Regelungs- und Systemtheorie
Universität Kassel

Consistent Hierarchical Control in Cooperative Autonomous Driving

Dissertation zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

vorgelegt im Fachbereich Elektrotechnik/Informatik
der Universität Kassel

von

Jan Michael Eilbrecht

Eingereicht am: 30. September 2021
Disputation am: 22. April 2022
Erstgutachter: Prof. Dr.-Ing. Olaf Stursberg
Zweitgutachterin: Prof. Dr. Kathrin Flaßkamp

Contents

Zusammenfassung

Diese Arbeit thematisiert das Problem der Wegfindung für Gruppen kooperierender autonomer Fahrzeuge im Straßenverkehr. Dieses ist aufgrund nichtlinearer Systemdynamiken und nichtkonvexer Beschränkungen selbst unter vereinfachenden Annahmen (z. B. Einhaltung von Verkehrsregeln durch nicht-kooperative Verkehrsteilnehmer, fehlerfreie Kommunikation zwischen Fahrzeugen, Abwesenheit von Messfehlern) schwierig. Übliche Lösungsansätze beruhen auf einer oft hierarchischen Zerlegung in leichter, nacheinander zu lösende Teilprobleme. Dabei bestehen Abhängigkeiten, da nachfolgende auf der Lösung vorhergehender aufbauen. Existierende Verfahren ignorieren diese, weshalb nachgelagerte Teilprobleme ggf. unlösbar sind, da Lösungen vorhergehender Teilprobleme ungeeignet sind.

Diese Arbeit stellt eine Entwurfsmethodik vor, die Abhängigkeiten in einem hierarchischen Ansatz berücksichtigt. Zur Untersuchung wird ein Ansatz mit drei Ebenen angesetzt: neben der Erstellung von Referenztrajektorien (dem *Planen*) und der unterlagerten Folgeregelung legt eine dritte, diesen überlagerte Ebene die kooperierenden Gruppen sowie von diesen durchzuführende Aktionen fest. Neben dem Garantieren *hierarchischer Konsistenz*, also sicheren Zusammenwirkens, ist ein Fokus der Arbeit die Recheneffizienz.

Kern der Planung von Referenztrajektorien ist die Problemformulierung als gemischtganzzahliges Programm, dessen Lösung global optimal ist und Einhaltung von Beschränkungen garantiert. Dabei werden Prädiktionen zukünftiger Werte verwendet. Um Abweichungen zum tatsächlich resultierenden Verhalten gering zu halten, werden die Optimierungsprobleme nach dem Prinzip des *gleitenden Horizonts* regelmäßig erneut gelöst.

Zur Verbesserung der Recheneffizienz nutzt das sog. *Manöverkonzept* die inhärente Struktur des Straßenverkehrs aus. Ein Manöver charakterisiert eine Menge qualitativ ähnlicher Trajektorien einer Gruppe von Fahrzeugen und wird als hybrider Automat formuliert. Dies ermöglicht die Berechnung *steuerbarer Mengen*, die Anfangszustände enthalten, für die eine Durchführung des Manövers möglich ist. So können zulässige Optionen schnell und ohne jedesmalige Planung bestimmt und anschließend die vielversprechendste ausgewählt werden. Die *approximative Berechnung* der steuerbaren Mengen erlaubt die Anwendung des Konzepts auf höherdimensionale Zustandsräume sowie die Approximation der Lösung des Planungsproblems. Dies umgeht das Optimieren im Betrieb und reduziert Rechenzeiten.

Da die Umsetzung von Referenztrajektorien durch Fahrzeuge oft nicht fehlerfrei erfolgt, sind Sicherheitsabstände zu Hindernissen einzuplanen, während sichere Interaktion mit der Folgeregelung erfordert, nur Trajektorien zu planen, für die der größtmögliche Regelfehler definierte Sicherheitsabstände nicht verletzt. Dieser wird ermittelt anhand einer invarianten Menge der Fehlerdynamik unter Berücksichtigung von Beschränkungen, Unsicherheiten und möglicher Referenztrajektorien. Dies erfolgt nur beim Entwurf; der Betrieb des Reglers erfordert lediglich recheneffiziente Operationen. Sowohl theoretische Resultate als auch numerische Simulationen zeigen die Funktionsfähigkeit des Ansatzes.

Summary

This thesis is set within the context of the problem of motion planning in autonomous driving. Due to nonlinear system dynamics and non-convex constraints, this problem is challenging even under simplifying assumptions (e.g. non-cooperating traffic participants obey traffic rules, lossless inter-vehicle communication, absence of measuring errors). Typical solution approaches rely on a decomposition of the problem into subproblems which are easier to solve. This decomposition often results in a hierarchy of subproblems which are to be solved sequentially, where succeeding problems rely on the solution of preceding ones, such that dependencies between subproblems exist. Existing procedures for the design of such approaches have in common that these dependencies are not accounted for by the solution methodology. This can lead to situations where succeeding problems cannot be solved due to unsuitable solutions of preceeding subproblems.

This thesis presents a design methodology which is able to account for such dependencies in a hierarchical solution approach. The methodology is developed under the assumption of a three-layer hierarchical solution approach comprising trajectory planning, tracking control, and a high-level controller which is to determine cooperating groups and their actions. While the major focus of the thesis is on *hierarchical consistency*, i.e., safe interaction of the framework's layers, computational efficiency is also a main concern.

The basis of the planning problem is its formulation as a mixed-integer optimization problem, whose globally optimal solution guarantees constraint compliance. This approach uses predictions of future values and conducts the optimization frequently anew in the spirit of *receding horizon control* in order to limit the impact of uncertainty.

The so-called *maneuver concept* is introduced in order to improve computational efficiency by exploiting structure in on-road traffic. A maneuver characterizes a set of qualitatively similar trajectories of a group of vehicles and is modeled as a hybrid automaton. This allows to compute *controllable sets*, containing initial states for which a maneuver can be executed. Given these, the high-level controller can assess the feasibility of many different options without computing plans when choosing the most promising one. Approximative set computations allow for both the application to typically high-dimensional state spaces in cooperative autonomous driving and for the approximation of solutions to the planning problem. Thus, no optimization is required online, which reduces computation times.

As vehicles often fail to perfectly follow a reference trajectory, safety margins to obstacles must be provided during planning. Safe interaction between planning and tracking layer then requires to only plan trajectories for which the maximum tracking error does not violate the pre-defined safety margins. The maximum error is determined based on computation of an invariant set of the tracking error dynamics, accounting for a set of possible reference trajectories, constraints on states and inputs, and uncertainty in a vehicle's tire model. This step is only carried out during offline design, while online operation of the tracking controller only requires computationally efficient algebraic operations. Both theoretical results and numerical simulations demonstrate the efficacy of the framework.

Acknowledgments

> The rank is but the guinea's stamp,
> the man's the gowd for a' that.
>
> *Robert Burns*

First of all, I would like to thank my supervisor Prof. Dr.-Ing. Olaf Stursberg for his enduring support throughout the past years. On the one hand, he has given me much freedom in conducting the research which lead to this thesis, while at the same time, he was always willing to provide feedback, and feedback is crucial in control!

Next, I thank Prof. Dr. Kathrin Flaßkamp for serving as co-supervisor, as well as Prof. Dr. Bernhard Sick and Prof. Dr. Ludwig Brabetz for serving on the thesis committee. To all of them, I am grateful for their valuable comments throughout the process.

Special thanks go to Prof. Dr. Arno Linnemann: having assisted in several of his lectures during the past years has shaped my view on control quite a bit. Not in the field of control, but with significant impact on my education, my former teachers Helmut Bambey, Lars Böttner, and Hans-Dieter Pitz have my sincerest gratitude. On the administrative side, Elena Rapp deserves gratitude for running the group smoothly. Furthermore, I want to thank my former collegues for numerous discussions, prolonged coffee and lunch breaks at all-you-can-eat restaurants, conference trips, and the fun we had in general.

Having spent significant parts of the non-research related time in boats on the Fulda, I also thank all the people involved in the Kassel University Rowing Team for the fun during practice and regattas, training camps, and long hours on the bus.

Finally, this thesis would not have been possible without support from my friends from Brakeler and Braunschweiger days as well as from my family. I'll thank all of you in person at some point and dedicate this thesis to my late father, Elmar Eilbrecht.

1. Introduction

This thesis considers the topic of guidance and control of cooperative autonomous vehicles in on-road traffic. In the recent past, the field of autonomous driving has experienced substantial research efforts, motivated amongst others by the following anticipated benefits [106]: As a driver's attention is not required anymore, long commutes must not be considered wasted time, but can be put to good use. Also, autonomy makes efficient car sharing a realistic opportunity, which saves resources and parking lots, as less cars are required to meet the same transportation demand. Another important promise of autonomous driving is increased road safety, as better reactions in safety-critical situations are expected from technical systems. In addition to these already substantial benefits, *cooperative* autonomous driving has the potential to achieve a more efficient road and energy utilization, as agreements between vehicles and information sharing allow for less cautious driving and an improved traffic flow [70].

