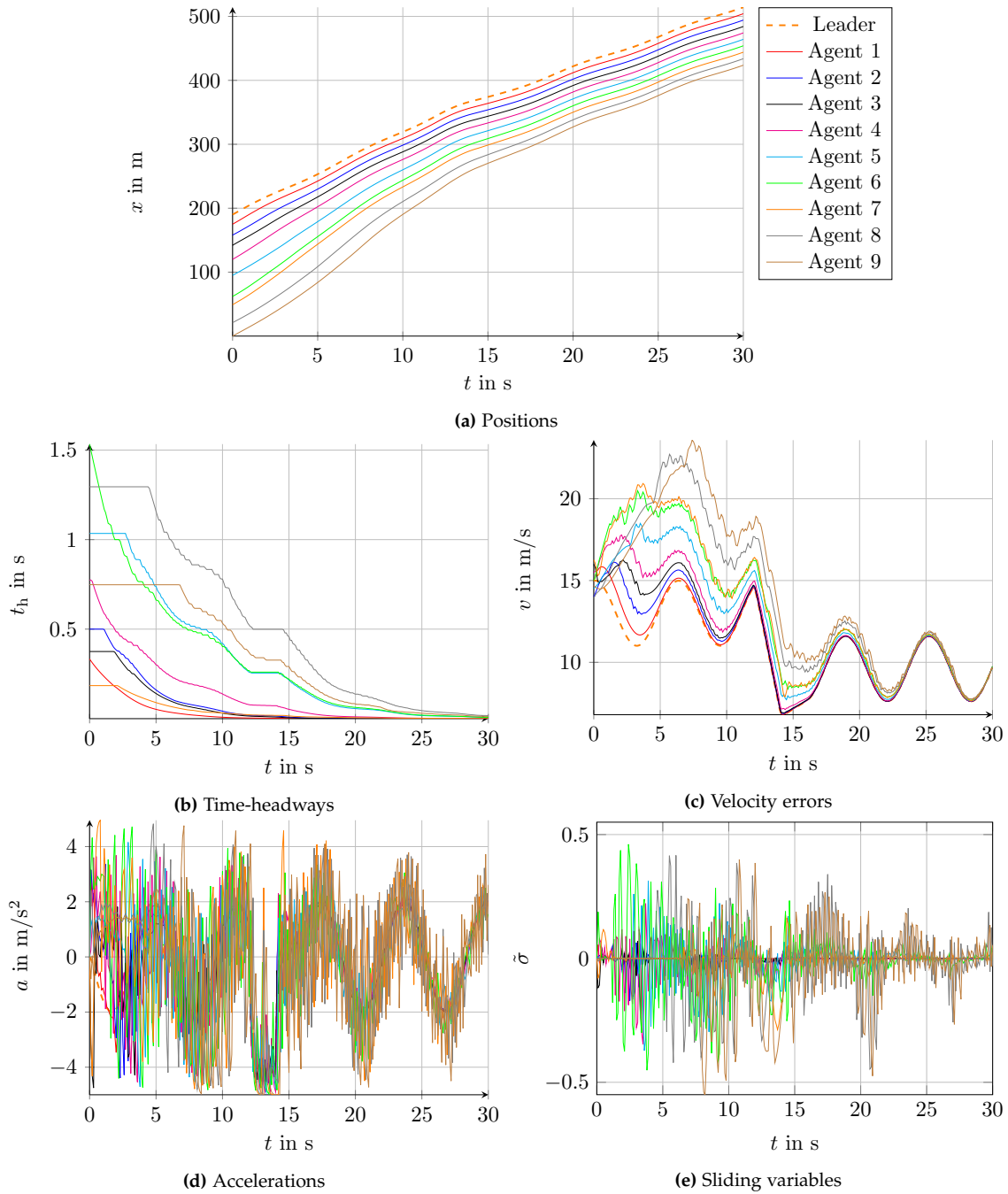


Figure 7.34.: Results using the sliding variable of the ATFC with the FOSMC in presence of actuator dynamics $\tau = 0.1$.

7. Formation Control in Highway Driving

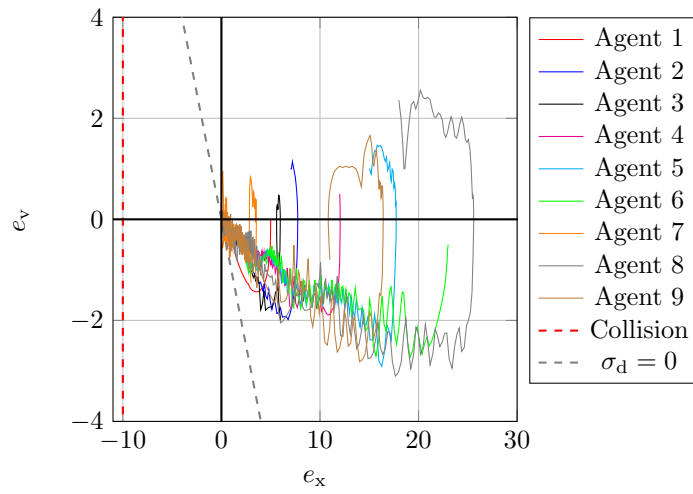


Figure 7.35.: Phase plane of the errors using the FOSMC with the sliding variable of the ATFC in presence of actuator dynamics $\tau = 0.1$

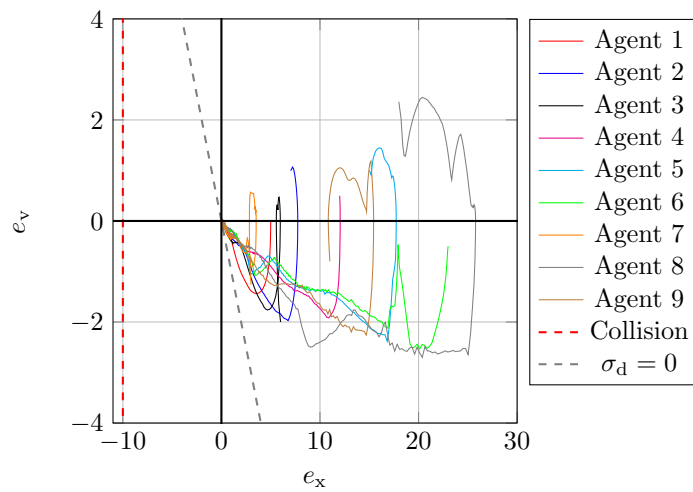


Figure 7.36.: Phase plane of the errors using the SOC with the sliding variable of the ATFC in presence of actuator dynamics $\tau = 0.1$

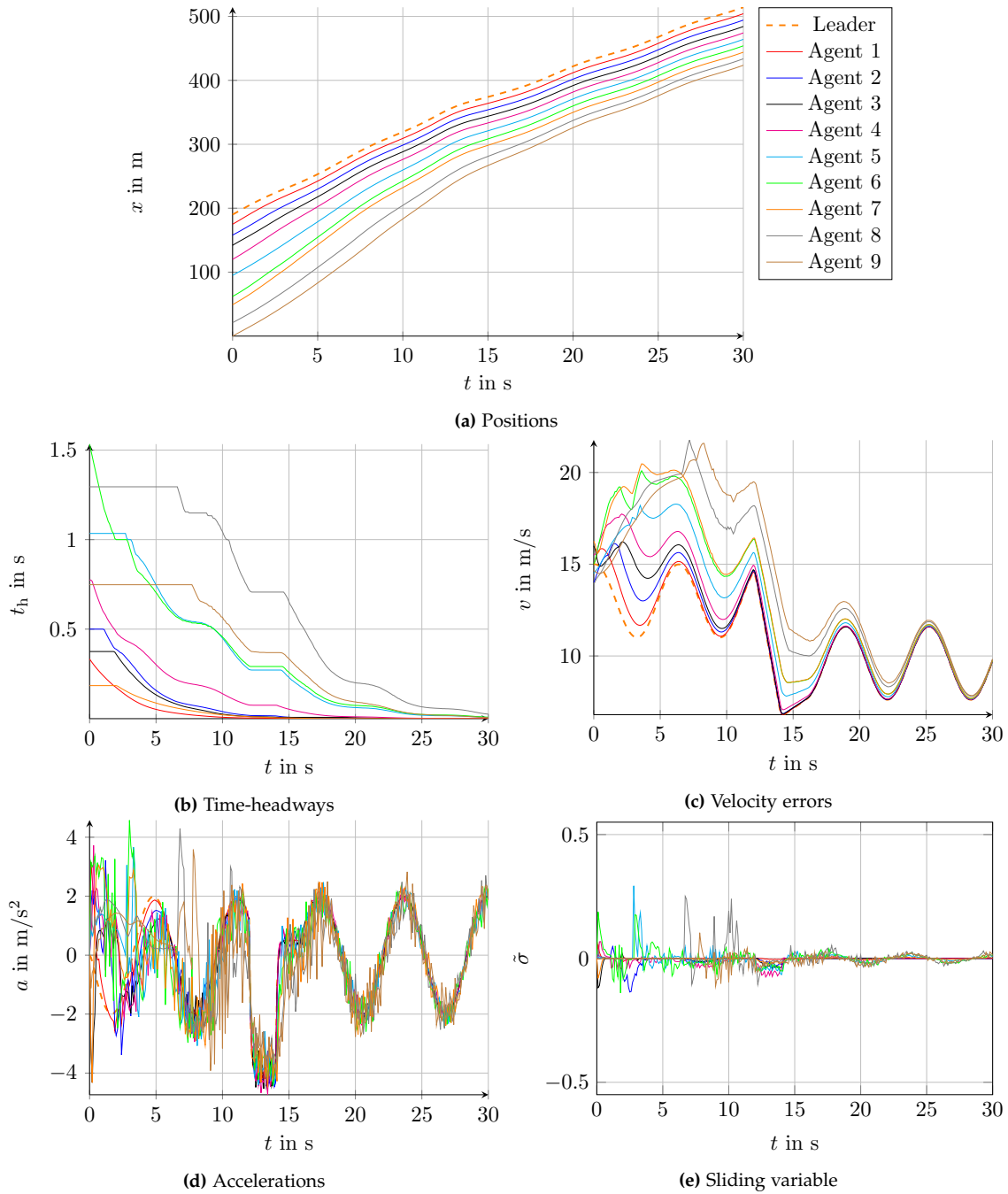


Figure 7.37.: Results using the sliding variable of the ATFC with the SOC in presence of actuator dynamics $\tau = 0.1$.

7. Formation Control in Highway Driving

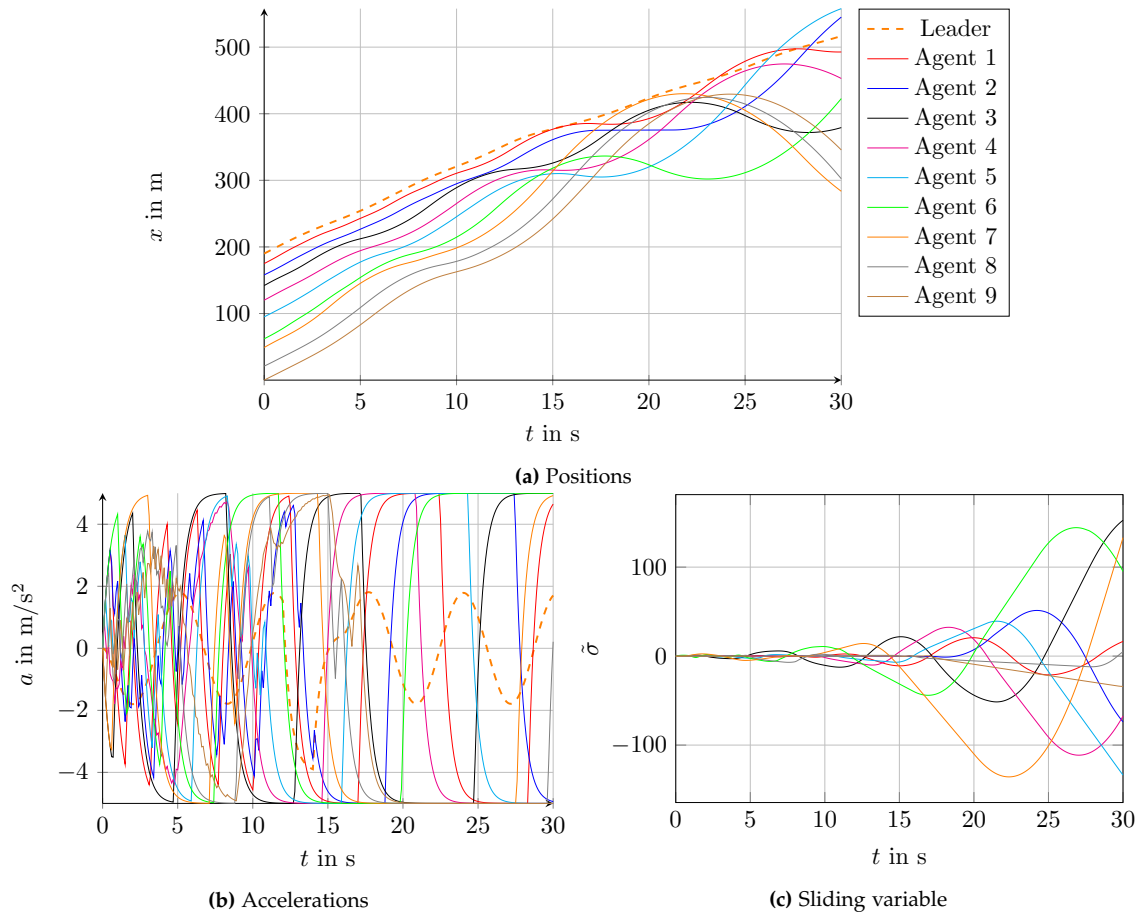
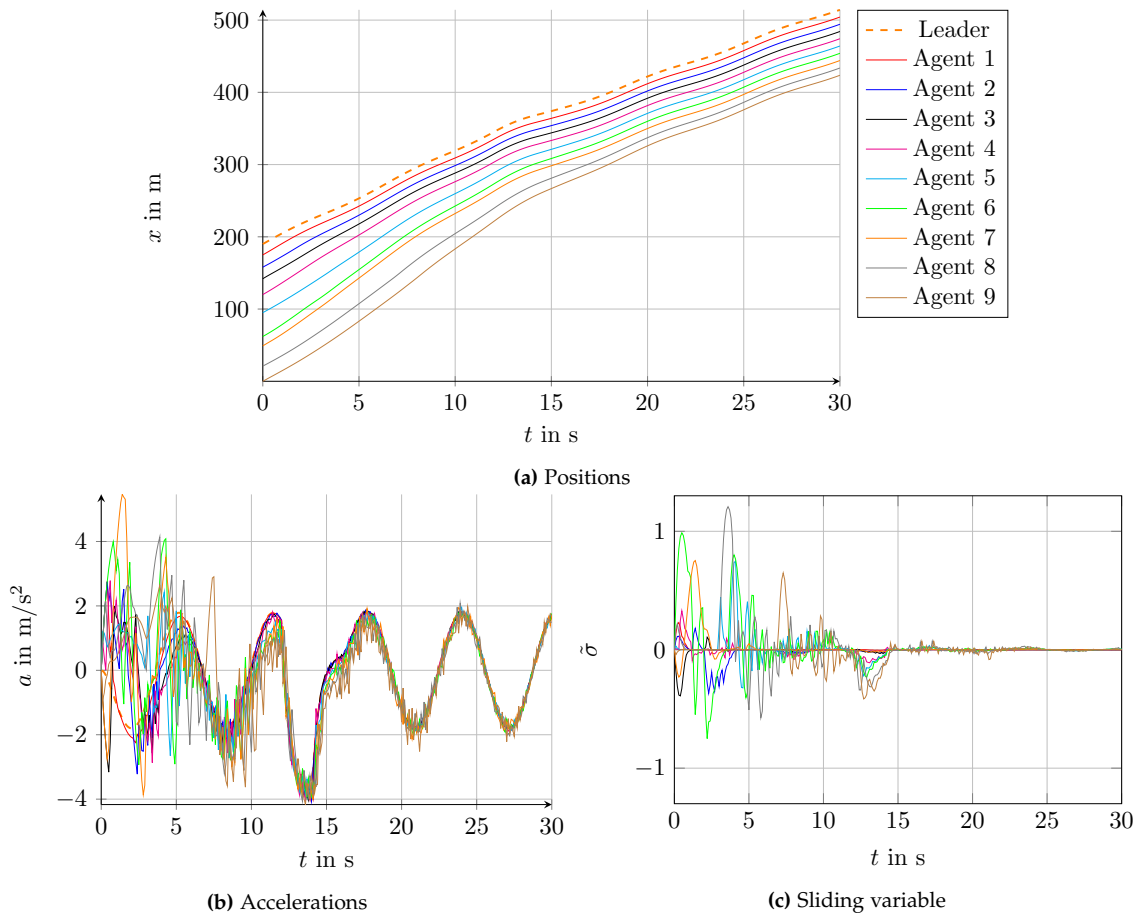


Figure 7.38.: Results using the FOSMC in presence of actuator dynamics $\tau = 0.5$

The last simulation example with $\tau = 0.5$ shows that larger time constants can still be handled by the SOC as shown in Figure 7.39, but the system with the FOSMC is unstable and results in collisions as in Figure 7.38.

Note that the oscillations in the signals are only present since sliding cannot be maintained; then, other agents are affected by this disturbance, which is amplified along the string. Although the performance of the SOC in the example is satisfactory, string stability cannot be guaranteed for an arbitrary platoon length.

Since the time-headway plays an important role in string stable platooning, future work will focus on reaching a desired time-headway spacing instead of a constant distance spacing. A proper choice of the time-headway in presence of unmodeled dynamics will be investigated.

Figure 7.39.: Results using the SOC in presence of slow actuators with $\tau = 0.5$

Part III.

Application to Small-Scale Vehicles

8. Testbed for Automated Driving using Small-Scale Vehicles

In order to test the proposed algorithms for automated driving of Part I and Part II of this work, experiments on real vehicles have been conducted¹. However, testing in real-world environments is difficult due to several reasons:

1. legislation: autonomous driving is forbidden by law on standard roads (Vienna Convention [21]); only designated test tracks can be used.
2. costs: the fully equipped car is expensive and replacing parts of the car is very costly.
3. safety: a test driver, which needs to have special training, has to be inside the car in order to take over control in case of a dangerous situation. The safety of people must be ensured at all times.
4. efficiency: due to the high organizational effort, testing the algorithms takes a lot of time. Moreover, all components have to work properly before testing the algorithms, i.e., sensor fusion and controllers of the vehicles have to be fully installed.

Hence, the concepts have been tested on small-scale vehicles. This enables safe testing at low cost, while the performance of the algorithms can be evaluated. A testbed for automated driving has been built up at the Institute of Automation and Control, Graz University of Technology, which is described in detail in this chapter.

Considering the issues of testing automated driving listed above, the following requirements for the setup of the testbed have been considered:

- + focus on control engineering
- + scalability (adding different vehicles or functionalities)
- + distributed systems: the algorithms run locally on the vehicles
- + real-time capability
- + affordable hardware
- + easy access

The testbed is similar to the one at KTH Stockholm [27], but uses a much cheaper motion capture system and is dedicated to autonomous ground vehicles only. The setup is shown in Figure 8.1 and the different components are described subsequently, where the description is divided into a hardware and a software related part.

¹ Astrid Rupp, Markus Tranniner, Marko Pavlic, Raffael Wallner, Martin Steinberger, and Martin Horn. A Low-Cost Testbed for Automated Driving. Submitted to *Journal of Robotics and Autonomous Systems* in May 2018 (under review).

8. Testbed for Automated Driving using Small-Scale Vehicles

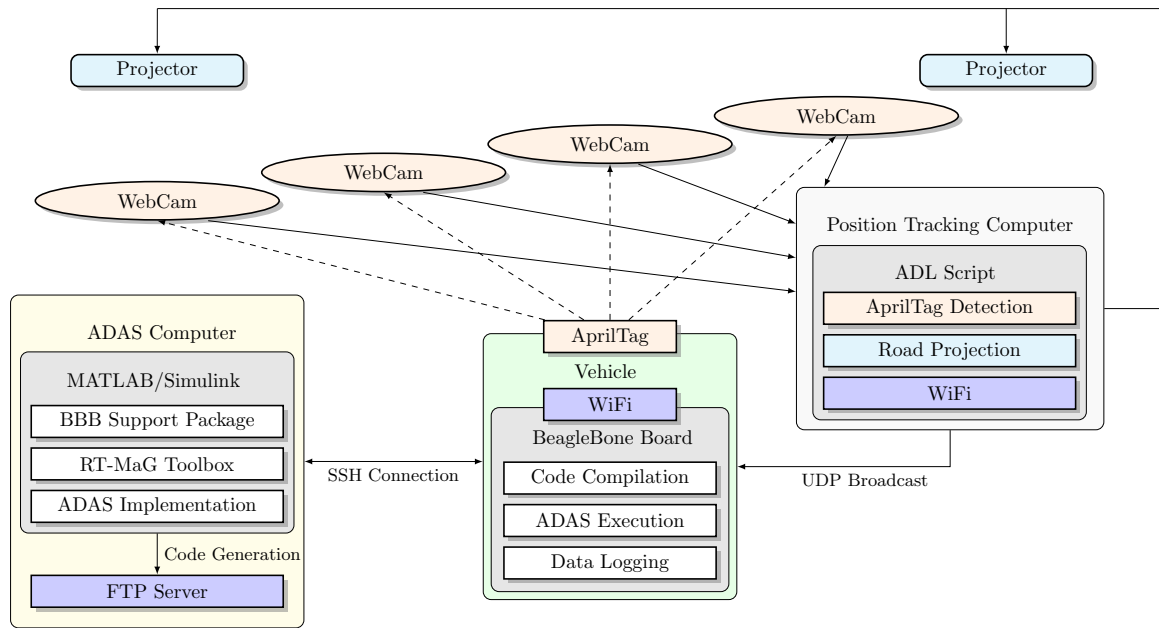


Figure 8.1.: Setup of the small-scale testbed for control engineering purposes

8.1. Hardware

Two different kinds of small-scale vehicles have been acquired: trucks for platooning maneuvers and parking with trailers, and model cars to test standard maneuvers. In this section, both types of model vehicles are presented briefly.