The development of fully autonomous vehicles requires contributions from diverse fields such as machine vision [122], localization [27], communication [145], and control [115]. The focus of this thesis lies on the control aspect, which – in its broadest notion – comprises the entire decision making process of routing a vehicle from a start to a destination. The decision making process in autonomous driving is characterized by an ever-changing environment, in which much room for decisions exists (Which road to take? Overtake? If so, when? How hard to accelerate?), which nonetheless must be made quickly, while wrong decisions can have devastating consequences. When considering cooperation as the willingness and ability of multiple vehicles to share information and to adapt their behavior, the decision making process is further complicated by an increase in both the amount of information to be considered and the number of possible decisions.

Typical realizations of the decision making process for an autonomous vehicle rely on a modular, hierarchical architecture, e.g. [118, 38, 39]. Such an architecture is attributed with computational efficiency and also flexibility of design, as sub-components are envisioned to be easiliy replacable [118]. The upper-most layer of such architectures is typically concerned with the guidance within a road network, while the lower-most layer, in contrast, is tasked with the actuation of the vehicle (often referred to as "stabilization task"[149]). Layers in between target the guidance within a road segment. Often, qualitative descriptions of a vehicle's actions (termed for example "maneuvers" [136] or "behaviors" [118]) are used in this context. Information in such architectures mainly flows top-down [105], with bottom-up flow being limited to status reports in online operation, cf. [118, 10].

Despite its potential advantages, a modular, hierarchical architecture for the realization of the decision making process also introduces new challenges and can even put the safety of a vehicle and its passengers at risk, as such an architecture relies on solving subproblems

separately. These subproblems, however, are actually often intertwined, such that the solution of a subproblem on an upper level of the hierarchy can prevent the solution of one on a subordinate level if the interdependence has been ignored throughout the solution process. In such a situation, the decision making module potentially fails to return an appropriate control command in time, which can have severe consequences. Even though it should be addressed in the specification of the interfaces between modules of hierarchical architectures, this challenge does not seem to have gained attention. Accordingly, no constructive advice for its solution is available.

Even when ignoring this issue, it might still be possible to deploy a system which is functional in many (but not necessarily *all*) situations by careful system design. However, such an approach forbids itself in safety-critical applications such as autonomous driving. Rather, methodologically sound methods are desirable which explicitly account for the interdependence of subproblems during their solution and can therefore guarantee flawless interaction of the problem solving modules under specified conditions. Such procedures, however, do not seem to have been in the focus of research, which motivates the following investigations. These evolve from the setting described in Section 1.1, from which a detailed problem statement is then derived in Section 1.2.

1.1. Setting and Assumptions

As a starting point of the investigation, the hierarchical decision making framework shown in Figure 1.1 is assumed. It divides the decision making process into a sequence of simpler decisions, which are – focusing on cooperative autonomous driving – made for groups of cooperating vehicles. Qualitatively, the three layers of the framework interact as follows:

First, on the high-level control layer, the intended behavior of one (or several) vehicle(s) is determined, for example whether a group is to cooperatively enable another vehicle to merge onto a lane. Then, from this decision, a *plan* is derived, i.e., a sequence of way points at defined time instances which each cooperating autonomous vehicle should follow in order to realize the behavior prescribed by the high-level control layer. Plans are generated on the trajectory planning layer. However, a plan as a sequence of way points does not define how to actuate a vehicle; this is facilitated on the tracking control layer, where steering angle and acceleration commands are generated based on a given plan.

While decisions (i.e., the intended behaviors and the plans) are communicated explicitly downwards from upper to lower layers, an implicit dependency exists in the opposite direction because a vehicle's operation is subject to physical *constraints*, which likewise reduce the freedom of decision of the upper layers.

The following assumptions are made about the setting in which the framework is deployed – not necessarily because they all are deemed realistic, but in order to limit the scope of this thesis: First, all following considerations focus on a highway setting, which is arguably the most structured traffic environment in the sense that the number of emerging behaviors is comparably small, amongst others due to strict traffic rules and the absence of pedestrians and cyclists. For this reason, autonomous driving is likely to be realized in

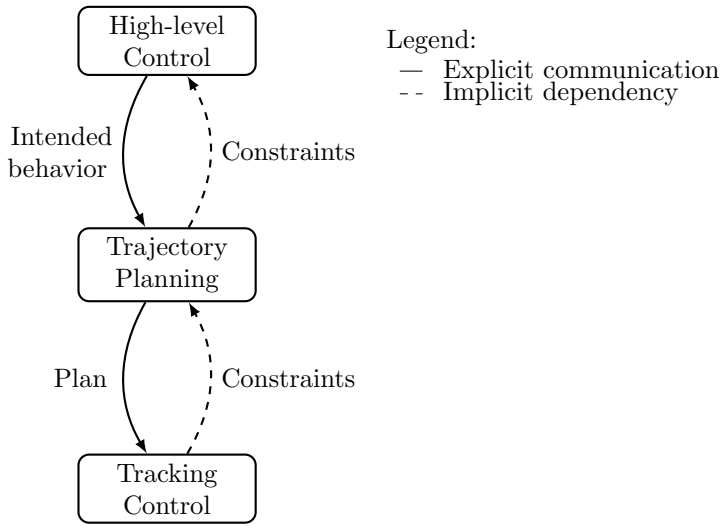


Figure 1.1.: Starting point of the thesis: a hierarchical decision making framework.

highway settings first. Note that even though pedestrians and cyclists will not be considered in this thesis, a first attempt in that direction has been presented in [42].

Furthermore, it is assumed that other traffic participants (especially the non-cooperating ones) do not violate traffic rules, as all other cases are primarily a concern of law reinforcement. No differences are assumed regarding the dynamics of all vehicles, which is obviously unrealistic, but does not make a conceptual difference for the following considerations, as the assumption can be resolved in a conceptually simple, yet tedious way. For the same reason, no curved roads are considered, refering to the method devised in [48] for reducing planning on curved roads to planning on straight roads.

Finally – and this is not unrealistic – it is assumed that each vehicle knows a route between its current position and the destination of its journey, e.g. determined by conventional navigation systems. This assumption allows to focus on guidance of the vehicle within the sequence of roads defined by the route. Questions pertaining to the problem of cooperative guidance, but not a subject of control engineering, such as communication between cooperating vehicles or distributed computation, are not considered in the following. More technical assumptions will be given throughout the thesis where required.

1.2. Overall Problem Statement and Thesis Outline

As pointed out in the beginning, a hierarchical framework can fail to operate correctly if dependencies between its layers are not accounted for. Given the setting from the previous section, the main objective of this thesis is to design the elements of the architecture shown in Fig. 1.1, ensuring *hierarchical consistency*, i.e., flawless interaction of the different layers despite implicit dependencies resulting from physical constraints on the dynamics of the controlled vehicles. This task consists of two subtasks, where the first one considers the

consistent interaction between the high-level control and the trajectory planning layer, while the second one focuses on the interaction of the trajectory planning and the tracking control layer. Furthermore, as in all real-time control systems, an important requirement is the need for computational efficiency, which must be accounted for in both subtasks. Paralleling these two, the remainder of this dissertation consists of two major parts:

Because it interacts with both other layers, the design of the trajectory planning layer is of fundamental importance and the main concern of part I. There, the design of the high-level control layer – while also relevant to the first subtask – will turn out to be less crucial, owing to the chosen design of the planning layer. Chapter 2 first provides a review of the literature related to trajectory planning, followed by the introduction of the so-called maneuver concept and the formal statement of the trajectory planning problem in the maneuver context. Also, controllable sets are introduced for feasibility assessment. On that basis, Chapter 3 first develops methods for the approximation of controllable sets, followed by methods for the approximation of solutions to the planning problem. Chapter 4 provides examples for the application of the methods derived in the two preceding chapters as well as for the interaction with an example high-level control algorithm.

Part II targets the interaction between the trajectory planning and the trajectory tracking layer. Because at this point, the trajectory planner will already have been designed as detailed in part I, the focus is on the design of the tracking controller. At first, Chapter 5 outlines the main idea and relates the approach to existing ones. Chapter 6 is concerned with the characterization of a set of vehicle states respectively reference trajectory and tracking error quantities for which admissible control inputs exist. This set is then used in Chapter 7 to prove the boundedness of the tracking error. Chapter 8 illustrates the use of the controller in an overtaking maneuver. Conclusions regarding the devised methods are drawn in Part III, which also outlines directions of future research.

1.3. Statement of Contribution

This thesis is not solely concerned with the solution of a practical problem, nor is its only focus on the development of a methodology without connection to a practical problem. Rather, it is an application-guided attempt to develop a methodology that could also be applied to similar fields (such as mobile robots or UAVs), contributing in several ways.

A major contribution with regard to the problem of vehicle guidance and control is the proposal of the so-called maneuver concept, which establishes a connection to the field of hybrid systems and their verification. This allows to reduce computation times by exploiting structure in mixed-integer programming formulations of the planning task and to characterize the admissible region of these optimization problems. Because exact computations are challenging in this context, approximation-based procedures are identified to mitigate this problem.

Another major contribution is the development of a tracking controller which substantially improves on existing approaches with respect to computational efficiency, capability to account for constraints on inputs and states, consideration of uncertainty, and depth of

analysis of the properties of the resulting closed-loop system. These results are enabled by use of symbolic quantifier elimination, which is identified as a valuable tool, potentially also for problems beyond those considered in this thesis.

The results presented in this thesis have partly been published in a preliminary form in [44, 45, 42, 46, 47, 49]. The results in [43, 48], while providing the basis for some assumptions outlined in Section 1.1, are of no direct relevance to the major questions of this thesis and therefore not detailed in the following in order to limit the scope of the thesis.

1.4. Notation

Apart from the fairly standard notation being used throughout this thesis, the following conventions deserve mentioning: Given a matrix $M \in \mathbb{R}^{n \times m}$ and integers $1 \leq i \leq n$ and $1 \leq j \leq m$, the entry in the ith row and jth column is addressed as $M_{[i,j]}$. Entries of the entire ith row or jth column are selected as $M_{[i,:]}$ and $M_{[:,j]}$, respectively. Subsets of all rows/columns can be selected by index sets; for example, the entries of all rows corresponding to the indices in a set $\mathcal{I}$ can be addressed as $M_{[\mathcal{I},:]}$ (returning the rows in order of ascending indices as a convention), etc. The entries of the main diagonal of a matrix M can be extracted by $\texttt{diag}\,(M)$, while $\texttt{diag}\,(d_1, d_2, \ldots, d_n)$ is used to denote a diagonal matrix with (block) entries d_1, d_2, etc. on the main diagonal. Matrices of dimension $n \times m$ with entries all-zero or all-one are denoted by $0_{n \times m}$ and $1_{n \times m}$, respectively.

Sets will typically be denoted by calligraphic upper-case letters. For a polyhedron $\mathcal{P} \subset \mathbb{R}^d$, denote by $\texttt{f}\,(\mathcal{P})$ the set of outward-pointing unit normals of the facets of $\mathcal{P}$, i.e., the faces of dimension $d-1$. A bounded polyhedron is referred to as a *polytope*, whose set of vertices is denoted as $\texttt{v}\,(\mathcal{P})$, such that the polytope is given by:

$$\mathcal{P} = \texttt{convh}\,(\texttt{v}\,(\mathcal{P}))\,,$$

where $\texttt{convh}\,()$ denotes the convex hull of a set. The interior of a polytope $\mathcal{P}$ is denoted by $\texttt{int}(\mathcal{P})$ and its boundary by $\partial\mathcal{P}$. Projections of a set on lower dimensions are obtained by application of the `proj`-operator, where the corresponding dimensions are given as indices. For example,

$$\texttt{proj}_x \left\{ \begin{bmatrix} x \\ y \end{bmatrix} \mid \text{some predicate} \right\}$$

denotes the projection on the dimension(s) corresponding to x. While the elements of sets are enclosed by curly brackets, those of sequences are contained by round brackets, e.g. $s = (s_1, s_2, \ldots, s_N)$, or shorthand $s = (s_i)_{i=1}^N$. If the number N of elements of the sequence is of relevance, it is provided as an index: s_N, while $s(\cdot)$ is used to emphasize the sequence nature of a symbol.

Part I.

Maneuver-based Decision Making

2. The Maneuver Concept

In order to make the rather generic hierarchical architecture from Figure 1.1 more concrete, the framework as shown in Figure 2.1 is proposed. There, the high-level controller consists of a *scheduler* and a *maneuver library*. The general functionality (which will be detailed in the following chapters) of these units is such that the maneuver library contains encodings of different cooperative behaviors (the *maneuvers*) of a group of vehicles, from which the scheduler selects one and determines those among the nearby vehicles which are to execute this maneuver. For a scheduled maneuver, a trajectory planner computes reference trajectories for each vehicle of a cooperative group, which are then passed to the tracking controllers of the individual vehicles. This process, however, only reflects the top-down interaction, while also a dependency of the superior layers on the subordinate ones exists, because the capability of a vehicle to follow a reference trajectory must be considered by both the trajectory planner and the scheduler (cf. Figure 1.1).

The maneuver concept to be proposed in this part is the key to account for this requirement: on the one hand, it allows the trajectory planner to account for constraints resulting from the tracking control task, while on the other hand, it enables a quick evaluation of the feasibility of a certain maneuver in a given traffic situation. The latter aspect is very beneficial for the design of the scheduler, as it allows to reduce the scheduling problem to a choice solely between those maneuvers from the library which are admissible in the current situation. This decouples concerns of admissibility of a decision from other aspects such as its optimality, which can be addressed separately by the design of corresponding rules for choosing between the admissible maneuvers. As the main focus of this thesis is on the smooth interaction of the different hierarchical layers – which pertains to the admissibility aspect of scheduled decisions rather than optimality – no realizations of scheduling algorithms are considered here in detail, even though an example for a possible design is provided in Section 4.2. Similarly, topics pertaining to the real-world implementation of the scheduler and trajectory planner, such as decentralized computation or inter-vehicle communication, are beyond the scope of this thesis.

The admissibility of a decision is generally difficult to assess in advance because it requires to predict the decision's effect on the involved vehicles. The common approach to obtain predictions of a system's behavior (which is also followed in this thesis) is based on the use of mathematical models, which are supposed to reflect the most important properties of the actual system. Many model-based control algorithms exist which allow to establish properties such as stability and constraint compliance *for a model* of a system. This is especially beneficial in safety-critical applications, where the (typically iterative) design process does not have to be carried out during operation of the actual system (just consider control of an unstable system!). However, model-based approaches are challenged

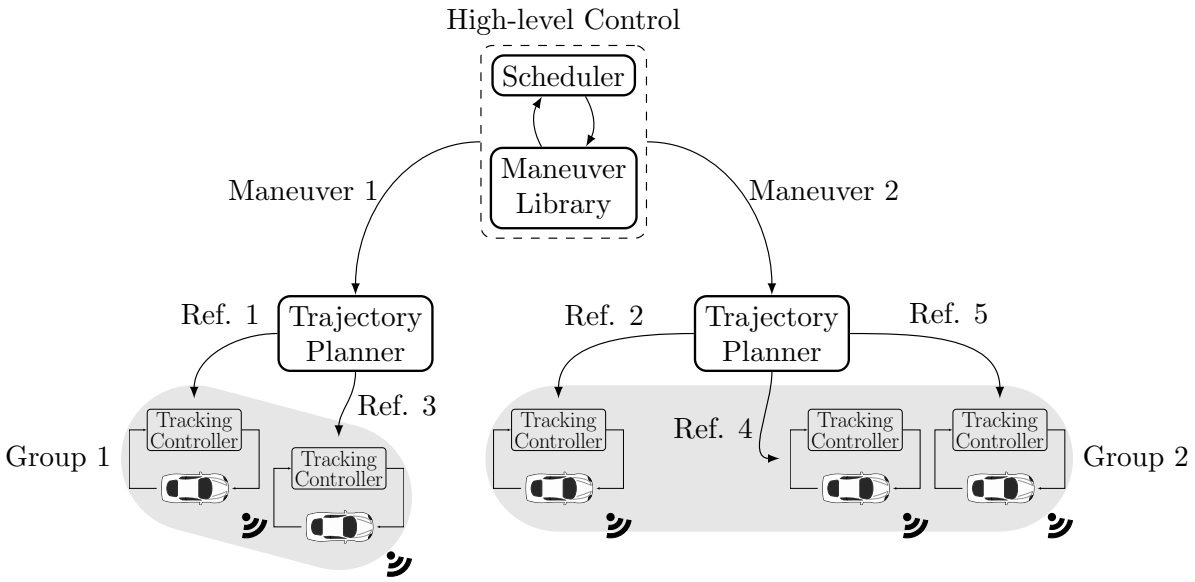


Figure 2.1.: Hierarchical Framework.

by the difficulty to establish how properties of a model transfer to the actual system. Even though verification methods and field testing can reduce this gap and therefore the likelihood of failure, the conceptual challenge remains. This thesis does not focus on such questions, but rather aims to establish consistency within the model-based controller design process. While this is clearly not sufficient to ensure the absence of harm, it is necessary for safety when a modular, hierarchical approach is chosen.