8.1.1. Trucks

In Figure 8.2, two trucks in scale 1:14 [12] are shown. Each vehicle has been equipped with a BeagleBone Black [14], which is a low-cost single-board computer. To receive the actual position in the room or to communicate with other vehicles, a WiFi module [15] has been installed.

The trucks are actuated by a motor and servos for gear changing and steering. It is also possible to replace these actuators by others, which makes the setup of the vehicles very flexible. The steering angle is within the range $|\delta_{\text{truck}}| \leq 17^\circ$, and a backlash has been observed. The servo motor for changing the gears is currently not used, since the trucks do not exceed a velocity of 1.5 m/s. This limit has been defined due to the limited space of the testbed.

8.1.2. Cars

Three cars in scale 1:10 [13] as in Figure 8.3 have been built up, which are also equipped with a BeagleBone Black and a WiFi module. However, the actuation differs from the trucks: the speed of the included motors depended strongly on the charge of the battery of the

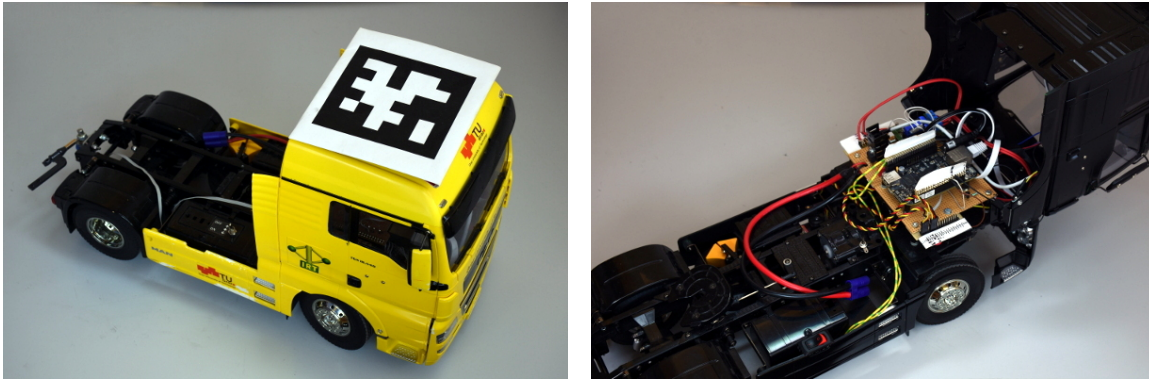


Figure 8.2.: Model trucks equipped with BeagleBone Black Boards in order to test different ADAS on the testbed

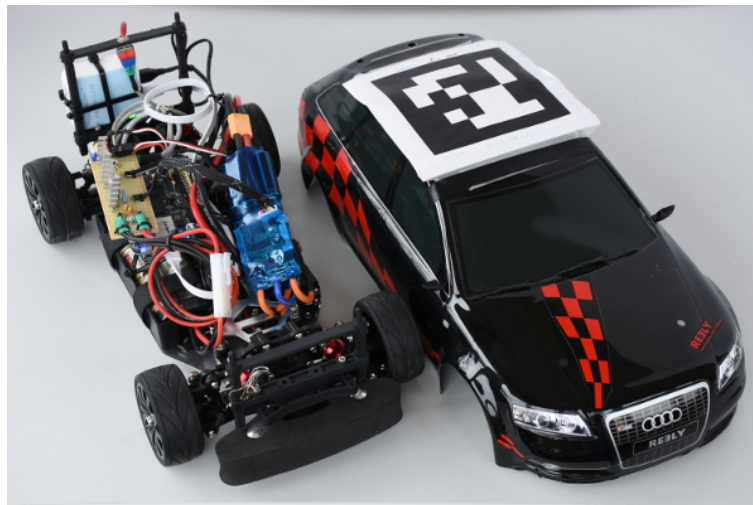


Figure 8.3.: Small-scale car, which is also equipped with BeagleBone Black Boards, and AprilTags [1] on top of the vehicle are used for position detection.

cars and did not yield accurate results. Hence, brushless DC (BLDC) motors with Hall sensors [16] and motor speed controllers [10] have been used to achieve better performance at low speeds. In addition, the transmission gears have been replaced, which also improves driving at low speeds. The steering of the cars is more accurate than the trucks' steering, but the maximum steering angle is in a similar range with $|\delta_{\text{car}}| \leq 20^\circ$.

8.1.3. Sensors

The following sensors are currently used for data acquisition:

- webcams on the ceiling and AprilTags [1] on the vehicles for position tracking, which emulates GPS and is described in Section 8.2,
- ultrasonic sensors, which are mounted on the front side of the vehicles for triggering emergency braking if necessary.

8. Testbed for Automated Driving using Small-Scale Vehicles

This setup is sufficient to study platooning and formation control as considered in this work. In the future, other sensors can be added in order to get an insight into sensor fusion algorithms, which is similar to the problem statements on real vehicles.

8.1.4. Actuation

In order to actuate the vehicles, a lookup table, where the velocity is a function of the PWM (pulse width modulation) width, has been created by measuring the velocity with a GoPro camera [11] for various PWM values. Since the control input of the previous considerations is the acceleration, an integrator has been added to obtain the velocity. For the steering actuation, no lookup table was necessary because of a linear relation between steering angle and PWM signal. The slope and the offset have been calculated by measuring the radius of the circle driven by the vehicle for different steering angles. Note that actuator dynamics are currently not considered in the controller design; modelling and parameter identification can be studied on the testbed in future work.

8.1.5. Projection of Virtual Components

In order to demonstrate virtually defined components such as lanes, virtual obstacles, or traffic signs, two ultra short throw projectors [17] have been mounted on the walls so that the whole area of the testbed can be illuminated. In addition, a white floor foil has been used in order to see the projections more clearly. In the current status of the testbed, only the road is projected onto the floor in order to easily demonstrate that, e.g., lane keeping assists work properly. In future work, different virtual environments, such as other traffic participants or traffic signs, can be projected.

8.2. Position Tracking

In order to test automated driving, it is necessary to obtain position information of the controlled vehicle and surrounding obstacles for appropriate motion planning. The positions of these moving objects are estimated via off-the-shelf webcams [9] and AprilTags [1], [119] that are mounted on the vehicle as shown in Figure 8.2 and 8.3. The position tracking code is based on the AprilTags C++ Library [2]. This library provides fast and robust 3D position estimation and the tracking algorithm is insensitive to bad light conditions, which is important due to glass walls and several reflective surfaces in the room.

The tag detection algorithm is executed on a dedicated Linux workstation, called “Position Tracking Computer”, which is connected to the webcams via USB as depicted in Figure 8.1. The AprilTag detection delivers the AprilTags’ positions and orientation with a rate of 10Hz and broadcasts the information via WiFi using UDP.

To cover a large area, four webcams with a slightly overlapping field of view as in Figure 8.5 covering an area of 8 m × 4 m are used in a room of 3 m height. The camera intrinsics for each webcam have been calibrated using AprilCal [138], which has to be done once for each

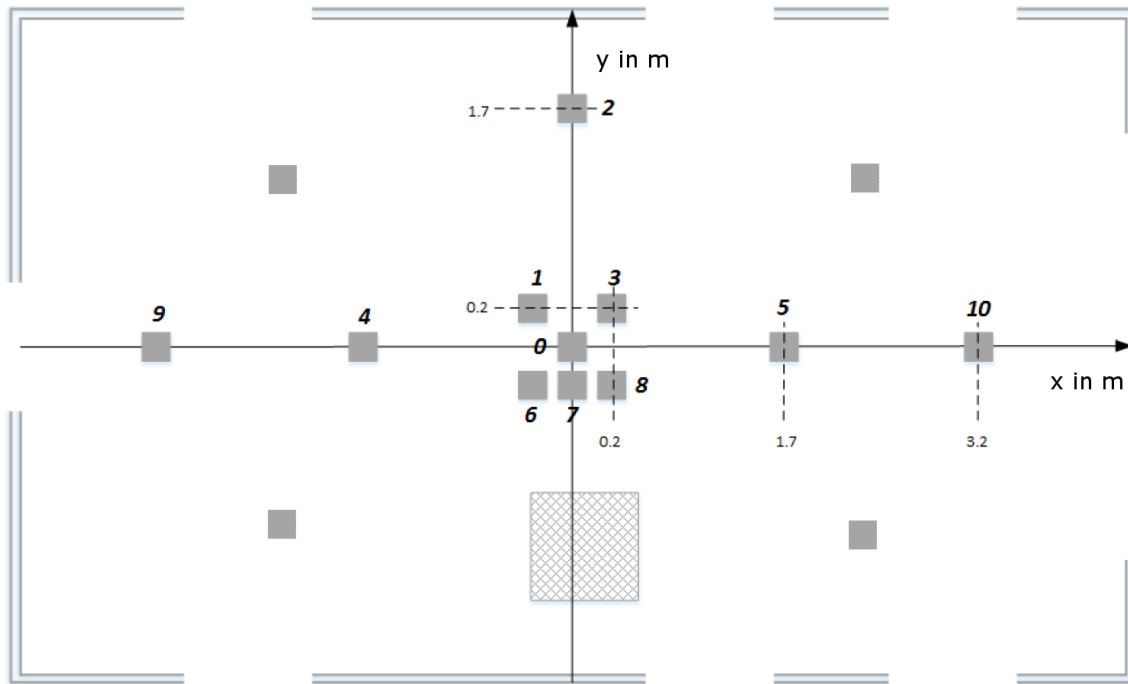


Figure 8.4.: Calibration of the planar (world) coordinate system. The gray squares represent tags for calibration, while the patterned square is a column in the room.

webcam. Note that for this testbed, it is sufficient to estimate a 2D position since the vehicles move in a planar environment. Hence, for the calibration of the camera extrinsics, a planar calibration target based on AprilTags is used. This target consists of different tags that are placed at known positions, which is schematically depicted in Figure 8.4. The calibration of the world coordinate system has to be executed if the position of a camera changes. This calibration can be done in a flexible and automatic manner using configuration files and thus it is straightforward to build up the testbed in a different environment (e.g., for presentation purposes).

An accuracy of approximately 0.03 m can be achieved, depending on the calibration of the webcams. For the experiments presented in Chapter 9, this accuracy is sufficient. Note, however, that in positions where several webcams detect the AprilTag, the average of the results is taken, which can result in non-smooth measurements. Moreover, the velocities of the vehicles are computed from the change in position that is computed by the AprilTag detection algorithm. Hence, the velocity computations have to be performed carefully: A rate limiter has been used in order to avoid too large velocities in presence of abrupt position changes. Moreover, if no position can be detected, but information is sent via UDP, this loss of information has to be treated separately, otherwise very large velocities are computed. Note that networked system analysis and sensor fusion techniques can be helpful in this setting, but are not part of this work.

8. Testbed for Automated Driving using Small-Scale Vehicles

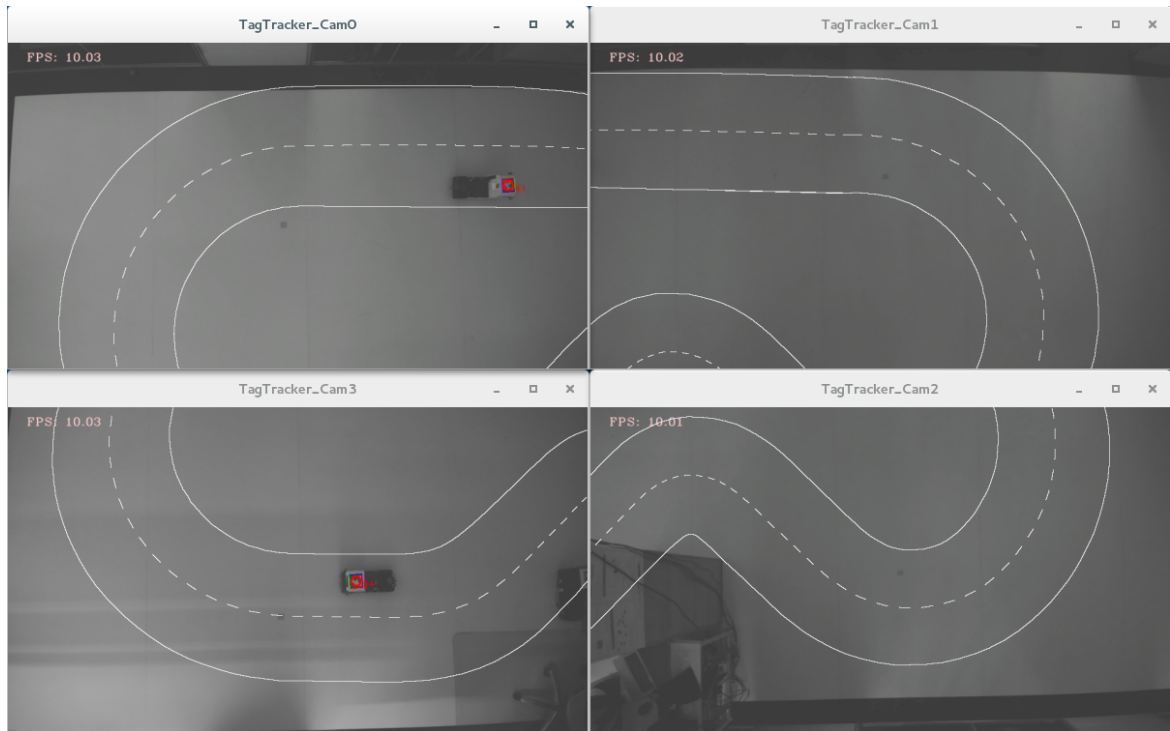


Figure 8.5.: Field of view of the webcams and the virtually defined road with one detected truck shown in red, as displayed on the position tracking computer.

Road Coordinates

The road is defined virtually as shown in Figure 8.6 and is stored on the vehicles, emulating a high definition road map for self-driving vehicles. Two lanes with lanewidth $w_1 = 0.5$ m are considered for highway scenarios, where the available space for the road is $8\text{ m} \times 4\text{ m}$ due to the coverage area of the webcams. A column in the room is indicated by a black rectangle in Figure 8.6. This column has to be avoided by the trucks, i.e., there is a lane reduction and all vehicles have to merge to the right lane in front of the column when driving in clockwise direction.

The road is not marked on the floor in order to allow various scenarios and road definitions, e.g., switching from a highway to an intersection or a parking lot, but it can be projected onto the floor to improve the demonstrations.

8.3. Software

Each vehicle is capable of executing ADAS functionalities on the BeagleBone Black in real-time. The focus is on the control engineer's tasks of ADAS such as planning, tracking or estimation techniques. Hence, an easy-to-use environment in MATLAB/Simulink [4] has been built up. The ADAS functionalities are modeled in MATLAB/Simulink and code is generated automatically on a Laptop (called "ADAS computer"), which is described in Section 8.3.2. The code is then downloaded and compiled on the BeagleBone Black and

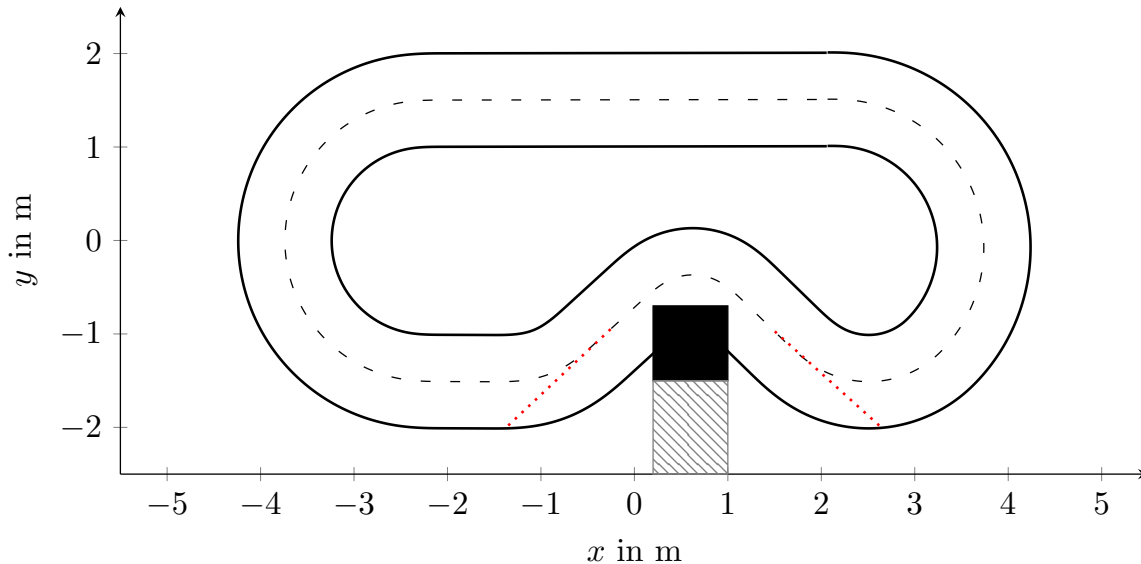


Figure 8.6.: Road coordinates defined for testing in the foyer of the institute. The black rectangle represents the column in the room, and is treated as an obstacle on one lane of the road.

executed locally on this on-board computer. The necessary software setup as depicted in Figure 8.1 is discussed subsequently.

8.3.1. Real-Time Operating System

On the BeagleBone Black, the algorithms are executed on a Debian Linux based operating system (OS) with the RT PREEMPT patch [8] (Version 4.4.x). This patch enables full preemption of the Linux kernel and thus the OS exhibits hard real-time behavior. However, in order to guarantee that the ADAS code generated in MATLAB/Simulink is executed in real-time, additional software is necessary. The BeagleBone Black Support Package from MATLAB is not real-time capable and thus, the RT-MaG Toolbox [103] has been used.

8.3.2. MATLAB/Simulink

The ADAS functionalities are implemented in MATLAB/Simulink [5] as shown in Figure 8.7. Inputs to the Simulink models are handled via UDP inputs, the outputs are taken from the BeagleBone Support package for Simulink. A Simulink library has been created in order to facilitate efficient testing and extensions. In the library, longitudinal and lateral controllers, different planning algorithms, computations for input or actuation, and also kinematic or bicycle vehicle models for simulation studies are included and can be used and extended.

Different Sampling Times

Due to computational costs, the planning level typically has a different sample rate than the controller, which is executed for smooth behavior with a sampling time of $T_{\text{ctrl}} = 10$ ms.

8. Testbed for Automated Driving using Small-Scale Vehicles

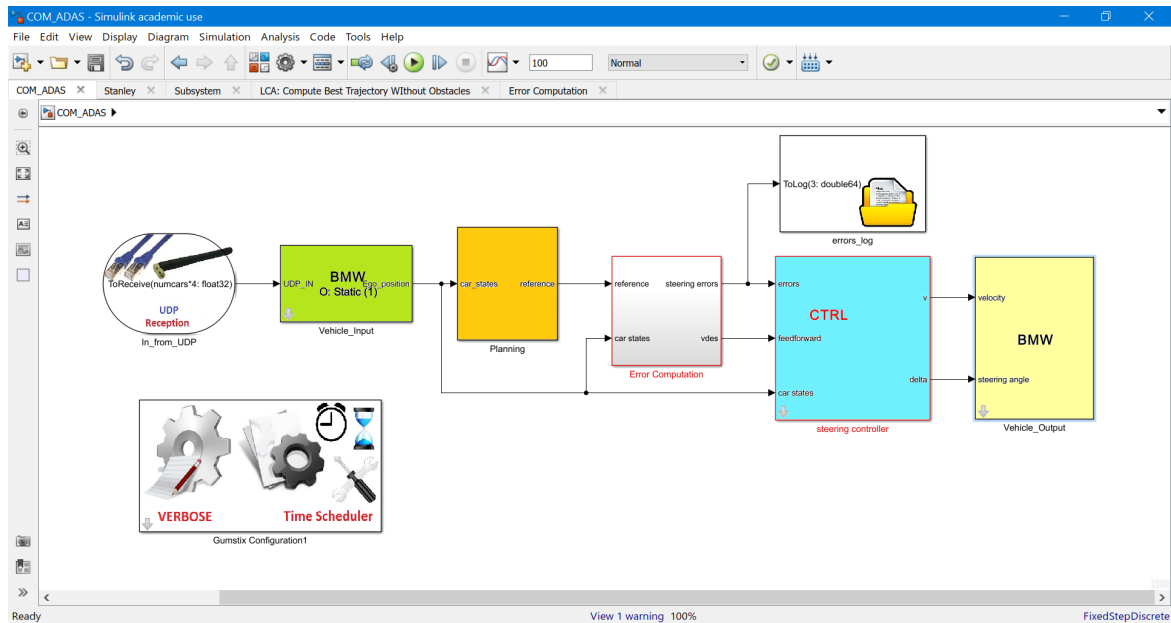


Figure 8.7.: Simulink block diagram for ADAS implementation with inputs (green) and outputs (yellow), the planning level (orange), error computation (light gray) and the tracking controllers (cyan). The RT-MaG toolbox configuration block is shown in the bottom left corner, and a data logging block (folder icon) is used for the inputs.