Even though non-cooperating vehicles cannot be controlled, it is necessary to predict their future behavior, because they can impact the cooperating, controlled vehicles. However, obtaining such predictions is especially challenging because one must rely on assumptions about the unknown intents of the non-cooperating vehicles. This represents another source of uncertainty in addition to that introduced by a potential mismatch between the actual system and its model. If uncertainty is high, decision making becomes difficult. The need to reduce the impact of uncertainty motivates to make frequent updates during the execution of decisions in line with the *receding horizon principle* [123]. Such updates allow to account for changes in the environment which have not been considered when making a current prediction. Updates are made on different time scales for the different layers of the framework, as a qualitative decision for a certain maneuver is typically made every few seconds, while the execution of a maneuver requires updates of the corresponding reference trajectory several times per second and the computation of control inputs is carried out in the millisecond range.

This part considers the problem of cooperative trajectory planning, which is concerned with settings like the one shown in Figure 2.2: a group of autonomous, cooperating vehicles is driving on a road, each given knowledge of the sequence of roads (a *route*) that lead it from its starting point to its destination. The cooperative vehicles share the road with non-cooperating ones, where neighboring non-cooperative vehicles driving ahead of cooperative

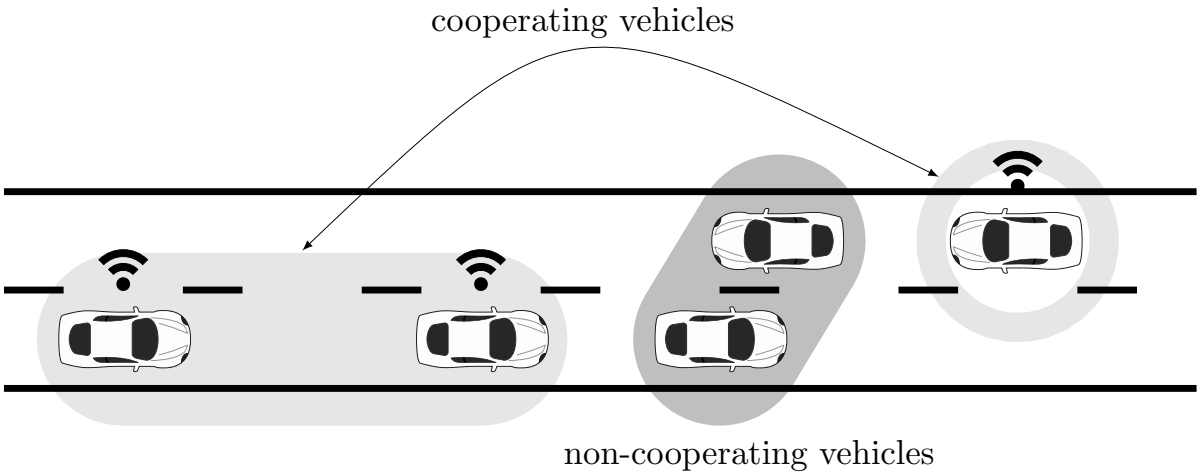


Figure 2.2.: The general setting: cooperating and non-cooperating vehicles share the road.

ones deserve special attention. Let each of these vehicles be uniquely identified by integers in $\mathcal{C} \subset \mathbb{N}$ (the *cooperating* group) and $\mathcal{N} \subset \mathbb{N}$ (the group of *non-cooperative, leading neighboring* vehicles), respectively. Let a model for the temporal behavior (the *dynamics*) of a cooperating vehicle $i \in \mathcal{C}$ be given by a function $f_i : \mathbb{R}^{n_{x_i}} \times \mathbb{R}^{n_{u_i}} \times \mathbb{R}^{n_{\nu_i}} \to \mathbb{R}^{n_{x_i}}$:

$$\dot{\chi}_i(t) = f_i\left(\chi_i(t), \mu_i(t), \nu_i(t)\right), \tag{2.1}$$

with the state $\chi_i \in \mathcal{X}_i \subseteq \mathbb{R}^{n_{x_i}}$, controlled inputs $\mu_i \in \mathcal{U}_i \subset \mathbb{R}^{n_{u_i}}$, and disturbances $\nu_i \in \mathcal{V}_i \subset \mathbb{R}^{n_{\nu_i}}$ at the current time $t \in \mathbb{R}^{\geq 0}$. Assume $\mathcal{U}_i$ and $\mathcal{V}_i$ to be bounded sets. Further, let the inputs μ_i at time t be determined based on the current state $\chi_i(t)$ and a time-varying *reference* value $\bar{x}_i(t) \in \mathbb{R}^{\bar{n}_{x_i}}$ by a controller function $k_i : \mathbb{R}^{n_{x_i}} \times \mathbb{R}^{\bar{n}_{x_i}} \to \mathbb{R}^{n_{u_i}}$:

$$\mu_i(t) = k_i\left(\chi_i(t), \bar{x}_i(t)\right), \tag{2.2}$$

where $\bar{x}_i(t)$ (and correspondingly, the entire model of the controlled dynamics) is defined for $t \in [t_0, t_{\text{plan}}] \subseteq \mathbb{R}^{\geq 0}$, $t_{\text{plan}} > t_0$.

In the following, denote by $\bullet(\cdot)$ a sequence of vectors $\bullet$ over a (not necessarily explicitly specified) time range (as opposed to $\bullet(t)$, giving the vector at a single point t in time). Let the state χ_i contain information about longitudinal and lateral positions $p_x^{(i)}$, $p_y^{(i)}$, respectively, such that matrices $C_{\text{pos}}^{(i)} \in \mathbb{R}^{2 \times n_{x_i}}$ allow to extract the position of vehicle i from its state χ_i:

$$\begin{bmatrix} p_x^{(i)}(t) \\ p_y^{(i)}(t) \end{bmatrix} = C_{\text{pos}}^{(i)} \chi_i(t). \tag{2.3}$$

Then, the following notion of reference values and sequences thereof is employed:

Definition 2.1 (Reference Trajectory of vehicle i)**.** *Let a sequence of position coordinates* $\left(\bar{p}_x^{(i)}, \bar{p}_y^{(i)}\right)$ *in an earth-fixed, cartesian coordinate system be given such that* $\bar{p}_x^{(i)}$:

$[t_0, t_{plan}] \to \mathbb{R}$ *and* $\bar{p}_y^{(i)} : [t_0, t_{plan}] \to \mathbb{R}$*, where* $\bar{p}_x^{(i)}$ *and* $\bar{p}_y^{(i)}$ *are* n*-times continuously differentiable on the open interval. Then, a* reference trajectory $\bar{x}_i(\cdot)$ *is defined as a sequence of vectors* $\bar{x}_i : [t_0, t_{plan}] \to \mathbb{R}^{\bar{n}_{x_i}}$*,* $\bar{n}_{x_i} = 2(n+1)$*:*

$$\bar{x}_i(t) := \tag{2.4}$$
$$\left[\bar{p}_x^{(i)}(t) \quad \dot{\bar{p}}_x^{(i)}(t) \quad \ddot{\bar{p}}_x^{(i)}(t) \quad \dots \quad \bar{p}_x^{(n+1),(i)}(t) \quad \bar{p}_y^{(i)}(t) \quad \dot{\bar{p}}_y^{(i)}(t) \quad \ddot{\bar{p}}_y^{(i)}(t) \quad \dots \quad \bar{p}_y^{(n+1),(i)}(t)\right]^{\mathsf{T}}.$$

Along with given reference trajectory $\bar{x}_i(\cdot)$, initial state χ_0, initial time t_0, final time t_{plan}, and a disturbance trajectory $\nu_i(\cdot)$, (2.1) together with (2.2) defines an initial value problem. Its solution (whose existence and uniqueness is assumed) is the trajectory $\chi_i(\cdot)$. If the dependency on the above parameters is to be emphasized, it is alternatively written as:

$$\chi_i(\cdot; t_0, t_{\text{plan}}, \bar{x}_i(\cdot), \nu_i(\cdot)), \tag{2.5}$$

where a semicolon is used to separate the varying time argument from constant parameters of the trajectory, and

$$\begin{aligned} \chi_0 &= \chi_i(t_0; t_0, t_{\text{plan}}, \bar{x}_i(\cdot), \nu_i(\cdot)), \\ \frac{\mathsf{d}}{\mathsf{d}t}\chi_i(\cdot; t_0, t_{\text{plan}}, \bar{x}_i(\cdot), \nu_i(\cdot)) &= f_i(\chi_i(t), k_i\left(\chi_i(t), \bar{x}_i(t)\right), \nu_i(t)) \end{aligned}$$

for all $t \in [t_0, t_{\text{plan}}]$.

According to (2.5), the only means to actively influence the closed-loop trajectory is the choice of the reference trajectory $\bar{x}_i$, which consequently must be chosen such that $\chi_i(t) \in \mathcal{X}_i$ and $k_i\left(\chi_i(t), \bar{x}_i(t)\right) \in \mathcal{U}_i$ hold for all considered disturbances and future times. These constraints stem from a vehicle's dynamics and define *admissible* values for, e.g., its velocity or acceleration. In contrast, the road topology or the need to avoid collisions with other vehicles specify *forbidden* values (typically positions). Let these be encoded by a time-varying set $\mathcal{F}_i(t) \subset \mathbb{R}^2$, which – as an additional requirement – must be avoided by proper choice of $\bar{x}_i$. Thus, the constraints $\mathcal{X}_i$, $\mathcal{U}_i$, and $\mathcal{F}_i(t)$ on the state and control of the vehicle eventually also constrain the reference trajectory which is to be computed by the trajectory planner. This is the implicit dependency between the tracking control and the trajectory planning layer as indicated in Figure 1.1.