Since the position information arrives at a higher sampling time of $T_{\text{pos}} = 100$ ms, a position estimation algorithm is used. This estimation has been implemented based on the MATLAB example for position estimation using the Kalman Filter Block [3]. Moreover, different concepts can be used to handle the different sampling times, e.g., zero order hold, filtering techniques, interpolation, and so on. Note that the planning level has to generate the references so that different sampling times can be handled.

Planning

Different planning algorithms can be tested on the testbed. In order to test the MWC, the planning level as presented in Part I has been implemented with a sample time of $T_{\text{TP}} = 100$ ms. Other concepts can be investigated, such as model predictive control (MPC) for collision avoidance, turning on intersections or parking scenarios. Note, however, that the sampling time of the planning level has to be adapted accordingly.

Tracking Controller

For the lateral and longitudinal guidance, simple controllers have been used. As studied in [145], the performance of the controllers is not decisive for driving if the references are chosen appropriately. The sampling time $T_{\text{ctrl}} = 10$ ms is sufficient for smooth behavior in trajectory tracking scenarios, if the references are well-defined. The steering controller

computes the steering angle δ using

$$\delta = k_{\theta}e_{\theta} + \arctan\left(\frac{ke_y}{v}\right), \quad (8.1)$$

with the positive parameters k_{θ} for orientation error e_{θ} and k for lateral deviation e_y . This steering controller has been proposed for the autonomous car “Stanley”, as described in [163], and is hence called “Stanley controller”.

The longitudinal control input for velocity tracking of the reference v_{ref} is given by

$$v_c(t) = \int_0^t k_p(v_{\text{ref}}(\tau) - v(\tau))d\tau, \quad (8.2)$$

where v is the measured velocity of the vehicle, v_c is the input to the lookup table described in Section 8.1.4, and k_p is a parameter. The parameters of controllers have been tuned experimentally. Note that (8.2) is equivalent to first order actuator dynamics: taking the time derivative with $\dot{v}_{\text{ref}} = u$ and $\dot{v} = \dot{v}_c = a$, one arrives at

$$\dot{a} = \frac{1}{\tau}(\dot{v}_{\text{ref}} - \dot{v}) = \frac{1}{\tau}(u - a), \quad (8.3)$$

with $k_p = \frac{1}{\tau}$, and τ the time constant of the actuator dynamics.

In future projects, models of the small-scale vehicles can be identified in order to allow basic parameter tunings in simulations.

Data Logging

Data can be logged locally on the BeagleBone Black using the RT-MaG toolbox for MATLAB/Simulink. It is also possible to log the CPU execution times of the different tasks, which allows to investigate the real-time capability of the generated code on the BeagleBone Black. The data can then be easily transferred to the ADAS computer via an SSH connection.

Car-to-Car Communication

The RT-MaG Toolbox only allows to use one UDP input and one UDP output block, where the same IP address for both blocks has to be used. Hence, adaptations in the MATLAB code of the toolbox that generates the C code have been made. With this adapted version of the toolbox, several input blocks can be used and different IP addresses can be specified. Hence, it is possible to receive both the position data from the Position Tracking Computer and the acceleration from the preceding vehicle.

8.4. Conducting an Experiment

This section describes the steps necessary to conduct an experiment using the testbed.

First, the Position Tracking Computer has to be set up, which is done by a dedicated script that starts the AprilTag detection, sends UDP packets of the detected positions to the vehicles in a dedicated network, and activates the projectors to project the defined road on the floor. Different options are available, and the script automatically saves appropriate parameter files.

Then, the vehicles that are used have to be positioned such that the AprilTags can be detected, and the BeagleBone Boards are booted.

In MATLAB/Simulink on the ADAS computer, C code is generated and then stored on an FTP server. Then, by using an SSH connection to the BeagleBone that is in the same network, the code is downloaded and compiled on the BeagleBone Board. After successful compilation, the execution of the program is started via SSH connection. The vehicles execute the program until the pre-defined end time; during the experiment, each vehicle computes its references in a distributed manner. After the experiment, the logged data can be transferred to the ADAS computer.

9. Experimental Results

Basic ADAS such as LKA or ACC have been tested on the small-scale vehicles in order to evaluate the performance on the testbed. Then, the algorithms proposed in Part I and Part II have been implemented. The results of the experiments are given in this Chapter¹. In order to investigate the real-time capability of the trajectory planning level presented in Part I, the algorithms have been compiled on the BeagleBone Black Board of the small-scale vehicles, and the code is executed on the boards.

The chapter is structured as follows. In a first step, the controller parameters of the steering controller have been tuned in a lane keeping scenario with a constant velocity; the results are shown in Section 9.1. Then, an ACC has been added and the parameters of the velocity controller have been tuned. In addition, platooning experiments have been conducted with and without C2C communication, and are presented in Section 9.2. In a next step, a lane change maneuver has been tested, i.e., the planning level of the higher-level ADAS is used, where the lane change is triggered at fixed coordinates to test the lane change capability, which is discussed in Section 9.3. Then, experiments using different planning approaches are shown in Section 9.4, which are then compared to the trajectory planning level of Part I in Section 9.5. Finally, results of the ATFC of Part II are given in Section 9.6.

9.1. Lane Keeping with Constant Velocity

In a first test, the vehicles have to follow the virtually defined road shown in Figure 8.6 by controlling the steering angle. The longitudinal control input, i.e., the velocity v_c of the vehicles, has been set to a constant value. The reference in the center of a lane is computed based on the current position with a look-ahead distance similar to [114]. The goal of the vehicles is to track the right-hand lane, which has been accomplished by the Stanley controller (8.1) as shown in Figure 9.1. The cars' speed has been set to 0.5 m/s as depicted in Figure 9.2, while the trucks keep a velocity of 0.35 m/s. It can be seen in Figure 9.2a that the lookup table, which computes the PWM signal from the controlled velocity v_c , is accurate. Large peaks in velocity are due to faults in the position measurement: due to different camera calibrations, position deviations are encountered in the area where two cameras overlap, and packet dropouts may lead to wrong estimates of the velocities, which are computed from position data only. The controller's sample time is 10 ms, and the execution time is less than the sample time as shown in Figure 9.2b. The planning level in this scenario only computes the coordinates of the road at a certain look-ahead distance as reference points, i.e., no trajectories are generated nor obstacles predicted, and thus the planning level's execution time is lower than the one of the controller.

¹ Videos can be found at <https://www.tugraz.at/en/institutes/irt/automated-driving-lab/videos/>

9. Experimental Results

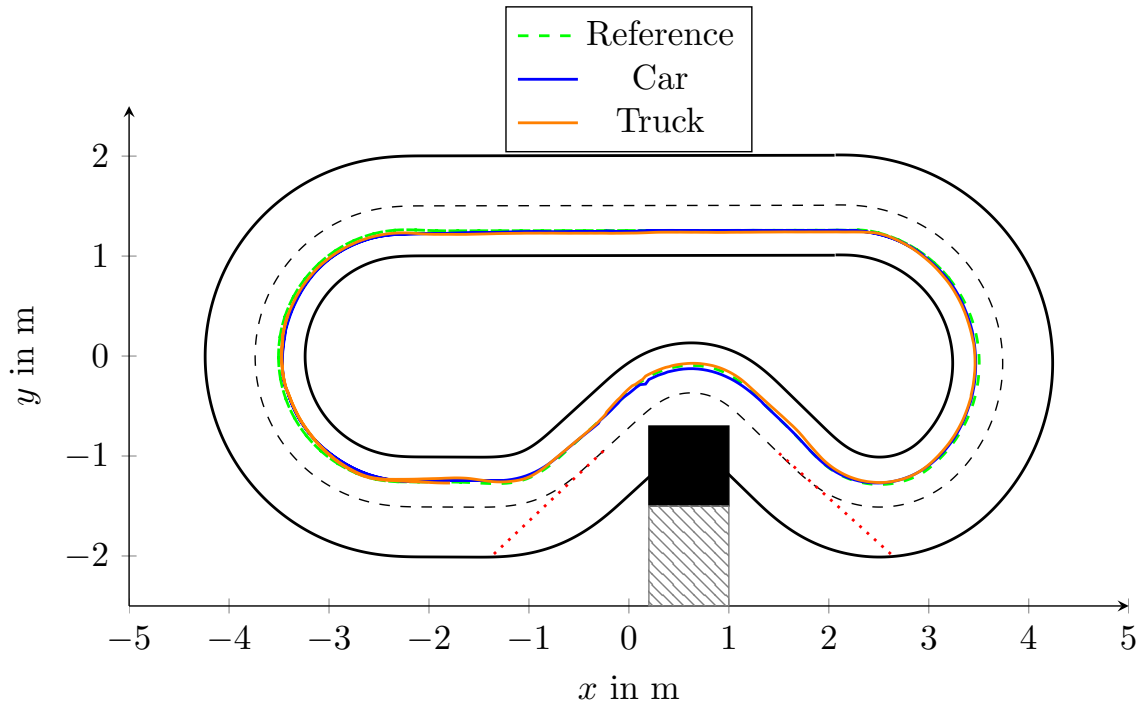


Figure 9.1.: Lane keeping results using a car and a truck. The vehicles drive with a constant velocity on the right-hand lane (clockwise).

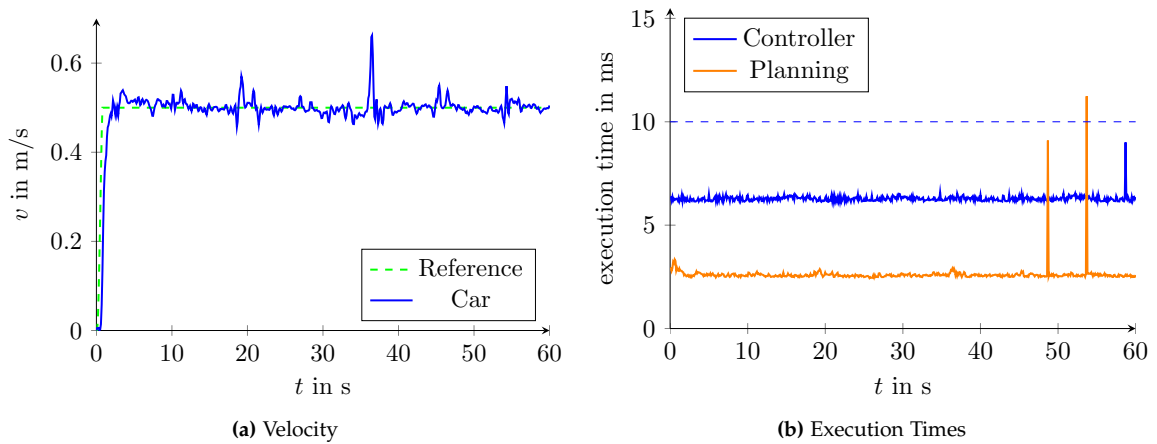


Figure 9.2.: Results of the lane keeping tests using a small-scale car. a) The constant velocity can be maintained. The peak arises due to overlapping areas of two webcams. b) The sampling times are given by $T_{TP} = 100$ ms and $T_{ctrl} = 10$ ms (indicated by the blue dashed line), hence all tasks can be executed in time.

9.2. Platooning

The following experiments focus on velocity control, where a standard adaptive cruise controller (without C2C communication) and a cooperative adaptive cruise controller (with C2C communication) have been implemented; for a description see, e.g., [108]. The experiments have been performed with three trucks, where the first one uses a velocity

controller to track a desired velocity profile. The other two vehicles have to follow the vehicle in front using appropriate velocity control techniques. In these experiments, no lateral trajectory planning has been performed. The steering controller has been used to keep the truck on the desired lane (using the lane keeping assist from Section 9.1, i.e., using controller (8.1)), while the velocity controllers have been tested and the parameters have been tuned.

In platooning scenarios using C2C communication, each truck sends its acceleration via UDP to the following truck, which uses this information as feed-forward. String stability using different spacings can thus be investigated experimentally as in real applications, see [128]. The constant time-headway spacing $e_{t,i}$ is given by

$$e_{t,i} = s_{i-1} - \Delta_i - t_h v_i, \quad (9.1)$$

where s_{i-1} is the relative distance between vehicle i and the preceding vehicle $i - 1$ along the road, Δ_i is the desired constant distance, t_h the time-headway and v_i the velocity of the vehicle i . Note that the longitudinal distance is denoted by s_i , whereas x_i is a global coordinate on the testbed.

Linear Control

The control input using a linear controller is computed by

$$v_c(t) = \int_0^t (\eta u_{i-1}(\tau) + k_p e_{t,i}(\tau) + k_d e_{v,i}(\tau)) d\tau, \quad (9.2)$$

with the tuning parameters k_p , k_d , the relative velocity $e_{v,i} = v_{i-1} - v_i$ and the spacing error $e_{t,i}$ defined in (9.1). The parameter $\eta = 1$, if feed-forward is used, and $\eta = 0$ otherwise.

With C2C communication ($\eta = 1$), collision-free platooning can be achieved even for a constant distance spacing with $t_h = 0$ s in (9.1) and a proper choice of the desired distance. In Figure 9.3, the results using C2C communication and a constant distance spacing with $\Delta_i = 0.65$ m are shown. The minimum distance on the testbed is $\Delta_{\min} = 0.45$ m, which corresponds to the length of the truck from center of the AprilTag to the rear end. Inter-vehicle distances below the minimum value result in collisions, which can be avoided in the given experiment. Note, however, that a communication delay or packet dropout can seriously degrade the performance of the platoon, which gives rise to networked control system analysis [76]. These dropouts can also be created artificially in Simulink, and collisions are observed if the communication fails.

In order to guarantee collision-free platooning without C2C communication, the constant time-headway spacing (9.1) has been used and the results with $\eta = 0$, $t_h = 1$ s and $\Delta_i = 0.65$ m are shown in Figure 9.4. Note that the reaction time of the vehicles without communication is larger, i.e., there is a delay in the velocities and thus, a larger distance must be maintained in order to avoid collisions.

Remark 23. *Large peaks in the spacings or in the velocities occur due to deficiencies in the position tracking of the vehicles. The velocities are derived from position measurements and hence, loss of position data also results in inaccurate velocities. Since the time intervals of the missing data,*

9. Experimental Results

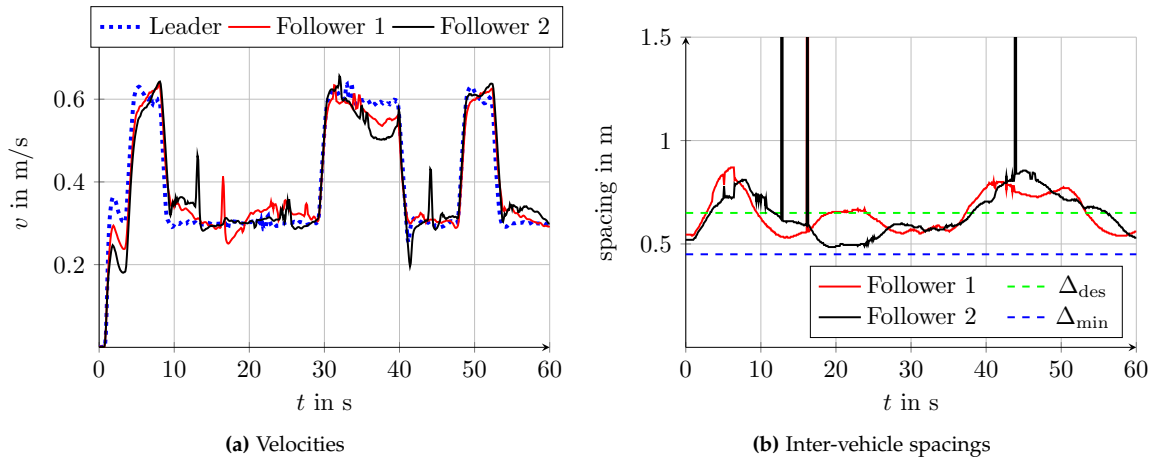


Figure 9.3.: Results of the platooning experiments using C2C communication and constant distance spacing

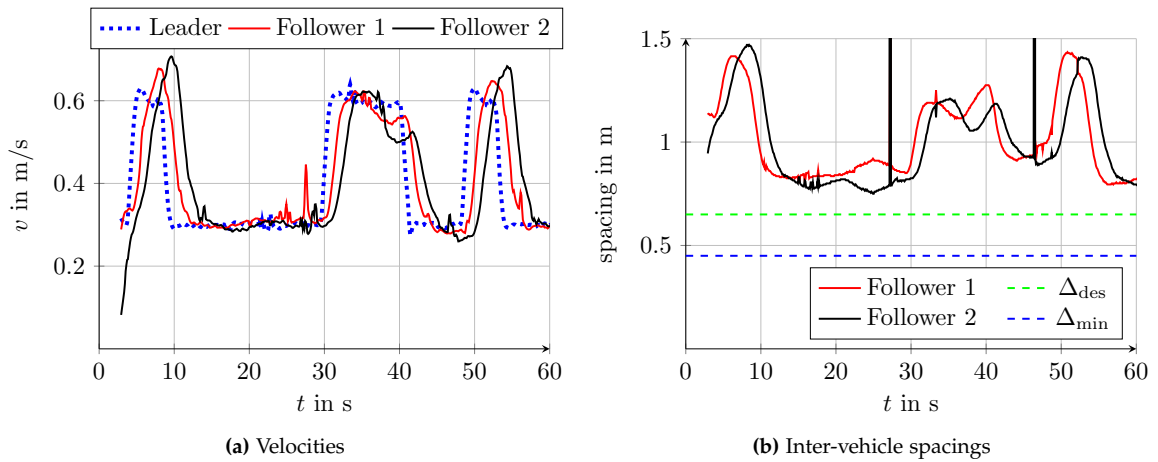


Figure 9.4.: Inter-vehicle spacings of the platooning tests without C2C communication. The constant time-headway spacing has to be used, which results in large inter-vehicle spacings.

however, are very short, the performance of the platoons is not significantly altered. These sensor faults will be handled in future work.

Sliding Mode Control

Platooning with sliding mode based controllers as discussed in Chapter 6 has been tested; the results using a constant distance spacing with $\Delta_i = 1$ m using a FOSMC with $k = 0.2$ are shown in Figure 9.5 for two following agents and a leader with constant velocity. Oscillations in the sliding variable occur due to unmodeled actuator dynamics and additional effects on the testbed, e.g., discretization effects, time-delays, and measurement noise. These oscillations can be observed in the velocities and the distances between the vehicles, and the amplitudes of the oscillations of the second follower are much larger than the amplitudes of the first vehicle's oscillations. Although no collisions occur for two following agents,

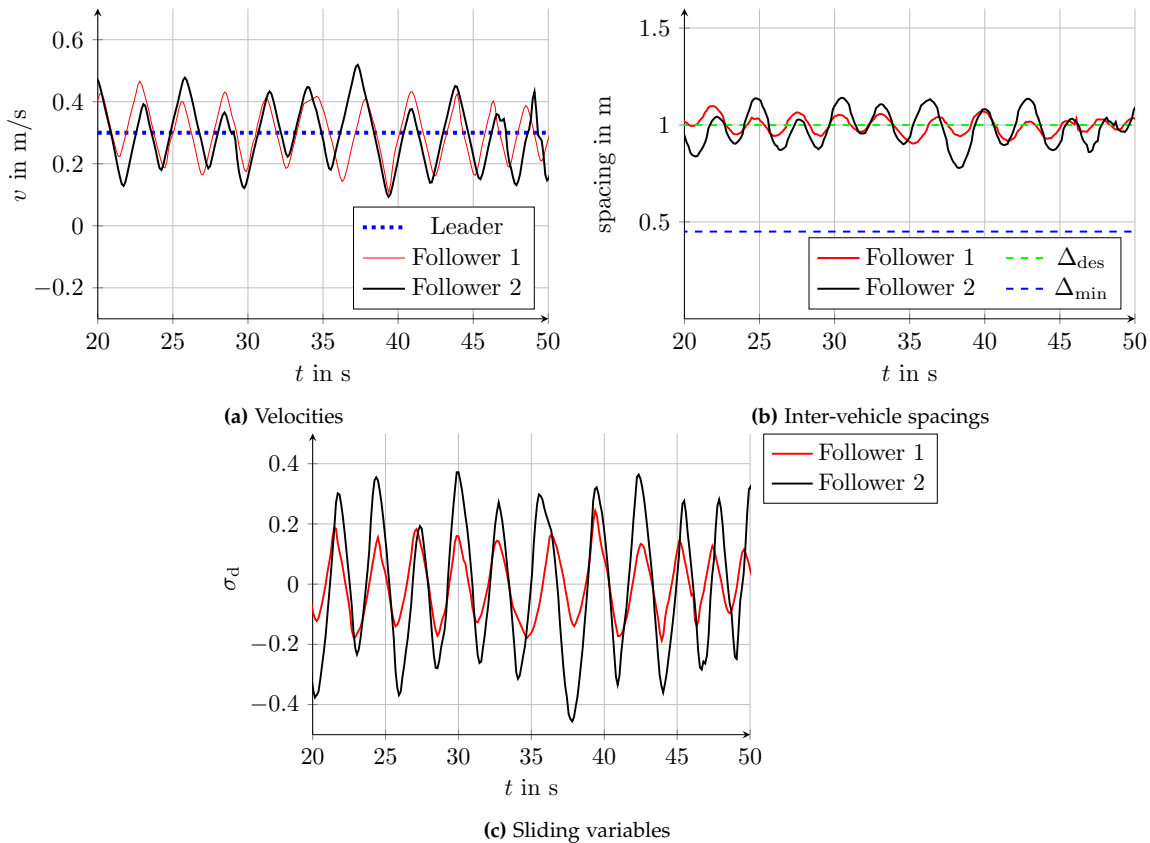


Figure 9.5.: Results of the platooning tests with a FOSMC and the constant distance spacing. Amplification of oscillations along the string cannot be avoided.

these amplifications will result in collisions for a larger number of vehicles, and the platoon is string unstable.

The results of the constant time-headway spacing with three following agents and a constant leader's velocity using a FOSMC are shown in Figure 9.6, and the results of the SOC with constant time-headway spacing and two followers are presented in Figure 9.7. It is worth mentioning that the choice of the time-headway has a large influence on the amplification of the oscillations, while the controller choice does not have a significant effect on the results; the SOC and the FOSMC both result in amplification of the oscillation amplitudes with the constant distance spacing, while the oscillations using the constant time-headway spacing with $t_h = 3$ s with both controllers are not amplified along the string. Note that the oscillations occur for all followers, but the amplitudes do not increase with increasing position in the platoon. Again, the oscillations can be observed in positions, velocities, and sliding variable, but the amplitudes are smaller compared to the constant distance spacing. Moreover, the SOC results in slightly lower amplitudes than the FOSMC with the same time-headway.

To summarize, the inter-vehicle spacing is larger than in the constant distance case, but oscillations are not amplified and string stability may be concluded when the constant time-headway spacing is used with a proper parameter choice, even if the sliding variable is not zero for all times. The oscillations in both FOSMC and SOC show that the vehicles are

9. Experimental Results

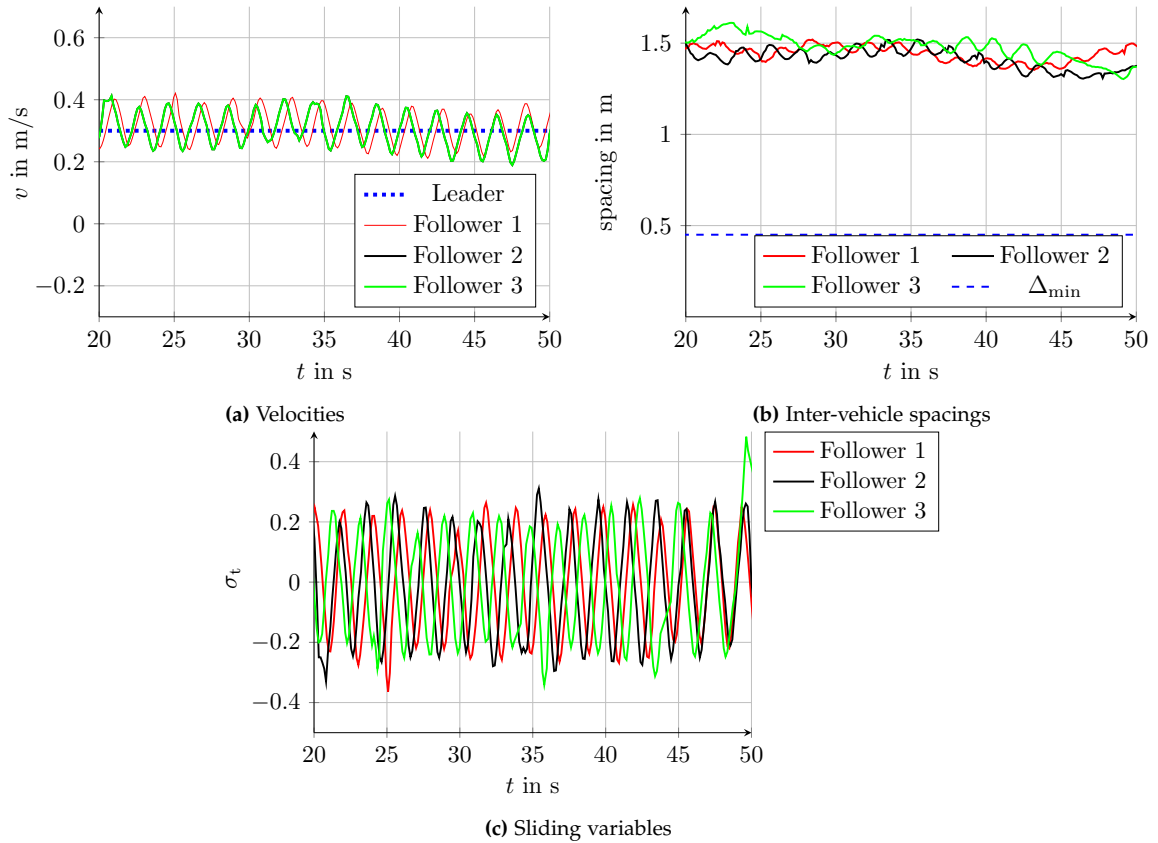


Figure 9.6.: Results of the platooning tests with three following agents using a FOSMC and the constant time-headway spacing with $t_h = 3$ s. Oscillations occur due to unmodeled dynamics, but are not amplified along the string.

affected by unmodeled dynamics, which will be the focus of future research. In addition, the choice of the time-headway for different controllers will be investigated.

Moreover, in future experiments, it is possible to emulate different dynamics of the trucks in the platooning scenario, e.g., different masses or road slopes, due to the easy access via Simulink. Additional dynamics can then be added virtually, e.g., based on the position of the vehicle, and the projectors can be used to visualize these effects. Hence, platooning applications with challenges arising on real trucks can be tested and demonstrated easily on the testbed. Moreover, additional sensors will be added to measure the velocity directly, improving the results of the velocity controllers.

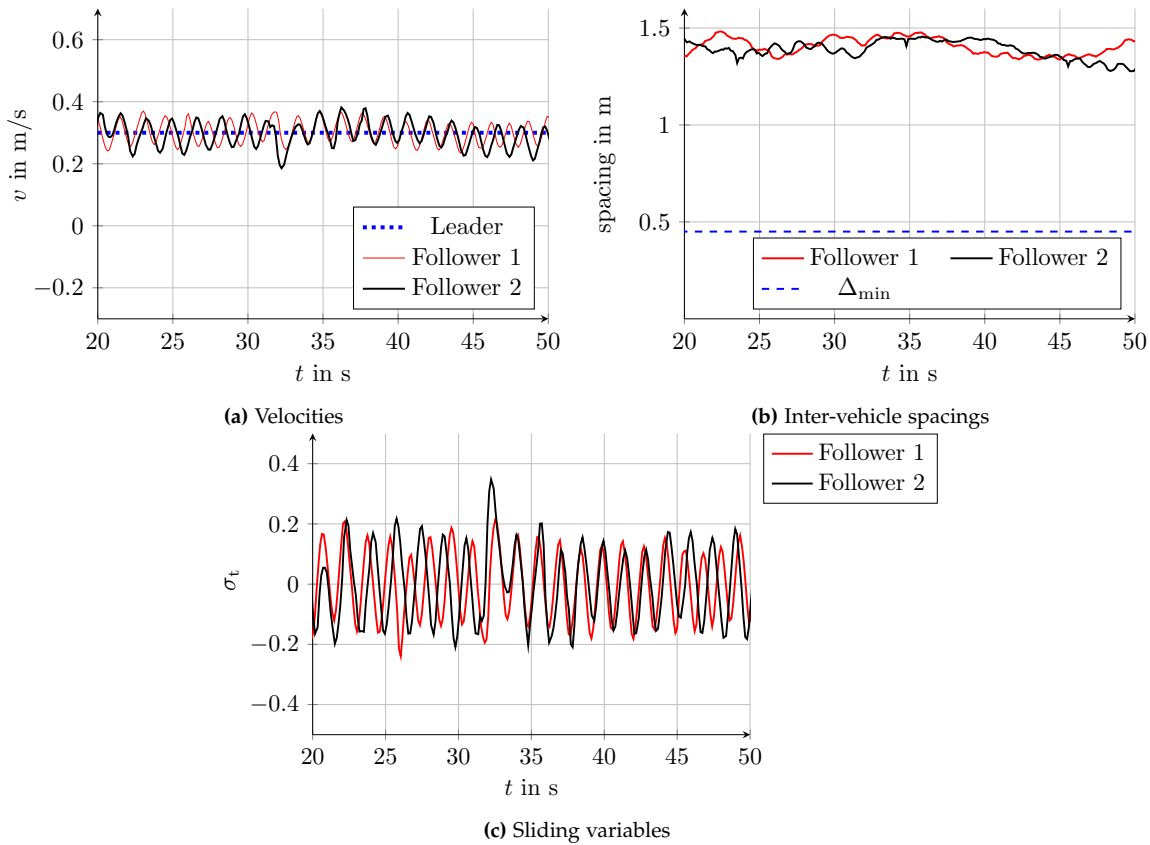


Figure 9.7.: Results of the platooning tests with the SOC and the constant time-headway spacing with $t_h = 3$ s. Oscillations are not amplified along the string.

9.3. Lane Changing

With well-tuned steering and velocity controllers, lane change maneuvers on highways have been tested in a next step. For this purpose, lane changes have been computed by the planning level at fixed positions without taking other traffic participants into account. The vehicles have to change lane in front and after the column as shown in Figure 9.8. Similar to merging areas on highways, slower velocities have to be maintained in the area around the column: the velocity at the column has been set to 0.3 m/s, while the desired velocity on the left lane has been chosen as 0.7 m/s.

Note that the reference for the steering controller is a point on a path, and the lateral deviation l is computed in the reference computation level as a function of way-length s . This yields better tracking results than using $l(t)$, which is used in the evaluation of the best trajectory. The planning level has a sampling time of $T_{TP} = 100$ ms, while the controller task is executed every $T_{ctrl} = 10$ ms. Between the execution time instants of the planning level, the reference path is interpolated to obtain the reference for the controller. The results are shown in Figures 9.8, 9.9. One can see that the desired lane changes can be executed with satisfactory performance as expected, even in road bends.

Moreover, the real-time capability of this reference generation has been investigated. The

9. Experimental Results

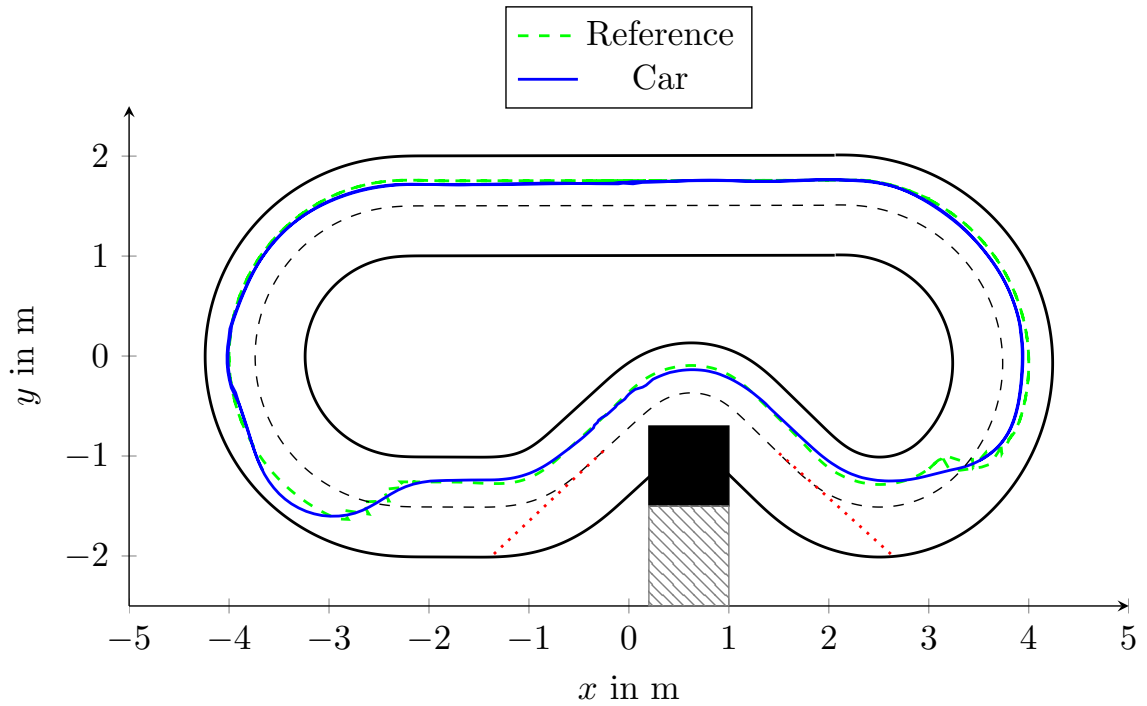


Figure 9.8.: Model car performing lane changes with different velocities. The lane change to the left lane (after the column) is difficult due to high acceleration and high curvature in this area. Hence, the error is larger, but the performance is still satisfactory.

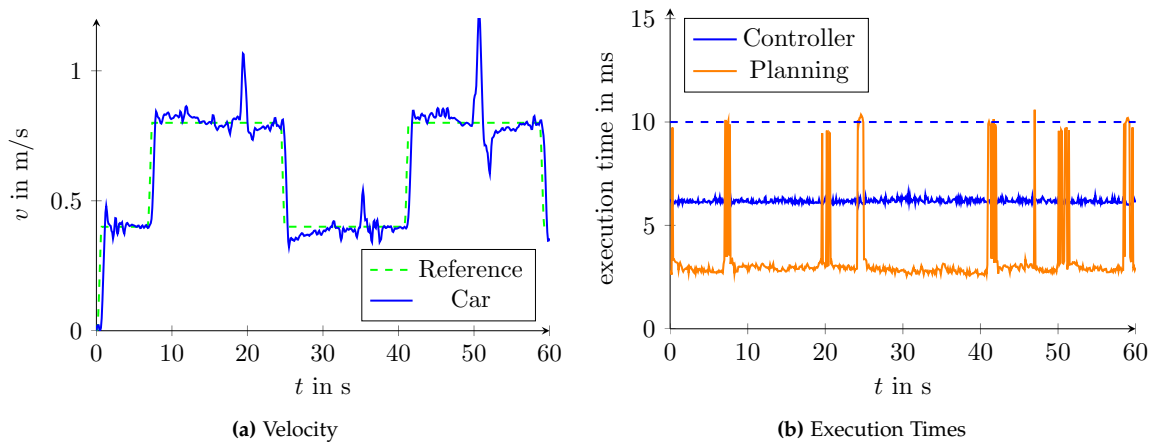


Figure 9.9.: Results of the lane changing tests. a) velocity profile for the small-scale car. b) Execution times of the planning level ($T_{TP} = 100$ ms not shown) and the controller ($T_{ctrl} = 10$ ms indicated by blue dashed line).

logged execution times are presented in Figure 9.9. One can see that in most time steps the planning task is even faster than the controller task. Since the planning level is sampled with $T_{TP} = 100$ ms, real-time capability can be concluded.

9.4. Alternative Reference Generation Techniques

Instead of the trajectory planning level presented in Part I, different alternative reference generation techniques may be used. It is, however, important to investigate the execution time of each technique, since real-time capability has to be ensured. Two alternative reference generation techniques have been tested on the testbed as described below.

9.4.1. Fifth order polynomials for Intersection Maneuvers

Fifth order polynomials as described in [107] have been computed for intersection maneuvers, e.g., turning or changing lanes. In contrast to the trajectory generation for highways described in Part I, the polynomials are functions of way-length s instead of time t , and include the orientation angle θ . Hence, a variety of trajectories can be generated, e.g., turning maneuvers at intersections or lane changes. Note that in contrast to the approach of Part I, optimization is necessary to compute the coefficients of the polynomials. Hence, in a first test, the polynomials were computed offline and only tracking was performed online as shown in Figure 9.10a for turning maneuvers at intersections. In the second test, the trajectories were re-computed once starting at $(x, y) = (-2, -0.95)$ online, i.e., while the vehicle was driving. The effects of this online computations have been investigated and the tracking results are shown in Figure 9.10b, where the controller is not able to track the reference around $x = -1.5$ m, $y = -0.5$ m, since the planning task cannot be executed in real-time with $T_{TP} = 100$ ms.

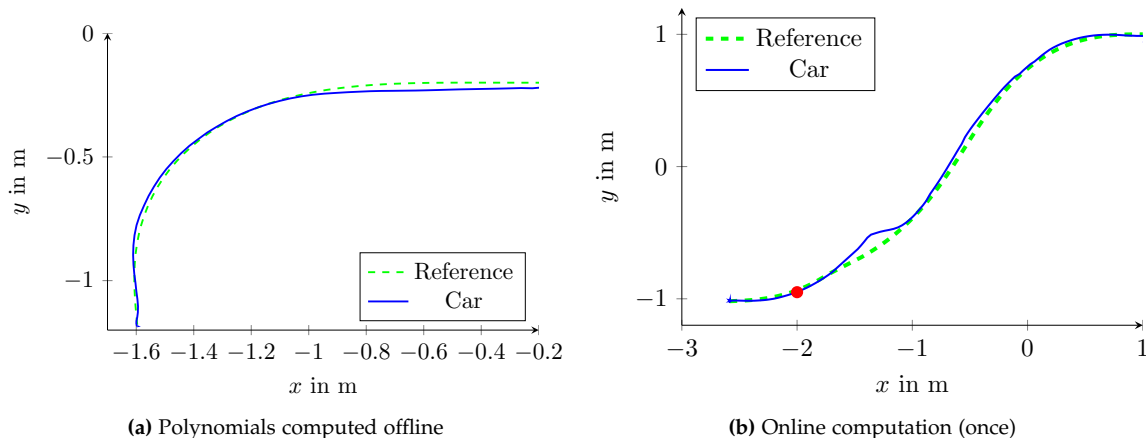


Figure 9.10.: Results of the different reference generation and tracking experiments using fifth order polynomials as function of waylength. a) turning maneuver with offline computation. b) lane change maneuver with one online computation starting at the coordinate indicated by the red dot. Due to high computational costs, the controller cannot track the reference.

9.4.2. Model Predictive Control

A model predictive planning for highway driving that is based on [115] has been implemented on the trucks. Simple models, specifically, double integrators for both longitudinal

9. Experimental Results

and lateral dynamics, have been used to generate collision-free paths on two-lane highways while considering input and environmental constraints. The model predictive planning has been tested with a sampling time $T_{MPC} = 200$ ms, otherwise real-time capability on the BeagleBone Black could not be achieved. The results of this planning level are shown in Figure 9.12. Three vehicles have been used: one truck (black) is standing on the right lane, while the blue vehicle executes the model predictive planning. The red obstacle tracks a constant velocity on the left lane and thus overtakes the black obstacle. The blue vehicle has to determine when to overtake the black obstacle: depending on the velocity of the red and blue vehicle, overtaking can either be performed in front of or behind the red vehicle. In the given case in Figure 9.11, the blue vehicle overtakes in front of the red vehicle. The velocities of the vehicles are shown in Figure 9.12a. Again, peaks in the velocities of both red and blue vehicle can be observed, which arise from the overlapping webcam areas as mentioned in Remark 23.

The execution times in Figure 9.12b show that the model predictive planning takes approximately 100 ms for the computation of the references. Note, however, that a sampling time of 100 ms does not yield satisfactory results, since the CPU load is too high in this case.