In order to make this dependency explicit, two preliminary assumptions are made at this point: First, considering obstacle avoidance, the choice of a reference trajectory must ensure that $C_{\text{pos}}^{(i)}\chi_i(t) \notin \mathcal{F}_i(t)$. Given matrices $\bar{C}_{\text{pos}}^{(i)} \in \mathbb{R}^{2\times\bar{n}_{x_i}}$ such that:

$$\begin{bmatrix} \bar{p}_x^{(i)} \\ \bar{p}_y^{(i)} \end{bmatrix} = \bar{C}_{\text{pos}}^{(i)}\bar{x}_i,$$

the following assumption – to be validated later on in Theorem 7.1 – provides the basis for collision avoidance:

Assumption 2.1 (Bounded Tracking Errors)**.** *Assume that there exists a positive scalar k_{pos} such that the deviation between reference positions and resulting positions fulfills:*

$$\left|\left|C_{pos}^{(i)}\chi_i(t;t_0,t_{plan},\bar{x}_i(\cdot),\nu_i(\cdot)) - \bar{C}_{pos}^{(i)}\bar{x}_i(t)\right|\right|_\infty \leq k_{pos}\ \forall\ \nu_i(t)\in\mathcal{V}_i,\ t_0\leq t\leq t_{plan}.$$

In addition, it is reasonable to assume that no initial position error exists, as planning commences in the current position of a vehicle.

Furthermore – as the focus of the thesis is on regular planning and not emergency maneuvering – assume:

Assumption 2.2 (Collision-free Initial State)**.** *At t_0, no collision between any of the considered vehicles exists or is inevitable.*

Collisions can be avoided also at future times if the reference is chosen such that it does not intersect with the set of forbidden positions plus a safety margin $\Delta p\in\mathbb{R}^2$ as follows:

$$\bar{C}_{\text{pos}}^{(i)}\bar{x}_i(t)\notin\mathcal{F}_i(t)\oplus\left\{\Delta p|\ ||\Delta p||_\infty\leq k_{\text{pos}}\right\}, \tag{2.6}$$

where $\oplus$ denotes the Minkowski sum. Then, $\chi_i(t)\notin\mathcal{F}_i(t)\ \forall t\in(t_0,t_{\text{plan}}]$.

Next, in order to account for state and input constraints $\mathcal{X}_i$ and $\mathcal{U}_i$, assume further:

Assumption 2.3 (Admissible Reference Trajectory)**.** *A set $\mathcal{C}_{\bar{x}}$ is given such that:*

$$\begin{aligned}\bar{x}_i(t)\in\mathcal{C}_{\bar{x}}\ \forall t\in[t_0,t_{plan}]\Rightarrow&\\ &\chi_i(t;t_0,t_{plan},\bar{x}_i(\cdot),\nu_i(\cdot))\in\mathcal{X}_i,\\ &k_i\left(\chi_i(t;t_0,t_{plan},\bar{x}_i(\cdot),\nu_i(\cdot)),\bar{x}_i(t)\right)\in\mathcal{U}_i\\ \forall\nu_i(t)\in\mathcal{V}_i,\ \forall t\in[t_0,t_{plan}].&\end{aligned}$$

A justification for this assumption along with a procedure to design a controller k_i is given in Part II. In summary, these two assumptions allow to directly impose the constraints on the reference trajectory, which is a tremendous advantage in terms of computational efficiency. The less efficient alternative would be the following: first, a reference trajectory is computed, then, its effect on the closed-loop dynamics is predicted, and after that, the constraint compliance is considered.

It must be noted that (2.6) introduces a dependency of $\bar{x}_i$ on $\mathcal{F}_i$ and therefore on the behavior of other vehicles. The problem of computing a reference which fulfils these requirements is considered here in a *cooperative* setting, where vehicles within the cooperating group $\mathcal{C}$ are willing and able to communicate information with other vehicles in $\mathcal{C}$ and to adapt their behavior. The behavior of a cooperating group as well as the impact of non-cooperating neighbors is modeled by a so-called *maneuver* as will be detailed in the following. Assume in this part that the assignment of vehicles to the groups $\mathcal{C}$ and $\mathcal{N}$ as well as the selection of a maneuver have been made. An exhaustive treatment of the design of the corresponding decision algorithms is a topic on its own and, focussing on the interaction between high-level controller and trajectory planner, beyond the scope of this thesis. Nonetheless, an example procedure is given in Section 4.2. The following section gives a review of related work on the planning problem, followed by the introduction of basic concepts which will be used in the proposed approach to be outlined afterwards.

2.1. Related Work on Trajectory Planning

Different notions of the planning problem exist in the literature: if the objective is only to compute a sequence of position coordinates without temporal information – a so-called *path* –, the corresponding problem is often referred to as *path planning*. If temporal information is assigned to the position coordinates, a *trajectory* results, and its computation is termed *trajectory planning*. In the past decades, a large number of approaches to either problem has been proposed in stationary and mobile robotics, aerospace or marine applications, and on-road autonomous driving or racing. This diversity and the correspondingly large number of approaches – consider only the recent survey articles just for autonomous driving [60, 115, 61, 31] – puts an exhaustive review beyond the scope of this thesis. Rather, a sketch of general developments as well as a discussion of approaches relevant for the following chapters is provided.

A related problem exists in control theory, where the computation of reference trajectories is a common task. Often, it is cast as an optimal control problem, which can sometimes be solved analytically [29], but usually requires specialized numerical methods, cf. [20] for a general overview or [93] for an application in an automotive context. If applied in online operation, these methods are typically realized according to the receding-horizon principle, where problems are solved repeatedly for a certain time horizon, e.g. [163]. Solutions are often computed based on nonlinear, non-convex optimization, which is computationally demanding and makes an analysis of convergence difficult. Analytical solutions, on the other hand – even though sometimes helpful [142, 149] – typically do not fulfill the requirements of autonomous driving such as compliance with state and input constraints. Especially state constraints for obstacle avoidance are a major contributor to the problem's complexity, as these make the space in which planning is conducted non-convex. These reasons have motivated the development of specialized algorithms for the trajectory planning problem in robotics.

Gridding-based methods

A very popular class of algorithms generates plans by connecting a finite number of points in an agent's (i.e., a robot, ship, UAV, autonomous vehicle,...) state space. These points are taken from a set which has been specified during the design process, for example as a (not necessarily equidistant) grid in the state space, omitting the positions of obstacles. Early approaches typically did not account for the dynamics of the controlled vehicle and considered only its position. In order to extract a path from a set of points in position space, these approaches relied on graph-searching procedures such as Dijkstra's algorithm or the A*-algorithm, cf. [88] for an introduction. If a trajectory rather than a path is desired, the so-called path-velocity-decomposition [77] can be applied, which assigns a velocity profile to a given path. However, depending on the shape of the path, the resulting profile may not be compatible with an agent's actual dynamics.

As the run-time of graph searching algorithms depends (for a fixed dimension of the search space) on the number of points to be considered, the general desire is to reduce the

number of points, while still obtaining results of sufficient quality. The need to achieve this online has motivated the development of randomized algorithms such as probabilistic road maps (PRM) or rapidly-exploring random trees (RRT), which provide probabilistic guarantees regarding their ability to find a solution as well as the quality of a solution.

More recent sampling-based algorithms such as RRT* or Monte Carlo trees [71, 84] are able to account for dynamics and have been successfully applied in practical experiments in a vehicle context [86]. Nonetheless, gridding-based methods suffer from the curse of dimensionality, as for higher dimensions, it gets increasingly difficult to sufficiently cover the state space by samples and to search over them. This is especially problematic in the context of cooperative autonomous driving, where the state space dimension is higher than when considering only a single vehicle. At the same time, especially the strength of sampling-based methods of being able to navigate cluttered environments cannot fully unfold in highly-structured on-road traffic.

Learning-based Approaches

After early applications to autonomous driving, e.g. [119], and a general interest in the context of dynamic programming [18], learning-based methods did not play a major role in the planning community until a recent renewal of interest in the wake of the deep-learning trend. A wide variety of different approaches exists, for example relying on supervised learning, where training data is obtained from human experience [130, 90, 141] or numerical optimal control [104]. If no training data is available, unsupervised learning techniques such as reinforcement learning can be employed [91], which tries to find solutions on its own.