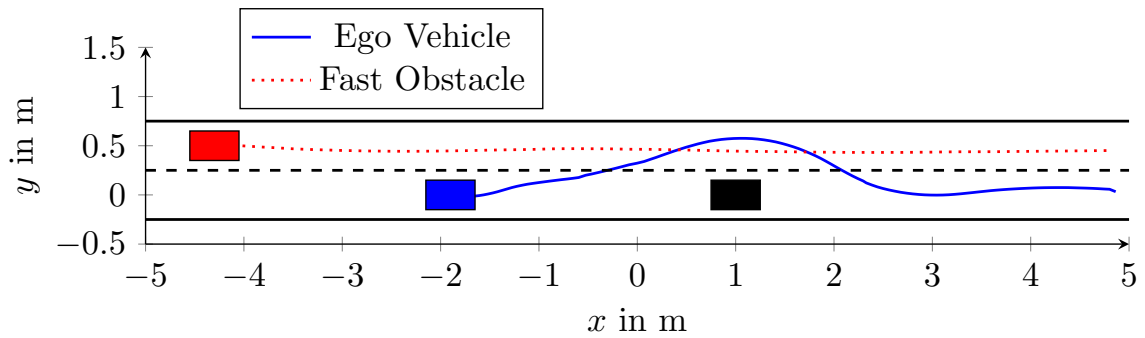


Figure 9.11.: Results of an overtaking maneuver with model predictive reference computation

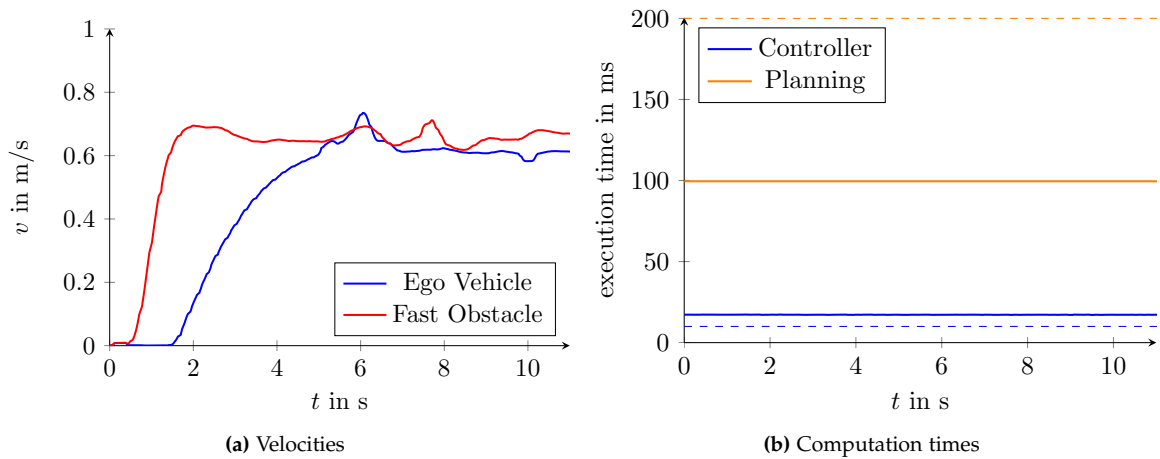


Figure 9.12.: Results using the model predictive reference computation with fast rear obstacle. a) velocities of the vehicles. b) sampling time of the planning level ($T_{TP} = 200$ ms indicated by orange dashed line) and the controller ($T_{ctl} = 10$ ms indicated by blue dashed line). Note that the controller violates its execution time.

Remark 24. Note that contrary to the classical MPC, the predicted states are used instead of the actual states of the vehicle, i.e., the MPC is used as a pure feed-forward in the planning level, and a low-level controller is used separately for tracking. Hence, if large errors occur in the tracking level due to disturbances, no appropriate reference can be computed.

9.5. Trajectory Planning Level

The trajectory planning level proposed in Part I has been tested on the trucks with a sampling time of $T_{TP} = 100$ ms. First, a collision avoidance scenario has been tested, where a vehicle has to avoid two other vehicles, which are at standstill. The tests have been performed with ten generated trajectories, where two lateral and five longitudinal endpoints have been considered. In contrast to the simulation results in Chapter 3, however, only one planning cycle has been used. For the evaluation and obstacle prediction, 20 samples are stored, and the parameters of the ellipses have been adapted to the small-scale testbed. The results of the trajectory planning level with obstacle prediction and a prediction observer as presented for the MWC are shown in Figure 9.13.

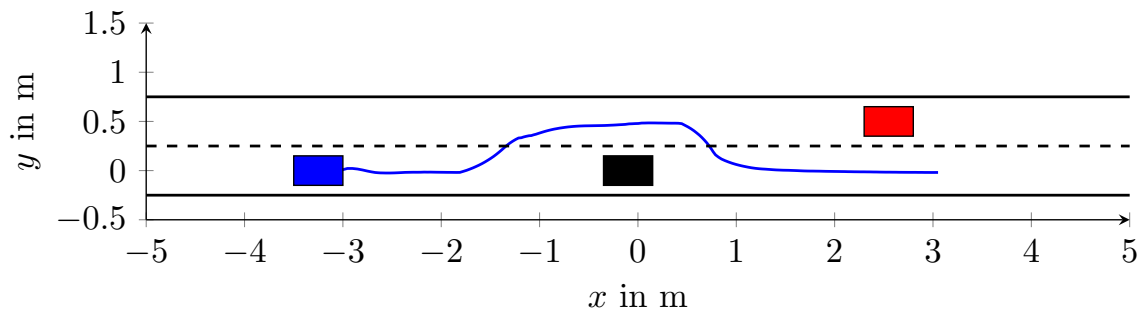


Figure 9.13.: Results of the trajectory planning level in a collision avoidance scenario with two obstacles at standstill

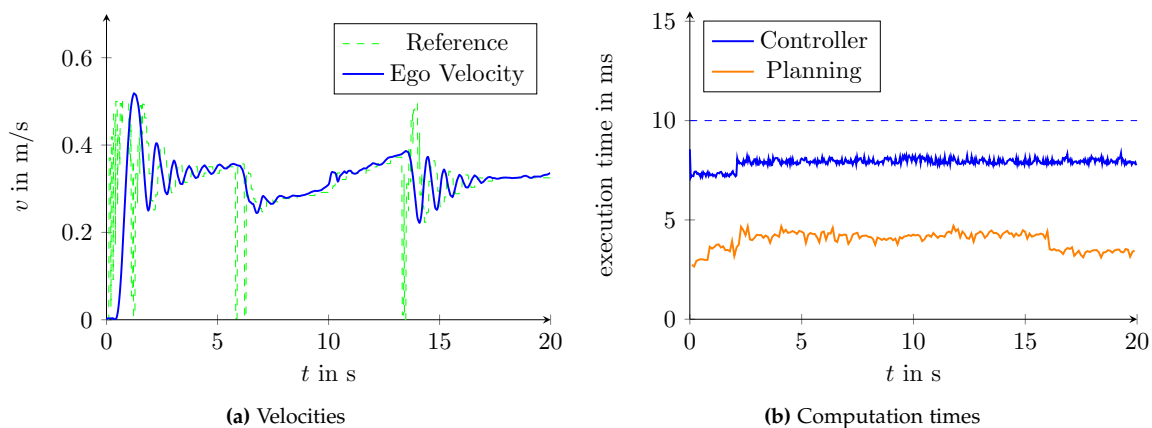


Figure 9.14.: Results of the trajectory planning level in a collision avoidance scenario. The actual states are used in the planning level and oscillations in the velocities can be observed. b) execution times of the planning level (sampling time $T_{TP} = 100$ ms not shown) and the controller ($T_{ctrl} = 10$ ms indicated by blue dashed line). The execution times are low compared to the MPC.

9. Experimental Results

Note that compared to the path computed by the MPC reference generation, the trajectory planning level computes a more distinct lane change and keeps a larger distance to the obstacles. The computation times in Figure 9.14b are below the defined sampling time of $T_{TP} = 100$ ms, and real-time capability can be achieved.

In a second experiment, the ego vehicle passes a slow obstacle in front, while a fast obstacle is approaching from behind. The resulting paths are shown in Figure 9.15, with the corresponding velocities in Figure 9.16a. As depicted in Figure 9.16b, the tasks can be computed in real-time, since the planning takes far less time than the sampling time $T_{TP} = 100$ ms and on average needs even less time than the controller.

The costs of the trajectories as described in Section 2.2.4 are shown in Figure 9.17. On the right lane, large velocities result in early collisions with the black obstacle, as indicated by the obstacle costs in Figure 9.17a. On the left lane in Figure 9.17b, low velocities result in collisions with the rear vehicle and hence, the ego vehicle speeds up at $t \approx 9$ s.

Compared to the simulations with IPG CarMaker in Section 3.2, the velocity results in Figures 9.14a and 9.16a are worse. Note that the planning level computes collision-free paths and appropriate references, but the tracking level is not capable of following the

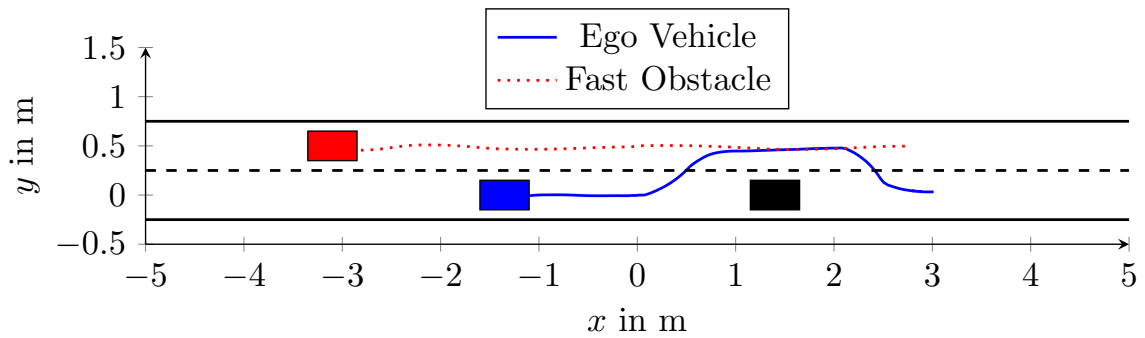


Figure 9.15.: Results of the trajectory planning level using the small-scale vehicles in overtaking maneuvers

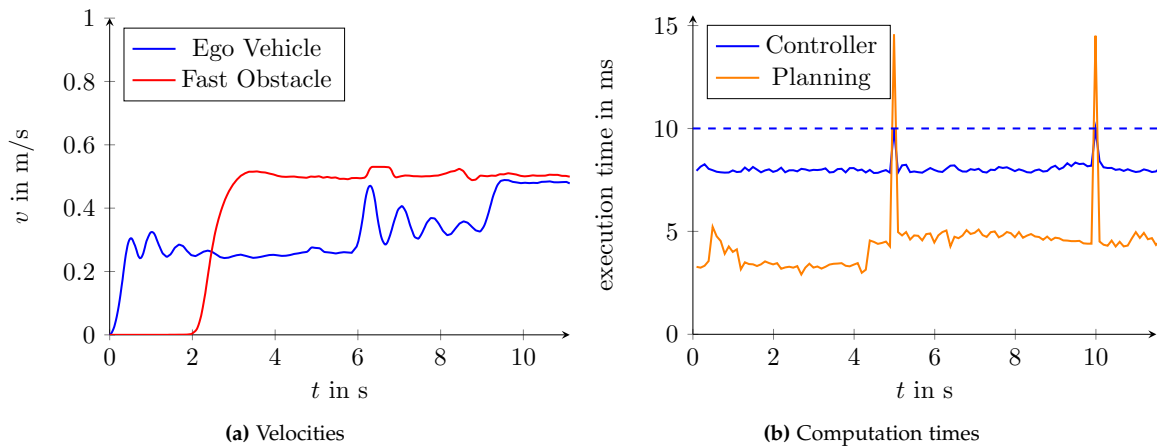
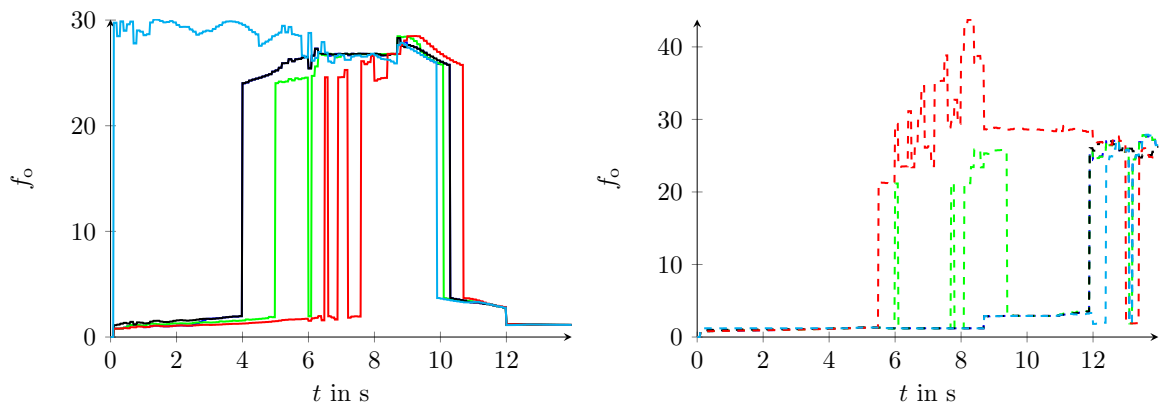


Figure 9.16.: Results of the trajectory planning level in overtaking maneuvers using small-scale vehicles. a) velocities of the vehicles. b) execution times of the planning level (sampling time $T_{TP} = 100$ ms not shown) and the controller ($T_{ctrl} = 10$ ms indicated by blue dashed line).

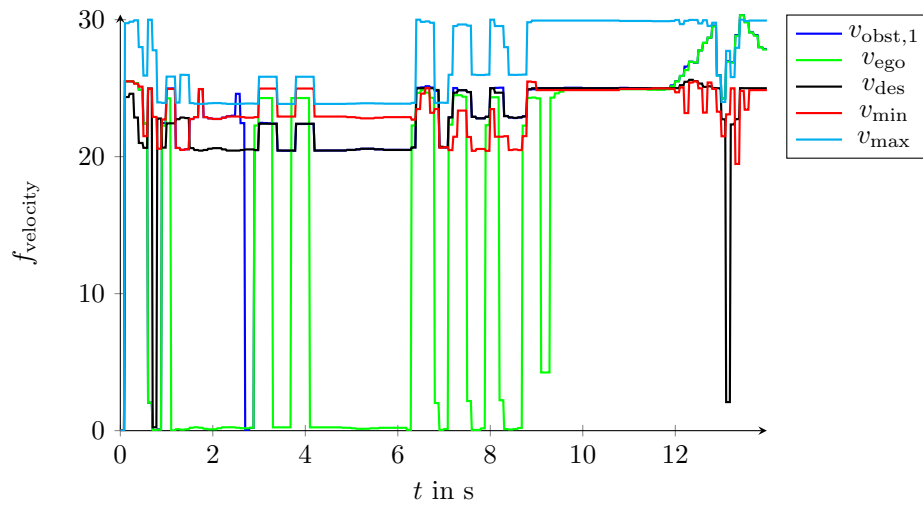
velocity correctly. On the one hand, the actuators of the small scale vehicles are slower than the actuation of the simulated cars, and on the other hand the velocities are only derived from position measurements. Hence, errors in the velocity computations arise. These errors, however, are used in the reference generation: the oscillations in the reference signal arise from the fact that the trajectory with the current ego velocity is chosen as best trajectory as in Figure 9.17d, but this ego velocity is not constant and delayed due to the actuator dynamics, which results in the high variation of the ego velocity costs in Figure 9.17c.

These oscillations can be attenuated by either handling the actuator dynamics of the ego vehicle in the reference generation explicitly, or by changing the weights of the velocity component of the overall cost function. Note that the velocity using the MPC in Figure 9.12 is smoother since it does not consider actual states of the vehicle; if the TP does not consider the actual velocity, similar results can be obtained. However, considering actual states of the vehicle, i.e., to have a feedback as opposed to pure feed-forward in order to reject disturbances, is important and hence, these oscillations have been tolerated in this work. Their mitigation will be treated in future work.

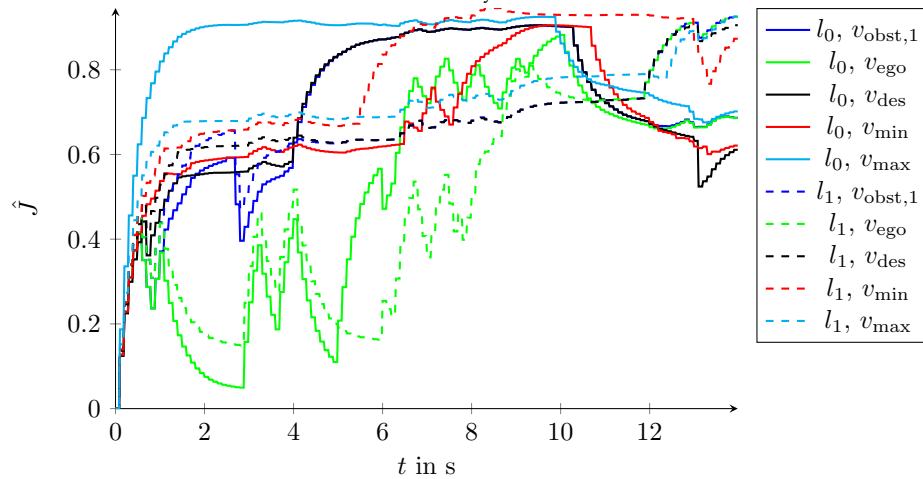
9. Experimental Results



(a) Obstacle costs of trajectories ending on the right lane (l_0) (b) Obstacle costs of trajectories ending on the left lane (l_1)



(c) Velocity costs



(d) Filtered overall costs

Figure 9.17.: Cost components of the trajectory planning level

9.6. Adaptive Time-Headway Formation Controller

In a last step, the ATFC of Chapter 6 has been tested. Note that actuator dynamics as discussed in Section 7.4 are present, and velocity measurements are subject to noise and sensor faults. Hence, maintaining sliding cannot be guaranteed; however, the controller usually keeps the sliding variable within a band; when this is the case, the time-headway is decreased, and the time-headway is reset if the sliding variable leaves this band. Similar to Chapter 7, the FOSMC and the SOC have been used for a platooning scenario with non-zero initial errors. The controller parameter has been chosen as $k = 0.2$, otherwise too large accelerations are applied if the position information is lost. Consequently, the leader's acceleration is also bounded; however, due to uncertainties in the estimation of the velocities, larger accelerations might occur. These act as disturbance, which deflects the sliding variable from the boundary layer.

FOSMC

Figure 9.18 shows the results using the ATFC with a FOSMC. Since the sliding variable in Figure 9.18a exceeds a specified tolerance of the boundary layer several times, the time-headway in Figure 9.18b is reset several times.

The velocity and the computed acceleration of the vehicle are shown in Figure 9.19. Due to measurement noise and packet dropouts, the velocities of both agents vary strongly and hence, the system has to deal with large perturbations and uncertainties. However, the agent is capable of following the leader while avoiding collisions, which can be concluded from the phase plane of the errors in Figure 9.18c. The ATFC is real-time capable with a sampling time of $T_{\text{ctrl}} = 10 \text{ ms}$ as depicted in Figure 9.18d.

9. Experimental Results

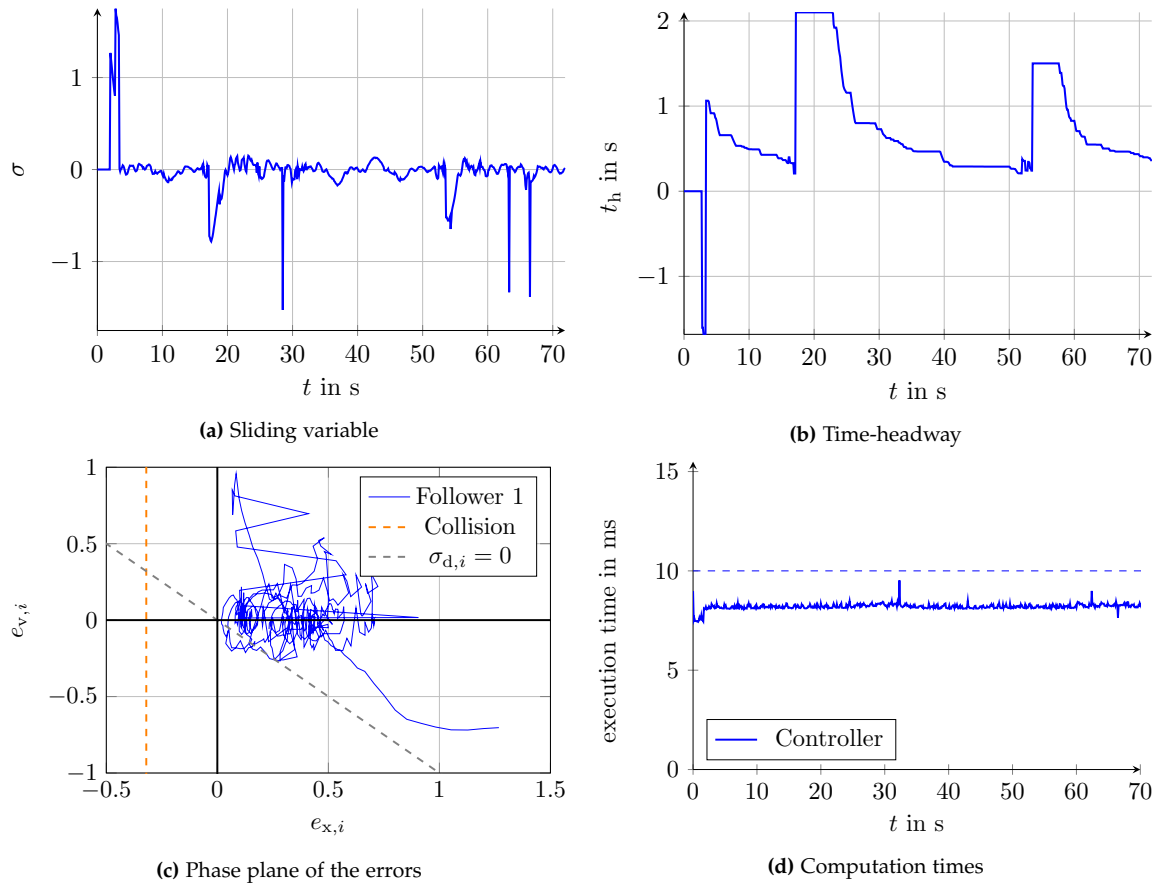


Figure 9.18.: Results of one agent approaching the leader using the ATFC with a FOSMC

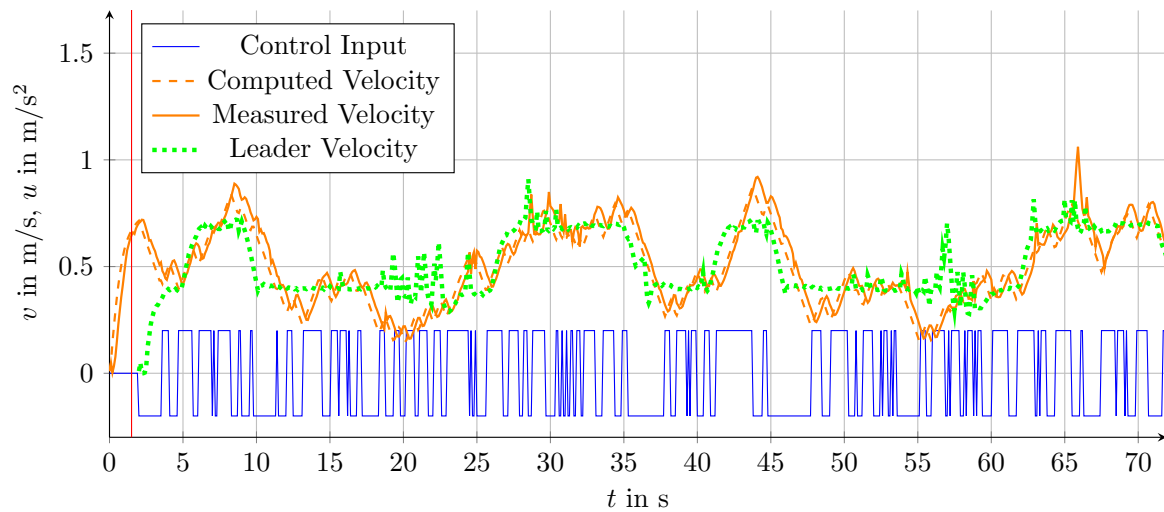


Figure 9.19.: Computed velocity and control input in the experiment of the ATFC with a FOSMC

SOC

The suboptimal controller has been used in order to account for the actuator dynamics as discussed in Chapter 7. The adaptive parameter of the SOC has been set to a constant value $\alpha_i = 1$; otherwise large accelerations are applied and collisions occur. The velocities and computed accelerations are shown in Figure 9.20, which are very similar to the experiment with the FOSMC in Figure 9.19.

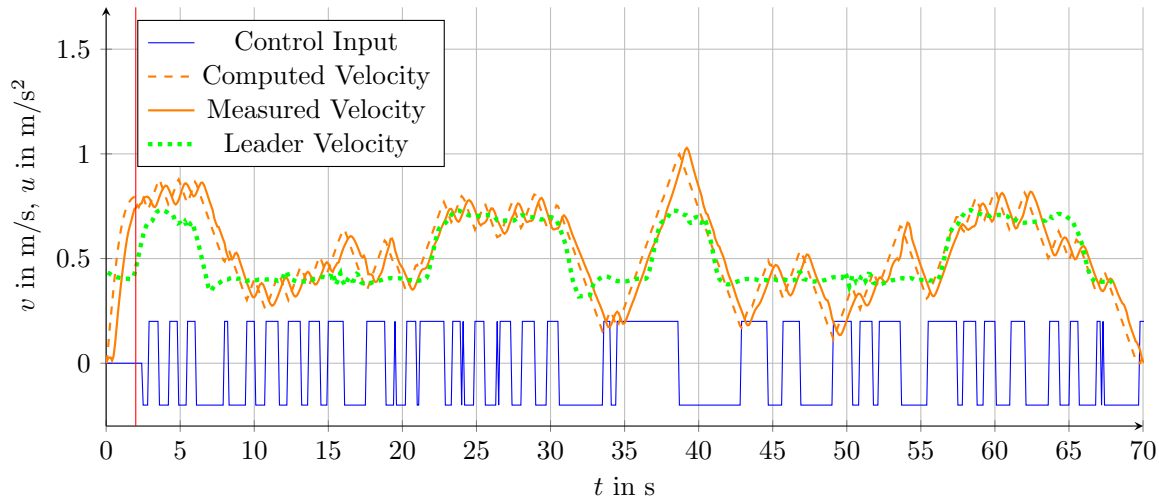


Figure 9.20.: Computed velocity and accelerations of the ATFC using the SOC

The sliding variable and the time-headway using the SOC are shown in Figure 9.21. Small oscillations can be seen, and the sliding variable again leaves the boundary layer several times due to several reasons: If the leader's acceleration is larger than the parameter k , e.g., around $t = 32$ s, then the sliding variable is deflected. Moreover, at $t \approx 48$ s, the sliding variable leaves the band and the time-headway is reset due to a packet dropout and that led to wrong velocity estimates. The phase plane of the errors depicted in Figure 9.21c shows that the errors exhibit a periodic motion. Thus, additional unmodeled dynamics are present on the testbed that have not been considered so far. Future work will deal with this unmodeled dynamics and uncertainties. Finally, the times of the execution tasks are depicted in Figure 9.21d, and it can be concluded that the ATFC can be computed at a sampling time of $T_{\text{ctrl}} = 10$ ms. At two time instances, however, the computation time is higher than the nominal value; the cause has to be investigated further on.

In presence of unmodeled dynamics, a small time-headway results in string unstable behavior. Hence, instead of reducing the time-headway to zero, it can be reduced to a certain critical value, which will be investigated in future work.

9. Experimental Results

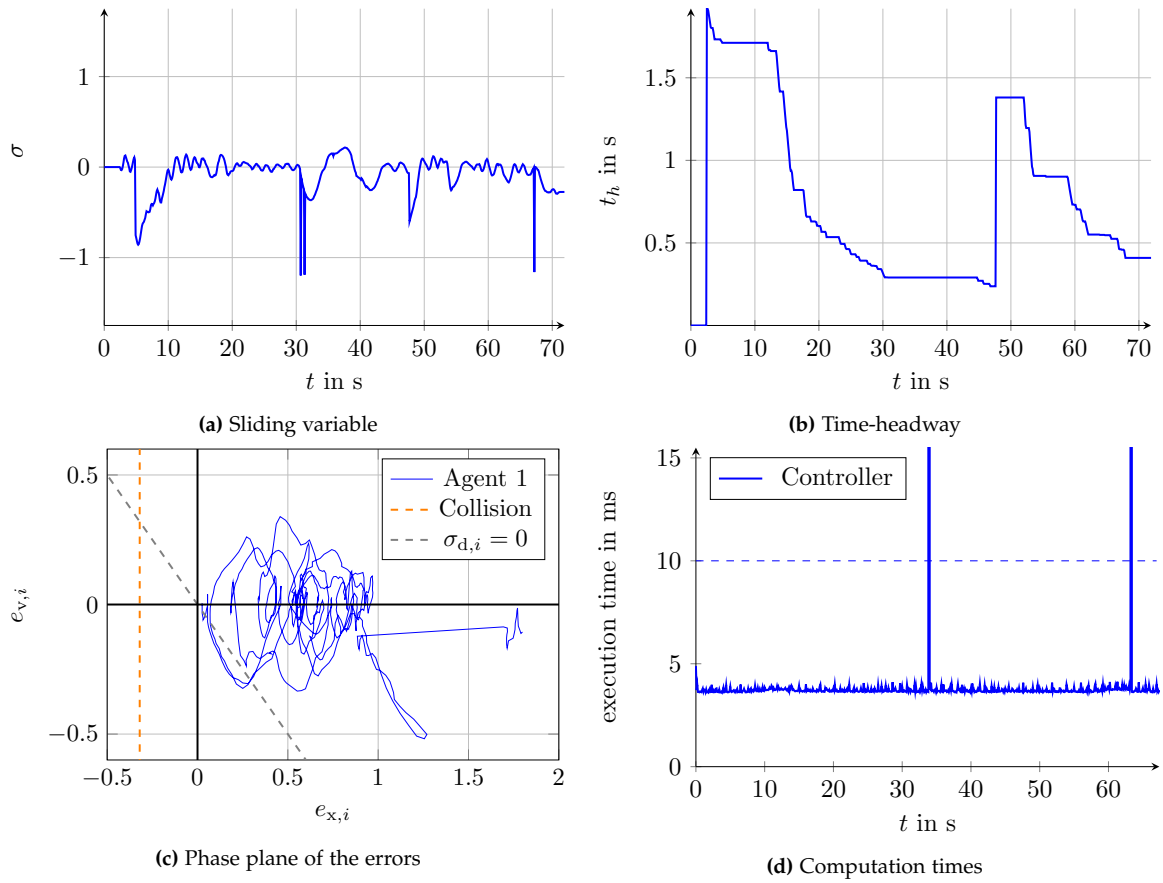


Figure 9.21.: Results of the ATFC using the SOC

Concluding Remarks to Chapter 9

Note that the actuators of the small-scale vehicles have not been designed for high-precision driving and are hence not suitable for highly dynamic driving. Although very little measurements are currently used, the different ADAS functionalities have been successfully implemented and tested on the automated driving lab. The trajectory planning level described in Chapter 2 and the cooperative merging assist in Chapter 7 are real-time capable when implemented on the BeagleBone Black Board. Main ADAS functionalities can be tested, such as platooning or overtaking maneuvers. Other reference generation methods have been implemented and analyzed with respect to their execution times. The proposed trajectory planning algorithm performs much better than alternative reference generation techniques with respect to handling, tuning and extension of the planning level, computational effort and performance in obstacle avoidance scenarios.

In future work, the velocity measurements and velocity control of vehicles will be improved. For this purpose, the vehicles will be equipped with additional sensors, sensor fusion algorithms will be investigated, and model identification will be performed. In addition, communication between the cars will allow for additional cooperative driving maneuvers. The projectors can also be used to project virtual cars on the road, which makes it possible to conduct experiments with more than six vehicles at once. The actuator dynamics have a significant impact on the results of the longitudinal controllers and need to be investigated separately.

10. Conclusion and Outlook

In Part I, a reference generation technique for automated highway driving has been discussed. For this purpose, a set of trajectories is generated and then evaluated based on a cost function, considering the prediction of other traffic participants. The tuning of the cost function seems to be tedious, but the same is true for other trajectory generation methods such as model predictive control, where the parameters highly influence the results. The lower computational complexity of the proposed method was decisive for the implementation in the project. Note that the separation of the planning and tracking level for standard maneuvers is sufficient, but highly dynamic maneuvers cannot be handled if simple models are used in the planning level. However, these simple models have been found to be sufficient for overtaking and lane keeping maneuvers. The trajectory planning has been tested in simulation, which have shown satisfactory behavior in several maneuvers with low traffic volume. In scenarios with dense traffic, however, standard ADAS encounter a major problem: the desired distance to a target vehicle cannot be maintained due to human drivers cutting in. Moreover, the obstacle prediction is very challenging. Hence, cooperative driving has been investigated in Part II.

Part II has focused on longitudinal control for platooning. In this context, string stability is an important requirement i.e., position errors and accelerations must not be amplified along a string of vehicles. First, linear controllers and the concept of string stability have been discussed. While linear controllers are not capable of rejecting the disturbance that arises from the leader's acceleration, robust control techniques such as sliding mode based controllers can result in string stable performance without communication for zero initial spacing errors. In the case of non-zero initial spacing errors, communication can be used to ensure that the vehicles reach the formation. However, since communication channels are typically subject to network imperfections such as packet dropouts or time-delays, platooning algorithms without information have been the focus of this work. In this work, a platooning algorithm called "Adaptive Time-Headway Formation Control" (ATFC) has been proposed for non-zero initial spacing errors. It permits reaching the desired position collision-free in a robust manner, i.e., regardless of the leader's acceleration. In a next step, merging maneuvers for lane reduction scenarios have been investigated. An algorithm to choose the correct merging sequence in a distributed manner has been presented, and the string stable ATFC has been extended for these maneuvers. Due to the robust performance of sliding mode based controllers, disturbances that arise due to different curvatures can be rejected as well. Finally, the effect of additional actuator dynamics is briefly discussed.

A testbed using small-scale vehicles has been presented in Part III, and the algorithms of Part I and II have been implemented on embedded units. The vehicles can perform basic maneuvers such as vehicle-following, overtaking, parking, and turning. Moreover, the trajectory planning algorithm of Part I has been compared to a polynomial and a model predictive approach. It has been shown that the predictive approach takes more

10. Conclusion and Outlook

computational power and its performance is not satisfactory. Sliding mode based controllers have been tested on the vehicles as well. Due to unmodeled dynamics, oscillations occur and sliding cannot be guaranteed. However, by appropriate choice of the time-headway, amplification of these disturbances may be avoided. Note that the sensor measurements and actuator dynamics render the vehicle's performance imprecise, but the planning levels can deal with these uncertainties.

Future Work

In a next step, the trajectory planning level of Part I will be implemented in a demonstrator vehicle. Moreover, the Motorway Chauffeur will be extended to include an emergency stop, i.e., a lane change to the emergency lane with a comfortable but fast slow-down. Moreover, the extension of this planning level to rural or urban scenarios will be studied.

The cooperative merging of Part II will be extended for roundabouts. Moreover, upper limits on the velocities have to be added for highway merging, and actuator dynamics need to be considered. Since first results show that a proper choice of the time-headway is very important, the ATFC will be modified so that this minimum time-headway is reached instead of a constant distance.

For the testbed in Part III, virtual traffic participants will be added that allow to simulate dense traffic for merging scenarios. In addition, road slopes or additional weight can be emulated via Simulink models on the testbed, which allows to test additional platooning challenges. Moreover, sensors will be added to improve the measurements, and sensor fusion algorithms will be tested. Finally, the higher order dynamics of the vehicles need to be taken into account, i.e., the tracking level will be refined.

Appendix

Appendix A.

Consensus Reaching with Sliding Mode Control

Consider the following consensus control problem: the leader is given by the double integrator dynamics

$$\begin{aligned}\dot{x}_0 &= v_0, \\ \dot{v}_0 &= u_r,\end{aligned}\tag{A.1}$$

with position x_0 , velocity v_0 and control input u_r , bounded by $|u_r| \leq u_{r,\max}$. The N leader-following agents are modeled by

$$\begin{aligned}\dot{x}_i &= v_i, \\ \dot{v}_i &= u_i,\end{aligned}\tag{A.2}$$

with position x_i , velocity v_i and control input u_i for agent $i = 1, \dots, N$. Let

$$\begin{aligned}e_{x,i0} &= x_i - x_0, \\ e_{v,i0} &= v_i - v_0,\end{aligned}\tag{A.3}$$

denote the position error and velocity error, respectively, between an agent i and the leader, which is agent 0. Note that the error is defined with respect to the leader, and other agents are not necessarily considered. The goal is then to reach consensus with respect to the leader's position, i.e.,

$$\lim_{t \rightarrow \infty} e_{x,i0}(t) = 0 \quad \forall i = 1, \dots, N.\tag{A.4}$$

For this purpose, a local control input, also called consensus protocol, u_i has to be designed based on the information of the neighboring agent $i - 1$ to achieve the asymptotic behavior (A.4) with respect to the leader.

For simplicity, consider the unidirectional platooning example, where each agent is only measuring the states of its preceding vehicle. Let

$$\begin{aligned}e_{x,i} &= x_{i-1} - x_i, \\ e_{v,i} &= v_{i-1} - v_i,\end{aligned}\tag{A.5}$$

denote the position error and velocity error, respectively, of agent i with respect to the agent in front $i - 1$. Suppose that the discontinuous twisting controller is used, which can be written as

$$u_i = -a \operatorname{sign}(e_{x,i}) - b \operatorname{sign}(e_{v,i}).\tag{A.6}$$

Note that a standard stability proof cannot be applied, since u_{i-1} is bounded by the same value as u_i , and hence the disturbance cannot be compensated. In [126], a stability proof for consensus problems has been presented for the twisting controller. This approach has been discussed further in [125], and the discussions are briefly stated here.

It has been shown that the leader-tracking (A.4) can be achieved for a system (5.1), (5.2) with the consensus error defined in (A.3) using the control input (6.13) that is based on the local information only, if the accelerations fulfill

$$\sup_t |u_r| < u_{\max} , \quad (\text{A.7})$$

$$|u_i| \leq u_{\max} = a + b . \quad (\text{A.8})$$

Since the leader's acceleration is bounded and less than the acceleration u_1 of Agent 1, the position error of the first agent $e_{x,1}$ converges to zero in finite time. Then in sliding, the equivalent control is given by $u_{\text{eq},1} = u_r$. Hence, the first agent's equivalent control is bounded and less than the maximum acceleration of the second agent u_2 , and the second agent can reach the sliding surface and maintain sliding with $u_{\text{eq},2} = u_{\text{eq},1} = u_r$. Then, the third agent's position error converges, and so on. Finally, given a finite number of agents, finite-time convergence of the leader-tracking problem can be proven.

In [125], the Lyapunov function

$$V = a \|\mathbf{M}\mathbf{e}_{x0}\|_1 + \frac{1}{2} \mathbf{e}_{v0}^T \mathbf{M} \mathbf{e}_{v0} , \quad (\text{A.9})$$

has been proposed with

$$\mathbf{M} = \mathbf{L}_f - \text{diag}(l_{1,0}, \dots, l_{N,0}) \quad (\text{A.10})$$

wherein \mathbf{L}_f denotes the Laplacian that is composed only of the followers, and $l_{N,0}$ denotes the corresponding matrix entry of \mathbf{L}_f . The symbol \mathbf{e}_{x0} denotes the stacked vector of the position errors with respect to the leader $e_{x,i0}$, and \mathbf{e}_{v0} is the vector of velocity errors $e_{v,i0}$ for $i = 1, 2, \dots, N$.

The authors claim the following: If the system is initialized in an arbitrary vicinity of the origin, then the system's trajectories cannot leave this vicinity. Eventually, it can be shown that the leader-tracking problem can be solved if the leader's acceleration is bounded and lower than the following agents' acceleration. For details, the interested reader is referred to [125].

The results of a leader-tracking scenario with 6 following agents with non-zero initial spacing errors using the twisting controller are shown in Figure A.1. Eventually, all agents can track the leader with $u_{\text{eq},i} = u_r$, and finite-time convergence of the position error occurs sequentially for all agents. Note also, that the second derivative of the sliding variable can be zero during the reaching phase of the preceding vehicle (i.e., when both sliding variables have the same sign), which in Figure A.2 corresponds to constant velocity errors.