These methods offer the advantage to not require a deep understanding of the problem in order to return somewhat useful results, and aspects such as nonlinear dynamics, which are problematic for more classical approaches, do not necessarily pose a challenge. In addition, it seems possible to capture aspects which are difficult to describe by more classical methods, such as driving styles. Also, online computation times are negligible for the planning itself, while the learning process – mainly carried out offline, but potentially continued online – can be computationally very demanding.

However, just as learning-based methods in general, these approaches face the conceptual challenge of being currently unable to give guarantees regarding the outputs resulting for inputs which have not been considered during the learning process. In particular, it is not – at least not yet – possible to guarantee constraint compliance and obstacle avoidance, which is problematic in a safety-critical application like the one considered in this thesis.

Exploiting structure: Maneuvers, Motion primitives, and Homotopies

The complexity of the planning task has motivated the development of approaches which seek to reduce it by exploiting problem structure, often relying on so-called *motion primitives*, *maneuvers*, or *homotopies* as reviewed below. Even though many different realizations exist, the general idea underlying such approaches is to reduce the size of the search

space. This had already motivated probabilistic approaches such as RRT/PRM mentioned above, however possibly leading to a reduced performance. Methods which rely on maneuvers/motion primitives/homotopies, in contrast, suppose that in a structured environment, control based on a small set of admissible decisions specified in advance allows to achieve almost equivalent results as approaches without such a reduced decision set. While this reduces the computational effort required during online operation, the design task carried out offline is typically more complicated, as it now requires to devise a reduced decision set. As outlined in the sequel, another advantage of such offline computations is to enable the use of set-based methods, cf. [23] and Chapter 3. Thus, it becomes possible to give guarantees regarding properties of the result, while the complexity of the design task further increases.

An early use of a motion primitive concept targeting a stationary robot is reported in [9], yet without detailing its realization. Later, the term *maneuver* was formally defined [55, 54] in an approach to the guidance and control of nonholonomic mechanical systems which exhibit symmetries and relative equilibria (trajectories of constant inputs and velocities) – two requirements which are reported there to be fulfilled by most vehicles, being equivalent to invariance to translation and rotations around the vertical axis. For such systems, a *maneuver library* is devised which contains so-called *trajectory primitives*: classes of trajectories which are equivalent under mentioned translations and rotations. Two types of primitives are considered in [55, 54]: *trim trajectories*, i.e., relative equilibria, and *maneuvers*, which encode finite-time transitions between these. The interaction of these primitives is modeled by a hybrid automaton termed the *maneuver automaton*. Two major tasks are addressed: first, the design of maneuvers, which can be solved in different ways, e.g. by numerical optimal control or relying on the flatness property of a system [80]. Second, safe transition between the discrete states of the hybrid automaton must be guaranteed, for which set-based concepts are employed. On this basis, the planning problem is cast as an optimal control problem and solved offline by approximate dynamic programming [18]. As the maneuvers are defined for the system model itself (and not a simplified planning model), all generated reference trajectories are executable and can be robustly controlled, given a corresponding controller. However, in order to account for moving obstacles, it is required to combine the approach with another approach for the selection of a collision-free maneuver, which can be difficult.

A different approach to structuring is considered in [135] and relies on hybrid automata, cf. [98]. Each discrete state of such an automaton in [135] models different continuous dynamics of a controlled system; different states can be enabled by a supervisory controller, and safety of transitions is envisioned (though not realized) to rely on reachability analysis. The approach is applied to an overtaking example with three involved vehicles. A conceptually similar approach is pursued by [37, 59], which rely on hybrid system models to address the problem of automatic aerial refueling of UAVs and the aerobatic flight of an autonomous quadrotor, respectively. There, a maneuver consists of a sequence of discrete states, between which a transition is only admissible if the system's continuous state is within a certain set, which is computed using the Level Set toolbox [108]. While the approach is able to account for disturbances and can guarantee the successful completion

of a maneuver, it is limited to low-dimensional systems by the computational burden of the set computations. Also, no obstacles are considered.

The approach in [69], similar in spirit to [137, 100], considers a motion primitive as the set of possible trajectories of the closed-loop dynamics of an agent and a tracking controller under bounded disturbances and a given reference input. Trajectories result from concatenation of several motion primitives by a graph-searching algorithm. As a primitive describes all possible behaviors of the controlled system, guarantees for collision avoidance can be given. To ensure conditions under which a concatenation is admissible is a major challenge in this kind of approach.

Sometimes, similar concepts are found under the notion of *homotopy*, for example to limit the search space of gridding-based planners [125, 157]. The strand of work [81, 82, 83] in contrast focuses on continuous solutions, essentially relying on interpolation of pre-computed trajectories to ensure collision avoidance. In the context of autonomous driving, the work [16] exploits the structure inherent in on-road traffic in order to compute trajectories. There, structure is understood as the existence of a limited number of homotopies (also referred to as maneuver variants) when planning in a given situation. Extensions of the maneuver notion to cooperative driving can be found in [101, 46].

Approaches based on Mixed-Integer Linear or Quadratic Programming

Originally mainly applied in operations research problems [22], mixed-integer linear programming (MILP) and mixed-integer quadratic programming (MIQP) [153] have gained increasing interest in the control community due to their ability to encode logical implications, which in turn allows to encode switching processes, changes in a system's dynamics, certain types of non-convex constraints, and so on. After first applications in the modeling of hybrid systems such as piecewise-affine systems or hybrid automata [98], MILP/MIQP were quickly discovered to allow for efficient solution of the planning problem. After first applications in aerospace [127, 126, 85] or robotics [117, 35], MILP/MIQP have found application to trajectory planning in autonomous driving [112, 120, 45, 46, 30, 109]. This growing interest in various application domains is complemented by further advances on the system-theoretical side [95, 103].

A major reason for the wide use of MILP/MIQP is the observation that, despite adverse theoretical run-time bounds, state-of-the-art solvers such as those in [72, 63] perform very well in practice and are guaranteed to find a globally optimal solution. Nonetheless, much research effort is devoted to a reduction of problem complexity by exploitation of problem-specific structure. Often, this either aims at a reduction of complexity in terms of the number of binary variables required for a problem formulation or at a reduction of the number of possible value combinations. This can be facilitated by imposing additional constraints, for example on the temporal sequence of variables [143, 46].

The following chapters present an approach which is both able to reduce the problem complexity and to characterize the feasible domain of the resulting MIQP – an aspect which is crucial for safe and computationally efficient operation of autonomous vehicles.

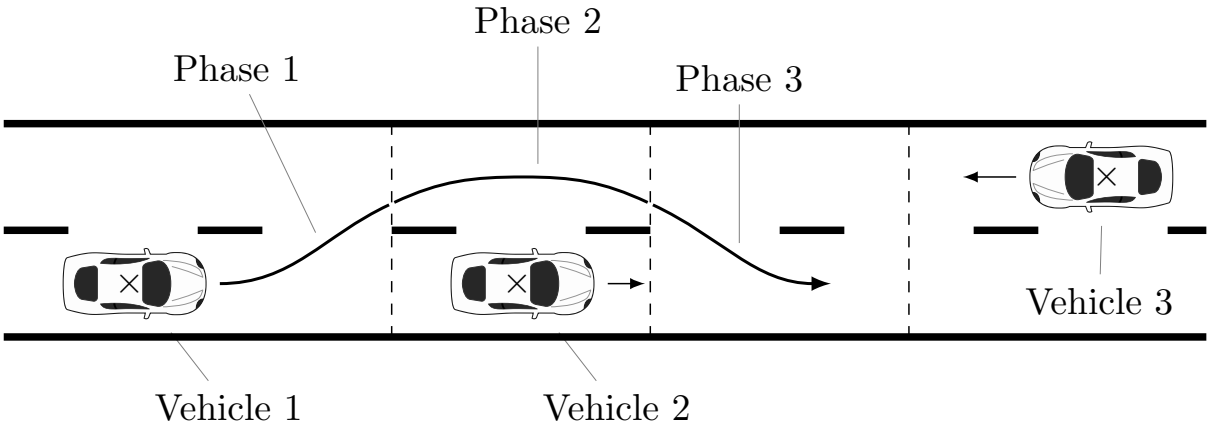


Figure 2.3.: Partition of an example maneuver into different phases.

2.2. Basic Concepts

The basic approach of this part is to generate reference trajectories for the tracking controller of a vehicle as outputs of a dynamical system in such a way that constraints on the resulting states and control inputs of the controlled vehicles $i \in \mathcal{C}$ hold and no collisions occur. While generally a difficult problem due to the presence of non-convex, time-varying constraints, the planning problem also exhibits a certain structure: many behaviors in onroad traffic are qualitatively similar in so far as that they exhibit a similar temporal order of events. Consider, for example, the overtaking maneuver in Fig. 2.3, which can be divided into three phases: in the first phase, the overtaking Vehicle 1 is behind Vehicle 2, in the second phase, it is next to it, and in the third phase, it has overtaken, yet not necessarily attained a position on the target right lane. Once this has been accomplished before the oncoming Vehicle 3 is too close, the maneuver is completed. The approach proposed in this section, referred to as *maneuver concept*, seeks to exploit such structure in order to enable computationally efficient planning.