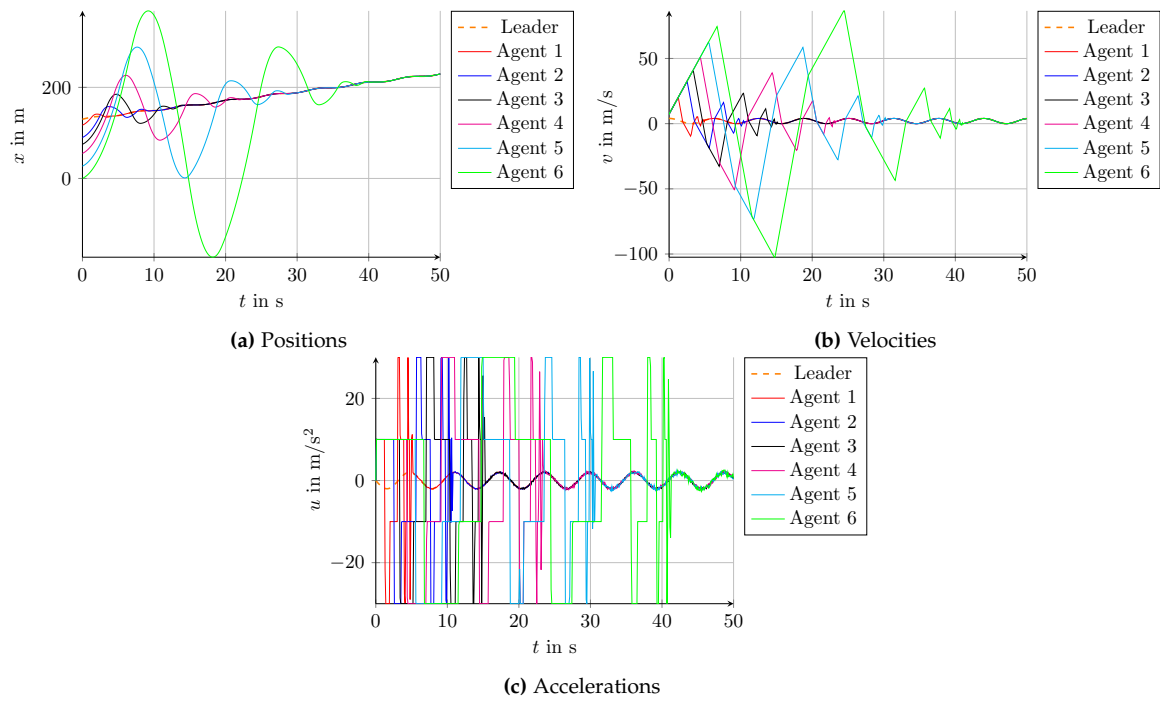


Figure A.1.: Results using the twisting controller for leader-tracking consensus problems

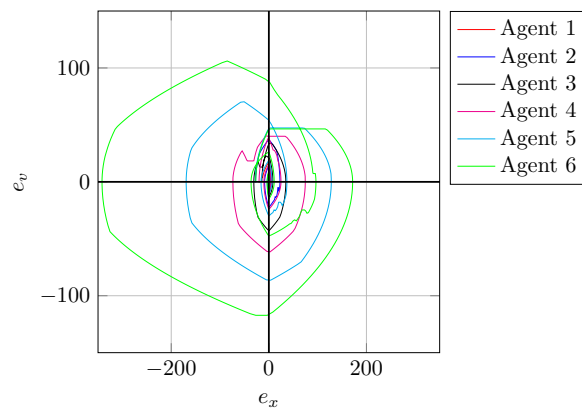


Figure A.2.: Phase plane using the twisting controller for leader-tracking consensus problems

Appendix B.

Linear Time-Varying Systems

Consider the linear time-invariant (LTI) system

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) , \quad (\text{B.1})$$

with states $\mathbf{x} \in \mathbb{R}^n$, control input $u \in \mathbb{R}$, and matrix \mathbf{A} and vector \mathbf{b} of appropriate dimensions. Then, the solution is given by

$$\mathbf{x}(t) = \Phi(t, t_0)\mathbf{x}(t_0) + \int_{t_0}^t \Phi(t, \tau)\mathbf{b}u(\tau)d\tau , \quad (\text{B.2})$$

where

$$\Phi(t, t_0) = e^{\mathbf{A}(t-t_0)} \quad (\text{B.3})$$

is the transition matrix of the system. The stability of the system is determined by the eigenvalues of matrix \mathbf{A} ; in particular, the system is asymptotically stable, i.e.,

$$\lim_{t \rightarrow \infty} \Phi(t, t_0) = 0 , \quad (\text{B.4})$$

if and only if all eigenvalues have negative real parts.

For the linear time-varying (LTV) system

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{b}(t)u(t) , \quad (\text{B.5})$$

however, the eigenvalues cannot be used to analyze stability. For example, the eigenvalues of the matrix

$$\mathbf{A}_1(t) = \begin{bmatrix} -1 - 2 \cos(4t) & 2 + 2 \sin(4t) \\ -2 + 2 \sin(4t) & -1 + 2 \cos(4t) \end{bmatrix} \quad (\text{B.6})$$

are at $s_{1,2} = -1$ for all times. Nevertheless, the system $\dot{\mathbf{x}}(t) = \mathbf{A}_1(t)\mathbf{x}(t)$ with (B.6) is unstable, because it has an unbounded solution

$$\mathbf{x}(t) = e^t \begin{bmatrix} \sin(2t) \\ \cos(2t) \end{bmatrix} \quad \text{for } \mathbf{x}(t_0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} . \quad (\text{B.7})$$

In the simple scalar case $b(t) = 1$, and $A(t) = a(t)$, for the solution in (B.2), the computation of the transition matrix is given by

$$\Phi(t, t_0) = e^{\int_{t_0}^t a(\tau)d\tau} . \quad (\text{B.8})$$

Appendix B. Linear Time-Varying Systems

Note that if $a(t)$ is negative for all t , the system is stable; however, the convergence is different to the LTI system case. Consider the following example with $x(t_0) = 0$ and a step function u ,

$$\text{LTI:} \quad \dot{x} = ax + u, \quad \text{with } a = -1 \quad (\text{B.9})$$

$$\text{LTV:} \quad \dot{x} = a(t)x + u, \quad \text{with } a(t) = -\min(kt, 1). \quad (\text{B.10})$$

Then, the solutions of the LTI system are given by (B.2) with the transition matrix (B.3), while the solution of the LTV system is computed by the transition matrix (B.8), and the results are shown in Figure B.1. Note that while the state of the LTI system (green dotted line) cannot be larger than u/a_{\max} for zero initial errors, the same cannot be guaranteed for the LTV system. Larger slopes k yield behavior similar to the LTI case (red line), while small $a(t)$ yield solutions that approach $u/a(t)$ as indicated by the blue line.

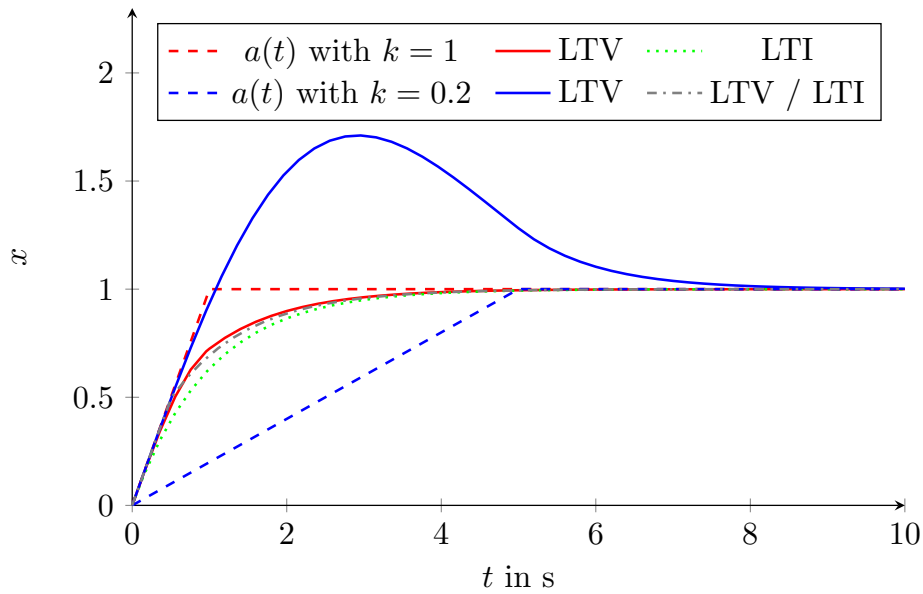


Figure B.1.: Example for LTI and LTV systems with different convergence

However, assume that a switch from the LTV to the LTI system is implemented at $x = 0.5$, then this is equivalent to the LTI system starting with $x(t_0) = 0.5$. No overshoot can appear as depicted by the gray dash-dotted line. Note that the switch is executed based on the state x . If the switch is performed based on time information only, it cannot be guaranteed that no overshoot occurs.

Bibliography

- [1] AprilTags. <http://april.eecs.umich.edu/software/apriltag.html>. Accessed: 2018-04-13.
- [2] AprilTags C++ Library. <http://people.csail.mit.edu/kaess/apriltags/>. Accessed: 2018-04-13.
- [3] MATLAB Kalman Filter Block. <http://de.mathworks.com/help/control/getstart/estimating-states-of-time-varying-systems-using-kalman-filters.html>. Accessed: 2018-04-13.
- [4] MathWork's Simulink . <http://de.mathworks.com/products/simulink.html>. Accessed: 2018-04-13.
- [5] Mathwork's MATLAB. <http://de.mathworks.com/products/matlab/>. Accessed: 2018-04-13.
- [6] IPG CarMaker. <http://ipg-automotive.com/products-services/simulation-software/carmaker/>. Accessed: 2018-04-13.
- [7] Simulation of Urban MObility. http://sumo.dlr.de/wiki/Simulation_of_Urban_MObility_-_Wiki. Accessed: 2018-04-13.
- [8] RT PREEMPT patch for BeagleBone (mainline (4.4.x lts)). http://elinux.org/BeagleBoardDebian#Mainline_.284.4.x_lts.29. Accessed: 2018-04-13.
- [9] Logitech WebCam C930e: business webcam with a wide field of view. <http://www.logitech.com/de-at/product/c930e-webcam>. Accessed: 2018-04-13.
- [10] VESC. <http://vedder.se/2015/01/vesc-open-source-esc/>. Accessed: 2018-04-13.
- [11] GoPro. <https://gopro.com/>. Accessed: 2017-06-15.
- [12] Tamiya Modelltrucks. <http://www.tamiya.de/de/produkte/rcmodelltrucks.htm>. Accessed: 2018-04-13.
- [13] Reely Straßenmodell AUDI RS6 Brushed 1:10 RC Modellauto. www.conrad.at/de/reely-strassenmodel-audi-rs6-brushed-110-rc-modellauto-elektro-strassenmodell-allradantrieb-rtr-24-ghz-238002.html. Accessed: 2018-04-13.
- [14] BeagleBone Black Board. <http://beagleboard.org/black>. Accessed: 2018-04-13.
- [15] TP-Link Router TL-WR802N. http://www.tp-link.com/at/products/details/cat-9_TL-WR802N.html. Accessed: 2018-04-13.
- [16] Brushless DC motor AUDI. <http://www.amainhobbies.com/lrp-vector-x20-brushless-motor-6.5-lrp50674/p226984>. Accessed: 2018-04-13.

Bibliography

- [17] Optoma ultra short throw projector. <http://www.optoma.de/projectorproduct/x320ust>. Accessed: 2018-04-13.
- [18] SAE Levels. https://www.sae.org/misc/pdfs/automated_driving.pdf. Accessed: 2017-04-03.
- [19] ABS - die Geschichte des Anti-Blockier-Systems. <http://www.auto-motor-und-sport.de/news/abs-die-geschichte-des-anti-blockier-systems-1790991.html>. Accessed: 2017-07-32.
- [20] Traffic Jam Assist. <https://www.continental-automotive.com/en-gl/Passenger-Cars/Chassis-Safety/Software-Functions/Cruising/Traffic-Jam-Assist>. Accessed: 2017-07-24.
- [21] Vienna convention on road traffic. https://treaties.un.org/Pages/ViewDetailsIII.aspx?src=IND&mtdsg_no=XI-B-19&chapter=11&Temp=mtdsg3&clang=_en,1968. Accessed: 2018-02-02.
- [22] California PATH program. <http://www.path.berkeley.edu/>, 1986. Accessed: 2017-08-17.
- [23] SARTRE Project. <http://www.roadtraffic-technology.com/projects/the-sartre-project/>, 2009-2012. Accessed: 2017-08-17.
- [24] DFG Schwerpunktprogramm 1835. <http://www.mrt.kit.edu/spp1835/>. Accessed: 2017-08-17.
- [25] European Truck Platooning Challenge. <http://www.eutruckplatooning.com/>, 2016. Accessed: 2017-10-18.
- [26] Ko-HAF. <http://www.ko-haf.de/startseite/>. Accessed: 2017-08-17.
- [27] KTH Smart Mobility Lab. <https://www.kth.se/en/ees/omskolan/organisation/avdelningar/ac/research/control-of-transport/smart-mobility-lab/smart-mobility-lab-1.441539>. Accessed: 2017-10-19.
- [28] NYS DMV - driver's manual - chapter 8: Defensive driving. <https://dmv.ny.gov/about-dmv/chapter-8-defensive-driving>. Accessed: 2017-11-13.
- [29] M. Amodeo, A. Ferrara, R. Terzaghi, and C. Vecchio. Slip control for vehicles platooning via second order sliding modes. In *2007 IEEE Intelligent Vehicles Symposium*, pages 761–766, June 2007.
- [30] K. J. Åström and T. Hägglund. *PID controllers: theory, design, and tuning*, volume 2. Instrument society of America Research Triangle Park, NC, 1995.
- [31] S. Baros and M. D. Ilić. A consensus approach to real-time distributed control of energy storage systems in wind farms. *IEEE Transactions on Smart Grid*, PP(99):1–1, 2017.
- [32] G. Bartolini, A. Ferrara, and E. Usai. Output tracking control of uncertain nonlinear second-order systems. *Automatica*, 33(12):2203 – 2212, 1997.
- [33] G. Bartolini, A. Ferrara, and E. Usai. Chattering avoidance by second-order sliding mode control. *IEEE Transactions on Automatic Control*, 43(2):241–246, Feb 1998.

- [34] G. Bartolini, A. Pisano, E. Punta, and E. Usai. A survey of applications of second-order sliding mode control to mechanical systems. *International Journal of control*, 76(9-10):875–892, 2003.
- [35] A. Bartoszewicz. Discrete-time quasi-sliding-mode control strategies. *IEEE Transactions on Industrial Electronics*, 45(4):633–637, Aug 1998.
- [36] L.D. Baskar, B. De Schutter, J. Hellendoorn, and Z. Papp. Traffic control and intelligent vehicle highway systems: a survey. *IET Intelligent Transport Systems*, 5(1):38–52, 2011.
- [37] E. Bauer, F. Lotz, M. Pfromm, M. Schreier, B. Abendroth, S. Cieler, A. Eckert, A. Hohm, S. Lueke, P. Rieth, V. Willert, and J. Adamy. Proreta 3: An integrated approach to collision avoidance and vehicle automation. *Automatisierungstechnik*, 60(12):755–765, 2012.
- [38] A. Bemporad, M. Heemels, and M. Johansson, editors. *Networked Control Systems*. Springer, 2010.
- [39] R. Bencatel, M. Faied, J. Sousa, and A. R. Girard. Formation control with collision avoidance. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 591–596, Dec 2011.
- [40] H.H. Bengtsson, L. Chen, A. Voronov, and C. Englund. Interaction protocol for highway platoon merge. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 1971–1976. IEEE, 2015.
- [41] D. Bevly, X. Cao, M. Gordon, G. Ozbilgin, D. Kari, B. Nelson, J. Woodruff, M. Barth, C. Murray, A. Kurt, K. Redmill, and U. Ozguner. Lane change and merge maneuvers for connected and automated vehicles: A survey. *IEEE Transactions on Intelligent Vehicles*, 1(1):105–120, March 2016.
- [42] S.P. Bhat and D.S. Bernstein. Geometric homogeneity with applications to finite-time stability. *Mathematics of Control, Signals, and Systems (MCSS)*, 17(2):101–127, 2005.
- [43] F. Borrelli, T. Keviczky, K. Fregene, and G.J. Balas. Decentralized receding horizon control of cooperative vehicle formations. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 3955–3960. IEEE, 2005.
- [44] W. Cao, M. Muka, T. Kawabe, H. Nishira, and N. Fujiki. Merging trajectory generation for vehicle on a motor way using receding horizon control framework. In *2014 IEEE Conference on Control Applications (CCA), Part of 2014 IEEE Multi-conference on Systems and Control*, pages 2127–2134, October 2014.
- [45] Y. Cao and W. Ren. Distributed Coordinated Tracking with Reduced Interaction via a Variable Structure Approach. *IEEE Transactions on Automatic Control*, 57(1):33–48, Jan 2012.
- [46] A. Chevalier, C. Copot, S.M. Cristescu, C.M. Ionescu, and R. De Keyser. Emulation of a highway bottleneck using leader-follower formation control. In *2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pages 131–136. IEEE, 2013.

- [47] B. Choi, S. Lin, and E.S. Peters. Extended driver-assisted merging protocol. Technical report, Columbia University in the City of New York, Department of Electrical Engineering, 2012.
- [48] D. Corona and B. De Schutter. Adaptive cruise control for a smart car: A comparison benchmark for mpc-pwa control methods. *IEEE Transactions on Control Systems Technology*, 16(2):365–372, March 2008.
- [49] A.K. Das, R. Fierro, V. Kumar, J.P. Ostrowski, J. Spletzer, and C.J. Taylor. A vision-based formation control framework. *IEEE transactions on robotics and automation*, 18(5):813–825, 2002.
- [50] L.C. Davis. Improving traffic flow at a 2-to-1 lane reduction with wirelessly connected, adaptive cruise control vehicles. *Physica A: Statistical Mechanics and its Applications*, 451:320 – 332, 2016.
- [51] M. Defoort, T. Floquet, A. Kokosy, and W. Perruquetti. Sliding-mode formation control for cooperative autonomous mobile robots. *IEEE Transactions on Industrial Electronics*, 55(11):3944–3953, 2008.
- [52] M. Defoort, T. Floquet, A. Kokosy, and W. Perruquetti. A novel higher order sliding mode control scheme. *Systems & Control Letters*, 58(2):102 – 108, 2009.
- [53] S. Dermann and R. Isermann. Nonlinear distance and cruise control for passenger cars. In *American Control Conference, Proceedings of the 1995*, volume 5, pages 3081–3085 vol.5, Jun 1995.
- [54] D. B. Edwards, T. A. Bean, D. L. Odell, and M. J. Anderson. A leader-follower algorithm for multiple auv formations. In *2004 IEEE/OES Autonomous Underwater Vehicles (IEEE Cat. No.04CH37578)*, pages 40–46, June 2004.
- [55] T. Eigel. *Integrierte Längs-und Querführung von Personenkraftwagen mittels Sliding-Mode-Regelung*. PhD thesis, Technische Universität Braunschweig, 2009.
- [56] S. Ellwanger and E. Wohlfarth. Truck platooning application. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 966–971, June 2017.
- [57] C. Englund, L. Chen, J. Ploeg, E. Semsar-Kazerooni, A. Voronov, H. H. Bengtsson, and J. Didoff. The grand cooperative driving challenge 2016: boosting the introduction of cooperative automated vehicles. *IEEE Wireless Communications*, 23(4):146–152, August 2016.
- [58] J. Eyre, D. Yanakiev, and I. Kanellakopoulos. String stability properties of AHS longitudinal vehicle controllers. *Transportation Systems*, pages 71–76, 1997.
- [59] J. Eyre, D. Yanakiev, and I. Kanellakopoulos. A simplified framework for string stability analysis of automated vehicles. *Vehicle System Dynamics*, 30(5):375–405, 1998.
- [60] P. Falcone, F. Borrelli, J. Asgari, H.E. Tseng, and D. Hrovat. A model predictive control approach for combined braking and steering in autonomous vehicles. In *2007 Mediterranean Conference on Control Automation. MED '07.*, pages 1–6, June 2007.
- [61] J.A. Fax and R.M. Murray. Information Flow and Cooperative Control of Vehicle Formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, Sept 2004.

- [62] D. Ferguson, T. Howard, and M. Likhachev. Motion planning in urban environments: Part i. In *Proceedings of the IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems*, September 2008.
- [63] D. Ferguson, T. Howard, and M. Likhachev. Motion planning in urban environments: Part ii. In *IROS'08*, pages 1070–1076, 2008.
- [64] A. Ferrara and G.P. Incremona. Sliding modes control in vehicle longitudinal dynamics control. In *Advances in Variable Structure Systems and Sliding Mode Control—Theory and Applications*, pages 357–383. Springer, 2018.
- [65] A. Ferrara, R. Librino, A. Massola, M. Miglietta, and C. Vecchio. Sliding mode control for urban vehicles platooning. In *2008 IEEE Intelligent Vehicles Symposium*, pages 877–882, June 2008.
- [66] A. Ferrara and C. Vecchio. Controlling a platoon of vehicles via a second order sliding mode approach. *IFAC Proceedings Volumes*, 39(12):439–444, 2006.
- [67] A. Ferrara and C. Vecchio. Collision avoidance strategies and coordinated control of passenger vehicles. *Nonlinear Dynamics*, 49(4):475–492, 2007.
- [68] A. Ferrara and C. Vecchio. Second order sliding mode control of vehicles with distributed collision avoidance capabilities. *Mechatronics*, 19(4):471 – 477, 2009. Robotics and Factory of the Future, New Trends and Challenges in Mechatronics.
- [69] M. Franceschelli, A. Pilloni, A. Pisano, A. Giua, and E. Usai. Finite-Time Consensus with Disturbance Attenuation for Directed Switching Network Topologies by Discontinuous Local Interactions. In *2013 IEEE 52nd Annual Conference on Decision and Control (CDC)*, pages 2611–2616, Dec 2013.
- [70] U. Franke, S. Mehring, A. Suissa, and S. Hahn. The daimler-benz steering assistant: a spin-off from autonomous driving. In *Intelligent Vehicles '94 Symposium, Proceedings of the*, pages 120–124, Oct 1994.
- [71] L. Fridman, Y. Shtessel, C. Edwards, and X. Yan. Higher-Order Sliding-Mode Observer for State Estimation and Input Reconstruction in Nonlinear Systems. *International Journal of Robust and Nonlinear Control*, 18(4-5):399–412, 2008.
- [72] O. Gehring and H. Fritz. Practical results of a longitudinal control concept for truck platooning with vehicle to vehicle communication. In *Proceedings of Conference on Intelligent Transportation Systems*, pages 117–122, Nov 1997.
- [73] 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*, 13(3):1008–1017, Sept 2012.
- [74] C. Goerzen, Z. Kong, and B. Mettler. A survey of motion planning algorithms from the perspective of autonomous uav guidance. *J. Intell. Robotics Syst.*, 57(1-4):65–100, January 2010.
- [75] M. Goli and A. Eskandarian. Evaluation of lateral trajectories with different controllers for multi-vehicle merging in platoon. In *2014 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 673–678. IEEE, 2014.