2.2.1. Maneuvers

The following definition, modifying that in [134], is the basis of the maneuver concept:

Definition 2.2 (Syntax of a Hybrid Automaton)**.** *A hybrid automaton* `HA` *is a tuple* $(\mathbb{T}, \mathcal{Q}, q_0, q_\mathrm{T}, \mathrm{inv}, \mathcal{U}, \mathcal{W}, \mathcal{X}_0, \mathcal{X}_\mathrm{T}, \Theta, g, f)$*, defining:*

- *a* time domain $\mathbb{T} := \{t_k := t_0 + k \cdot T_s | k \in \mathbb{N}\}$ *with* sampling time $T_s > 0$ *and initial time* $t_0 \geq 0$*,*
- *a finite set* $\mathcal{Q}$ *of* phases *with initial phase* $q_0 \in \mathcal{Q}$ *and target phase* $q_\mathrm{T} \in \mathcal{Q}$*,*
- *a state vector* $x \in \mathbb{R}^{n_x}$*, an input vector* $u \in \mathcal{U} \subset \mathbb{R}^{n_u}$*, and a disturbance vector* $w \in \mathbb{R}^{n_w}$ *such that* $\begin{bmatrix} x \\ w \end{bmatrix} \in \mathcal{W}$*, with compact constraint sets* $\mathcal{U}$ *and* $\mathcal{W}$*,*

- *an affine* flow *function* $f : \mathbb{R}^{n_x} \times \mathcal{U} \times \mathtt{proj}_w \mathcal{W} \to \mathbb{R}^{n_x}$*:*

$$f(x, u, w) := Ax + Bu + B_1 w + a, \tag{2.7}$$

  with matrices $A \in \mathbb{R}^{n_x \times n_x}$*,* $B \in \mathbb{R}^{n_x \times n_u}$*,* $B_1 \in \mathbb{R}^{n_x \times n_w}$*, and* $a \in \mathbb{R}^{n_x}$*,*

- *a map* $\mathrm{inv} : \mathcal{Q} \to 2^{\mathbb{R}^{n_x}}$ *called* invariant*, assigning to each phase* $q \in \mathcal{Q}$ *a set* $\mathrm{inv}(q) \subseteq \mathbb{R}^{n_x}$ *in which the state* x *may evolve over time,*

- *the sets* $\mathcal{X}_0 \subseteq \mathrm{inv}(q_0)$ *and* $\mathcal{X}_\mathrm{T} \subseteq \mathrm{inv}(q_\mathrm{T})$ *of initial and target states, respectively,*

- *a set* $\mathcal{T} = \{\theta_1, \theta_2, \dots\}$ *of* discrete transitions $\theta \in \mathcal{Q} \times \mathcal{Q}$ *and a map* $\Theta : \mathcal{Q} \to 2^{\mathcal{T}}$*, which assigns to each phase* q *the set* $\Theta(q) = \left\{\theta = (q^-, q) | \theta \in \mathcal{T}\right\}$ *of admissible* discrete transitions *leading to it, where in particular a sequence of transitions from* q_0 *to* q_T *must exist,*

- *a* guard *map* $g : \Theta(q) \to 2^{\mathbb{R}^{n_x}}$ *which assigns a guard set* $g(\theta) \subseteq \mathbb{R}^{n_x}$ *to each transition* $\theta = (q^-, q) \in \Theta(q)$*, with* $g(\theta) \subseteq \mathrm{inv}(q^-) \cap \mathrm{inv}(q) \neq \emptyset$*, where no two guard sets intersect by design,*

- *a linear,* n*-times continuously differentiable output function, mapping state* x *and input* u *to a reference value* $\bar{x}$ *(with* n *as in Definition 2.1):*

$$\bar{x} = h(x, u) := Cx + Du. \tag{2.8}$$

Note that the requirement regarding the intersection of guards and invariants does not arise in standard definitions of hybrid automata, e.g. [134], owing to the use of so-called *jump functions*, which are not employed in this thesis, however.

An example for a hybrid automaton is shown in Fig. 2.4. Often [98], the evolution of a hybrid system is considered on a continuous time domain. However, digital and especially optimization-based control cannot be realized in continuous time, which motivates the use of a discrete time domain $\mathbb{T}$ in Definition 2.2. The temporal evolution of states and phases of a hybrid automaton occurs according to Definition 2.3:

Definition 2.3 (Admissible Run of `HA`)**.** *Defining sequences of phases and states:*

$$\begin{aligned} q_{j+1}(\cdot) &:= \big(q(t_0),\ \ q(t_1),\ \ \dots,\ \ q(t_{j-2}),\ \ q(t_j)\big) \\ x_{j+1}(\cdot) &:= \big(x(t_0),\ \ x(t_1),\ \ \dots,\ \ x(t_{j-2}),\ \ x(t_j)\big) \end{aligned}$$

as well as inputs and disturbances:

$$\begin{aligned} u_j(\cdot) &:= \big(u(t_0),\ \ u(t_1),\ \ \dots,\ \ u(t_{j-2}),\ \ u(t_{j-1})\big) \\ w_j(\cdot) &:= \big(w(t_0),\ \ w(t_1),\ \ \dots,\ \ w(t_{j-2}),\ \ w(t_{j-1})\big)\,, \end{aligned}$$

an admissible run *of the automaton is a sequence* $\begin{pmatrix} q_{j+1}(\cdot) \\ x_{j+1}(\cdot) \end{pmatrix}$ *which fulfills the following: Starting in* $\mathcal{X}_0$*, the continuous state* x *evolves according to the flow function* (2.7)*, depending on the state, input, and disturbance values. The input* u *is chosen such that* $u \in \mathcal{U}$*, but*

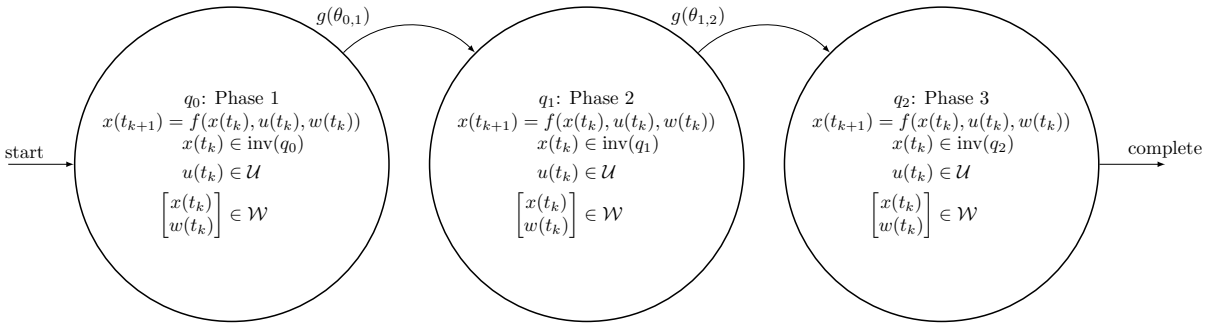


Figure 2.4.: Example hybrid automaton corresponding to an overtaking maneuver.

without knowledge of the disturbance w, for which $\begin{bmatrix} x(t_k) \\ w(t_k) \end{bmatrix} \in \mathcal{W}$ *holds for all* $0 \leq k \leq j+1$. *Upon entry of x into a guard set $g(\theta) = (q_0, q_i)$, an immediate transition from q_0 to phase q_i is enforced, where the evolution of x continues with $x \in \mathrm{inv}(q_i)$ until another guard set is entered, and so forth. Once x enters the terminal set $\mathcal{X}_\mathrm{T}$, its evolution stops.*

From the state x, reference trajectories for the involved cooperating vehicles will be derived in the sequel. Even though the dynamics (2.1) of a vehicle is typically nonlinear, an affine flow function is chosen in Definition 2.2 to describe the evolution of x because it allows for efficient computations. As the automaton is not involved in direct actuation of a vehicle, but is only used for the computation of reference trajectories for the low-level tracking controller, this mismatch is not per se a problem. The challenges associated with this choice are the topic of Part II. Also for reasons of computational efficiency, let all sets in Definition 2.2 be polyhedral. A hybrid automaton based on this definition is a central element of a maneuver:

Definition 2.4 (Maneuver)**.** *A maneuver is a tuple* $\mathtt{M} = \big(\mathcal{C}, \mathcal{N}, H_{plan}, \mathtt{HA}\big)$*, consisting of a set $\mathcal{C}$ of cooperating vehicles, their non-cooperating leading neighbor vehicles $\mathcal{N}$, a planning horizon H_{plan}, and a hybrid automaton* HA *according to Definition 2.2.*

Apart from maneuvers, *controllable sets* are another important concept, which is introduced next.