Bibliography

- [76] B. Großwindhager, A. Rupp, M. Tappler, M. Tranninger, S. Weiser, B. K. Aichernig, C.A. Boano, M. Horn, G. Kubin, S. Mangard, M. Steinberger, and K. Römer. Dependable internet of things for networked cars. *International Journal of Computing*, 16(4):226–237, 2017.
- [77] X. Guo, J. Wang, F. Liao, and R.S.H. Teo. Distributed adaptive integrated-sliding-mode controller synthesis for string stability of vehicle platoons. *IEEE Transactions on Intelligent Transportation Systems*, 17(9):2419–2429, 2016.
- [78] R. Hall and C. Chin. Vehicle sorting for platoon formation: Impacts on highway entry and throughput. *Transportation Research Part C: Emerging Technologies*, 13(5):405–420, 2005.
- [79] M.T. Hamayun, C. Edwards, and H. Alwi. *Fault Tolerant Control Schemes Using Integral Sliding Modes*. Springer, 2016.
- [80] C. Hatipoglu, K. Redmill, and U. Ozguner. Steering and lane change: a working system. In *Proceedings of Conference on Intelligent Transportation Systems*, pages 272–277, Nov 1997.
- [81] J. K. Hedrick, M. Tomizuka, and P. Varaiya. Control issues in automated highway systems. *IEEE Control Systems*, 14(6):21–32, Dec 1994.
- [82] K. Hnatek-Petrak and M. Vergeiner. Der Sicherheitsabstand beim Hintereinanderfahren. *Zeitschrift für Verkehrsrecht (ZVR)*, 10:355–360, 10 2002. <https://www.kfv.at/fileadmin/content/Taetigkeitsbereich/ZVR/2002/10.pdf>.
- [83] R. Hult and R.S. Tabar. *Path planning for highly automated vehicles*. PhD thesis, Master’s thesis, Chalmers University of Technology, Gothenburg, 2013.
- [84] P. Ioannou, Z. Xu, S. Eckert, D. Clemons, and T. Sieja. Intelligent cruise control: theory and experiment. In *Proceedings of 32nd IEEE Conference on Decision and Control*, pages 1885–1890 vol.2, Dec 1993.
- [85] A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, June 2003.
- [86] B. Johansson, T. Keviczky, M. Johansson, and K.H. Johansson. Subgradient methods and consensus algorithms for solving convex optimization problems. In *2008 47th IEEE Conference on Decision and Control*, pages 4185–4190, Dec 2008.
- [87] K.H. Johansson. Cyber-physical control of road freight transport. In *2016 IEEE International Conference on Autonomic Computing (ICAC)*, pages 4–4, July 2016.
- [88] H. Katayama. Sampled-data consensus control for nonlinear fully-actuated ships. In *49th IEEE Conference on Decision and Control (CDC)*, pages 3724–3729, Dec 2010.
- [89] C. Katrakazas, M. Quddus, W. 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*, 60:416 – 442, 2015.

- [90] E.S. Kazerooni and J. Ploeg. Interaction protocols for cooperative merging and lane reduction scenarios. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 1964–1970. IEEE, 2015.
- [91] H.K. Khalil. *Nonlinear Systems*. Prentice-Hall, 2002.
- [92] S. Khoo, L. Xie, and Z. Man. Robust Finite-Time Consensus Tracking Algorithm for Multirobot Systems. *IEEE/ASME Transactions on Mechatronics*, 14(2):219–228, April 2009.
- [93] S. Klinge and R.H. Middleton. Time headway requirements for string stability of homogeneous linear unidirectionally connected systems. In *Proceedings of the 48th IEEE conference on Decision and control, 2009 held jointly with the 2009 28th Chinese control conference. CDC/CCC 2009.*, pages 1992–1997. IEEE, 2009.
- [94] S. Kokogias, L. Svensson, G.C. Pereira, R. Oliveira, X. Zhang, X. Song, and J. Mårtensson. Development of platform-independent system for cooperative automated driving evaluated in gcdc 2016. *IEEE Transactions on Intelligent Transportation Systems*, 2017.
- [95] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J.P. How. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 17(5):1105–1118, 2009.
- [96] J. Kwon and D. Chwa. Adaptive bidirectional platoon control using a coupled sliding mode control method. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2040–2048, 2014.
- [97] M. Larsson, J. Lindberg, J. Lycke, K. Hansson, A. Khakulov, E. Ringh, F. Svensson, I. Tjernberg, A. Alam, J. Araujo, et al. Towards an indoor testbed for mobile networked control systems. In *First Workshop on Research, Development, and Education on Unmanned Aerial Systems (RED-UAS 2011), Seville (Spain)*, 2011.
- [98] M. Lauer. Grand cooperative driving challenge 2011 (its events). *IEEE Intelligent Transportation Systems Magazine*, 3(3):3, 2011.
- [99] L. Leclercq, V.L. Knoop, F. Marczak, and S.P. Hoogendoorn. Capacity drops at merges: New analytical investigations. *Transportation Research Part C: Emerging Technologies*, 62:171–181, 2016.
- [100] A. Levant. Sliding order and sliding accuracy in sliding mode control. *International Journal of Control*, 58(6):1247–1263, 1993.
- [101] A. Levant. Higher-Order Sliding Modes, Differentiation and Output-Feedback Control. *International Journal of Control*, 76(9-10):924–941, 2003.
- [102] S.E. Li, Y. Zheng, K. Li, and J. Wang. An overview of vehicular platoon control under the four-component framework. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 286–291, June 2015.

Bibliography

- [103] A. Manecy, N. Marchand, and S. Viollet. RT-MaG: An open-source SIMULINK toolbox for linux-based real-time robotic applications. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*. Institute of Electrical and Electronics Engineers (IEEE), dec 2014.
- [104] D. Marinescu, J. Čurn, M. Bouroche, and V. Cahill. On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 900–906. IEEE, 2012.
- [105] H. Marzbani, R. Jazar, and M. Fard. Better road design using clothoids. In Inge-mar Denbratt, Aleksandar Subic, and Jörg Wellnitz, editors, *Sustainable Automotive Technologies 2014. Lecture Notes in Mobility*, chapter 3, pages 25–40. Springer, 2015.
- [106] C. Mayer, K. Naab, and W. Kagerer. Interactive cruise control - a new driver interaction concept for adaptive cruise control systems. In *SAE 2006 World Congress & Exhibition*. SAE International, apr 2006.
- [107] M. McNaughton, C. Urmson, J.M. Dolan, and J. Lee. Motion planning for autonomous driving with a conformal spatiotemporal lattice. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4889–4895, May 2011.
- [108] 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*, 15(1):296–305, Feb 2014.
- [109] A. Morales and H. Nijmeijer. Merging strategy for vehicles by applying cooperative tracking control. *IEEE Transactions on Intelligent Transportation Systems*, 2016.
- [110] A. Mosebach, S. Röchner, and J. Lunze. Merging control of cooperative vehicles. *IFAC-PapersOnLine*, 49(11):168–174, 2016.
- [111] A. Mosser. Fahrzeug Umfeldmodell für autonomes fahren. Master’s thesis, Graz University of Technology, 2016.
- [112] 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*, 59(9):4268–4279, 2010.
- [113] G. Nestlinger, A. Rupp, H. Martin, M. Frischmann, J. Holzinger, Stabentheiner G., and M. Stolz. Virtual concept development on the example of a motorway chauffeur. In JKU Linz, editor, *tba*. Springer International Publisher AG, 2016.
- [114] G. Nestlinger and M. Stolz. Bumpless transfer for convenient lateral car control handover. *IFAC-PapersOnLine*, 49(15):132–138, 2016.
- [115] J. Nilsson, P. Falcone, M. Ali, and J. Sjöberg. Receding horizon maneuver generation for automated highway driving. *Control Engineering Practice*, 41:124–133, 2015.
- [116] R. T. O’Brien, P. A. Iglesias, and T. J. Urban. Vehicle lateral control for automated highway systems. *IEEE Transactions on Control Systems Technology*, 4(3):266–273, May 1996.
- [117] K. Oh, M. Park, and H. Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.

- [118] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, Sept 2004.
- [119] E. Olson. AprilTag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers (IEEE), may 2011.
- [120] S. Öncü. *String stability of interconnected vehicles : network-aware modelling, analysis and experiments*. PhD thesis, Technische Universiteit Eindhoven, 2014.
- [121] S. Öncü, N. van de Wouw, W.P. M. H. Heemels, and H. Nijmeijer. String stability of interconnected vehicles under communication constraints. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 2459–2464. IEEE, 2012.
- [122] M. Ono, G. Droge, H. Grip, O. Toupet, C. Scrapper, and A. Rahmani. Road-following formation control of autonomous ground vehicles. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 4714–4721, Dec 2015.
- [123] Y. Pan and K. Furuta. Adaptive vss controller based on sliding sector. *IFAC Proceedings Volumes*, 29(1):2633–2638, 1996.
- [124] W. Perruquetti and J. Barbot. *Sliding mode control in engineering*. CRC Press, 2002.
- [125] A. Pilloni, M. Franceschelli, A. Pisano, and E. Usai. Recent Advances in Sliding-Mode Based Consensus Strategies. In *2014 13th International Workshop on Variable Structure Systems (VSS)*, pages 1–6, June 2014.
- [126] A. Pilloni, A. Pisano, M. Franceschelli, and E. Usai. Finite-Time Consensus for a Network of Perturbed Double Integrators by Second-Order Sliding Mode Technique. In *2013 IEEE 52nd Annual Conference on Decision and Control (CDC)*, pages 2145–2150, Dec 2013.
- [127] A. Pilloni, A. Pisano, M. Franceschelli, and E. Usai. Integral sliding modes for the robustification of consensus-based multi-agent based systems. In *14th International Workshop on Variable Structure Systems (VSS) 2016*, pages 222–227. IEEE, 2016.
- [128] J. Ploeg, B.T.M. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer. Design and experimental evaluation of cooperative adaptive cruise control. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 260–265, Oct 2011.
- [129] J. Ploeg, D.P. Shukla, N. van de Wouw, and H. Nijmeijer. Controller synthesis for string stability of vehicle platoons. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):854–865, 2014.
- [130] J. Ploeg, N. Van De Wouw, and H. Nijmeijer. L_p string stability of cascaded systems: Application to vehicle platooning. *IEEE Transactions on Control Systems Technology*, 22(2):786–793, 2014.
- [131] R. Rai, B. Sharma, and V. Vanualalaia. Real and virtual leader-follower strategies in lane changing, merging and overtaking maneuvers. In *2015 2nd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*, 2015.

Bibliography

- [132] F. Rauskolb, K. Berger, C. Lipski, M. Magnor, K. Cornelsen, J. Effertz, T. Form, F. Graefe, S. Ohl, W. Schumacher, J. Wille, P. Hecker, T. Nothdurft, M. Doering, K. Homeier, J. Morgenroth, L. Wolf, C. Basarke, C. Berger, T. Guelke, F. Klose, and B. Rumpe. Caroline: An autonomously driving vehicle for urban environments. In *CoRR*, 2014.
- [133] H. Raza and P. Ioannou. Vehicle following control design for automated highway systems. *IEEE Control Systems*, 16(6):43–60, Dec 1996.
- [134] W. Ren. On Consensus Algorithms for Double-Integrator Dynamics. In *2007 46th IEEE Conference on Decision and Control*, pages 2295–2300, Dec 2007.
- [135] W. Ren and R.W. Beard. A decentralized scheme for spacecraft formation flying via the virtual structure approach. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 2, pages 1746–1751, June 2003.
- [136] W. Ren and R.W. Beard. Consensus Seeking in Multiagent Systems Under Dynamically Changing Interaction Topologies. *IEEE Transactions on Automatic Control*, 50(5):655–661, May 2005.
- [137] W. Ren, R.W. Beard, and E.M. Atkins. A Survey of Consensus Problems in Multi-Agent Coordination. In *American Control Conference, 2005. Proceedings of the 2005*, pages 1859–1864 vol. 3, June 2005.
- [138] A. Richardson, J. Strom, and E. Olson. AprilCal: Assisted and repeatable camera calibration. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Institute of Electrical and Electronics Engineers (IEEE), nov 2013.
- [139] J. Rios-Torres and A.A. Malikopoulos. Automated and cooperative vehicle merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–10, 2016.
- [140] J. Rios-Torres and A.A. Malikopoulos. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1066–1077, 2017.
- [141] J. Rosenzweig and M. Bartl. A review and analysis of literature on autonomous driving. *E-Journal Making-of Innovation*, 2015.
- [142] P. E. Ross. Robot, you can drive my car. *IEEE Spectrum*, 51(6):60–90, June 2014.
- [143] A. Rupp, M. Steinberger, and M. Horn. Sliding mode based platooning with non-zero initial spacing errors. *IEEE Control Systems Letters*, 1(2):274–279, Oct 2017.
- [144] A. Rupp and M. Stolz. Eine erweiterte Bewertungsfunktion für umfassende Trajektorienplanung auf Autobahnen. In *Beiträge der VW-Gemeinschaftstagung*, 2016.
- [145] A. Rupp and M. Stolz. Survey on Control Schemes for Automated Driving on Highways. In *Automated Driving - safer and more efficient future driving*. Springer Verlag, 2016.
- [146] A. Rupp, M. Stolz, and M. Horn. Decentralized cooperative merging control using sliding mode techniques. In *Proceedings of the 15th IFAC Symposium on Control in Transportation Systems (CTS 2018)*, 2018.

- [147] A. Rupp, R. Wallner, S. Koch, M. Reichhartinger, and M. Horn. Sliding mode based platooning with actuator dynamics. In *2018 15th International Workshop on Variable Structure Systems (VSS)*, July 2018.
- [148] A. Rupp and D. Watzenig. Herausforderungen in der Planungsebene kooperativer, automatisierter Fahrzeuge (Technische Herausforderungen der Evolution von Assistenzsystemen zum fahrerlosen Fahren). *FSV-Schriftenreihe*, 17, 11 2017.
- [149] S. Santini, A. Salvi, A. S. Valente, A. Pescapè, M. Segata, and R.L. Cigno. A consensus-based approach for platooning with inter-vehicular communications. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 1158–1166. IEEE, 2015.
- [150] R. Scarinci, B. Heydecker, and A. Hegyi. Analysis of traffic performance of a merging assistant strategy using cooperative vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):2094–2103, 2015.
- [151] J. Schiffer, T. Seel, J. Raisch, and T. Sezi. Voltage stability and reactive power sharing in inverter-based microgrids with consensus-based distributed voltage control. *IEEE Transactions on Control Systems Technology*, 24(1):96–109, Jan 2016.
- [152] R. Sebastian, T. Kaufmann, F. Bolourchi, and Han-Shue Tan. Design of an automated highway systems steering actuator control system. In *Proceedings of Conference on Intelligent Transportation Systems*, pages 254–259, Nov 1997.
- [153] S.E. Shladover. Longitudinal control of automated guideway transit vehicles within platoons. *Journal of Dynamic Systems, Measurement, and Control*, 100(4):302–310, 1978.
- [154] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant. *Sliding mode control and observation*. Springer, 2014.
- [155] F. Sánchez, M. Seguer, A. Freixa, P. Andreas, K. Sochaski, and R. Holze. From adaptive cruise control to active safety systems. In *Automotive and Transportation Technology Congress and Exposition*. SAE International, Oct 2001.
- [156] D. Swaroop. *String Stability Of Interconnected Systems: An Application To Platooning In Automated Highway Systems*. PhD thesis, UC Berkeley, 1997.
- [157] D. Swaroop and J.K. Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41(3):349–357, 1996.
- [158] D. Swaroop and D. Niemann. Some new results on the oscillatory behavior of impulse and step responses for linear time invariant systems. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 3, pages 2511–2512 vol.3, Dec 1996.
- [159] Ö. Ş. Taş, N. O. Salscheider, F. Poggenhans, S. Wirges, C. Bandera, M. R. Zofka, T. Strauss, J. M. Zöllner, and C. Stiller. Making bertha cooperate-team annieway’s entry to the 2016 grand cooperative driving challenge. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–15, 2017.
- [160] H. Tan, J. Guldner, S. Patwardhan, C. Chen, and B. Bougler. Development of an automated steering vehicle based on roadway magnets-a case study of mechatronic system design. *IEEE/ASME Transactions on Mechatronics*, 4(3):258–272, Sep 1999.

Bibliography

- [161] T. Tanaka, K. Isoda, M. Ohsaki, and T. Shigehara. Traction control system for improved driving safety. In *International Pacific Conference On Automotive Engineering*. Society of Automotive Engineers of Korea, Nov 1991.
- [162] T. Tank and J. P. M. G. Linnartz. Vehicle-to-vehicle communications for avcs platooning. *IEEE Transactions on Vehicular Technology*, 46(2):528–536, May 1997.
- [163] S. Thrun, H. Montemerlo, M. and 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. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Stanley: The robot that won the darpa grand challenge: Research articles. *J. Robot. Syst.*, 23(9):661–692, September 2006.
- [164] S. Tsugawa. Inter-vehicle communications and their applications to intelligent vehicles: an overview. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 2, pages 564–569 vol.2, June 2002.
- [165] S. Tsugawa, S. Jeschke, and S. E. Shladover. A review of truck platooning projects for energy savings. *IEEE Transactions on Intelligent Vehicles*, 1(1):68–77, March 2016.
- [166] M. Tsujii, H. Takeuchi, K. Oda, and M. Ohba. Application of self-tuning to automotive cruise control. In *1990 American Control Conference*, pages 1843–1848, May 1990.
- [167] C. Urmson, J. Anhalt, H. Bae, J.A. Bagnell, C.R. Baker, R.E. Bittner, T. Brown, M. N. Clark, M. Darms, D. Demitrish, J.M. Dolan, D. Duggins, D. Ferguson, T. Galatali, C.M. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. Howard, S. Kolski, M. Likhachev, B. Litkouhi, A. Kelly, M. McNaughton, N. Miller, J. Nickolaou, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, V. Sadekar, B. Salesky, Y. Seo, S. Singh, J.M. Snider, J.C. Struble, A. Stentz, M. Taylor, W. Whittaker, Z. Wolkowicki, W. Zhang, and J. Ziglar. Autonomous Driving in Urban Environments: Boss and the Urban Challenge. *Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part I*, 25(8):425–466, June 2008.
- [168] A. Vahidi and A. Eskandarian. Research advances in intelligent collision avoidance and adaptive cruise control. *IEEE Transactions on Intelligent Transportation Systems*, 4(3):143–153, Sept 2003.
- [169] B. van Arem, C. J. G. van Driel, and R. Visser. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):429–436, Dec 2006.
- [170] M. Walter, N. Nitzsche, D. Odenthal, and S. Müller. Lateral vehicle guidance control for autonomous and cooperative driving. In *2014 European Control Conference (ECC)*, pages 2667–2672, June 2014.
- [171] M. Wang, W. Daamen, S.P. Hoogendoorn, and B. van Arem. Cooperative car-following control: Distributed algorithm and impact on moving jam features. *IEEE Transactions on Intelligent Transportation Systems*, 17(5):1459–1471, May 2016.

- [172] 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*, pages 987–993, May 2010.
- [173] H. Winner, S. Hakuli, and G. Wolf, editors. *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*. Vieweg + Teubner Verlag, 2012.
- [174] L. Xiao and F. Gao. Practical string stability of platoon of adaptive cruise control vehicles. *IEEE Transactions on intelligent transportation systems*, 12(4):1184–1194, 2011.
- [175] 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*, PP(99):1–14, 2017.
- [176] Y. Zheng, S. E. Li, K. Li, and W. Ren. Platooning of connected vehicles with undirected topologies: Robustness analysis and distributed h-infinity controller synthesis. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–12, 2017.
- [177] Y. Zheng, S.E. 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*, 17(1):14–26, 2016.
- [178] J. Ziegler, P. Bender, T. Dang, and C. Stiller. Trajectory planning for bertha - a local, continuous method. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 450–457, June 2014.