2.2.2. Controllable Sets

In the sequel, it will be important to characterize (sets of) states from which an admissible run of a given hybrid automaton exists which ends in the target set $\mathcal{X}_\mathrm{T}$ in target phase q_T despite the influence of all admissible disturbances. Adapting [24, p. 184], [34, p. 42], or [79], this leads to the following concept:

Definition 2.5 (j-step robust controllable set of a hybrid automaton)**.** *Given a hybrid automaton as in Definition 2.2 and a target set $\mathcal{X}_\mathrm{T}$ in the invariant* $\mathrm{inv}(q_\mathrm{T})$ *of a target phase q_T, the set*

$$\mathcal{K}_j = \left\{ \mathcal{K}_j^{(1)}, \mathcal{K}_j^{(2)}, \dots \right\} \tag{2.9}$$

contains sets $\mathcal{K}_j^{(i)} \subseteq \mathcal{Q} \times \mathbb{R}^{n_x}$, $i = 1, 2, \ldots, |\mathcal{K}_j|$, *of states* $x \in \mathbb{R}^{n_x}$ *and phases* $q \in \mathcal{Q}$ *which can be robustly (i.e., for all considered actions of the disturbance) steered to the target set* $\mathcal{X}_\mathrm{T}$ *in the target phase* q_T *within* j *steps in* $\mathbb{T}$. *Its computation bases on the operator* $\mathtt{pre} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$:

$$\mathtt{pre}(\mathcal{X}_{start}, \mathcal{X}_{end}) = \left\{ x \in \mathcal{X}_{start} | \exists u \in \mathcal{U} : Ax + Bu + B_1 w \in \mathcal{X}_{end} \; \forall \; w : \begin{bmatrix} x \\ w \end{bmatrix} \in \mathcal{W} \right\}.$$

This operator is the basis for the operator $\mathtt{pre}_c : \mathcal{Q} \times \mathbb{R}^{n_x} \to \mathcal{Q} \times \mathbb{R}^{n_x}$:

$$\mathtt{pre}_c \left(\mathcal{K}_{j-1}^{(i)} \right) = \{q\} \times \mathtt{pre} \left(\mathrm{inv}\,(q)\,, \mathtt{proj}_x \mathcal{K}_{j-1}^{(i)} \right), \text{ with } q \text{ such that } \exists x : \begin{bmatrix} q \\ x \end{bmatrix} \in \mathcal{K}_{j-1}^{(i)}$$

and the operator $\mathtt{pre}_d : \mathcal{Q} \times \mathcal{Q} \times \mathbb{R}^{n_x} \to \mathcal{Q} \times \mathbb{R}^{n_x}$:

$$\mathtt{pre}_d \left(\theta = (q^-, q^+), \mathcal{K}_{j-1}^{(i)} \right) = \{q^-\} \times \mathtt{pre} \left(\mathrm{inv}\left(q^-\right), \left(\mathtt{proj}_x \mathcal{K}_{j-1}^{(i)} \right) \cap g(\theta) \right),$$

where q^+ *is such that* $\exists x : \begin{bmatrix} q^+ \\ x \end{bmatrix} \in \mathcal{K}_{j-1}^{(i)}$. *Given these operators, a controllable set is defined recursively, starting with* $\mathcal{K}_0 := \{\{q_\mathrm{T}\} \times \mathcal{X}_\mathrm{T}\}$:

$$\mathcal{K}_j = \bigcup_{i=1}^{|\mathcal{K}_{j-1}|} \mathtt{pre}_c \left(\mathcal{K}_{j-1}^{(i)} \right) \cup \left(\bigcup_{\theta \in \Theta(q)} \mathtt{pre}_d \left(\theta, \mathcal{K}_{j-1}^{(i)} \right) \right), \text{ with } q \text{ such that } \exists x : \begin{bmatrix} q \\ x \end{bmatrix} \in \mathcal{K}_{j-1}^{(i)}.$$

In Definition 2.5, pre returns all states from a specified set of initial states which can be robustly steered to a given target set in one step under affine dynamics and constraints on control and disturbance inputs. In the context of a hybrid automaton, the operator $\mathtt{pre}_\mathrm{c}$ makes use of pre to determine all controllable states within the same invariant as the target set, while $\mathtt{pre}_\mathrm{d}$ focuses on controllable states from other invariants with admissible transitions to that of the target set. The operator pre can generate several (potentially even overlapping) predecessor sets, depending on the number of discrete transitions. Thus, $\mathcal{K}_j$ is a *set* whose elements are denoted by $\mathcal{K}_j^{(i)}$ and whose union is non-convex.

According to [24], for affine dynamics (2.7) and polyhedral constraint sets of the form $\mathcal{U} = \{u | A_u u \leq b_u\}$, $\mathcal{X}_\mathrm{end} = \{x | A_\mathrm{end} x \leq b_\mathrm{end}\}$, $\mathcal{X}_\mathrm{start} = \{x | A_\mathrm{start} x \leq b_\mathrm{start}\}$, $\mathtt{pre}(\mathcal{X}_\mathrm{start}, \mathcal{X}_\mathrm{end})$ can be computed as follows: :

$$\mathtt{pre}(\mathcal{X}_\mathrm{start}, \mathcal{X}_\mathrm{end}) =$$
$$\mathtt{proj}_x \left\{ \begin{bmatrix} x \\ u \end{bmatrix} | \begin{bmatrix} A_\mathrm{end} A & A_\mathrm{end} B \\ 0 & A_u \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + \begin{bmatrix} A_\mathrm{end} B_1 \\ 0 \end{bmatrix} w \leq \begin{bmatrix} b_\mathrm{end} \\ b_u \end{bmatrix} \; \forall w : \begin{bmatrix} x \\ w \end{bmatrix} \in \mathcal{W} \right\} \cap \mathcal{X}_\mathrm{start},$$

which in turn is equivalent to:

$$\mathtt{proj}_x \left\{ \begin{bmatrix} x \\ u \end{bmatrix} | \begin{bmatrix} A_\mathrm{end} A & A_\mathrm{end} B \\ 0 & A_u \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \leq \begin{bmatrix} b_\mathrm{end} \\ b_u \end{bmatrix} - \tilde{b} \right\} \cap \mathcal{X}_\mathrm{start}, \tag{2.10}$$

where the l-th entry of $\tilde{b}$ results from:

$$\tilde{b}_{[l]} := \max_{w} A_{\mathrm{end}[l,:]} B_1 w \quad \text{s.t.} \quad \begin{bmatrix} x \\ w \end{bmatrix} \in \mathcal{W}. \tag{2.11}$$

It is important to note that – contrasting the standard setting considered in [24] – the coupling constraint $\mathcal{W}$ introduces a dependency of $\tilde{b}$ on x into (2.10), which requires special attention as detailed in Sec. 2.3.1.

Another important concept in the following is the notion of robust control invariant sets [23]:

Definition 2.6 (Robust Control Invariant Set)**.** *A set $\mathcal{O} \subset \mathbb{R}^n$ is robust control invariant if and only if:*

$$x(t_k) \in \mathcal{O} \Rightarrow \exists u \in \mathcal{U} : f(x(t_k), u(t_k), w(t_k)) \in \mathcal{O} \ \forall \ w : \begin{bmatrix} w(t_k) \\ x(t_k) \end{bmatrix} \in \mathcal{W}.$$

These concepts will be employed in the following to model maneuvers and to approach the planning problem on that basis.

2.3. Modeling of Maneuvers

In order to model a set of qualitatively similar behaviors of a group of vehicles based on the maneuver definition in Definition 2.4, proper choice of the elements of the corresponding hybrid automaton is required. Starting point in the modeling process is the choice of the sampling time T_s and of a desired degree n of continuity of the reference trajectory, which eventually determines the number of entries in the vector (2.4).

The next step considers the interaction and cooperation between all vehicles. Perceiving each vehicle as a subsystem, the entity of all vehicles forms a *distributed system*. In the theory of distributed systems, three interaction types are generally considered [74]: through common objectives (typically represented by a cost function), dynamics, and constraints. The definition of a maneuver targets the description of *admissible* behaviors, without judging about their quality. Therefore, the definition of a cost function is irrelevant at this point. Similarly, vehicles do not impact the dynamics (2.1) of each other, which leaves interaction through constraints:

As discussed earlier, this interaction is via the time-varying set $\mathcal{F}_i(t)$ of positions forbidden for vehicle i. Such restrictions not only apply at one point in time, but throughout the entire duration of a maneuver. Therefore, it is required to consider the temporal evolution of the vehicles' positions, which is modeled by the flow function f for all cooperating vehicles in $\mathcal{C}$ as well as the neighboring vehicles in $\mathcal{N}$ together. Therein, the inputs of the vehicles in $\mathcal{C}$ are associated with the control input u, while the inputs of those vehicles in $\mathcal{N}$ correspond to the disturbance w. States and inputs are then related to the reference trajectory by choice of the output function h according to (2.8). At this point, the difference between the flow function and the model (2.1) of the dynamics of vehicle i must be

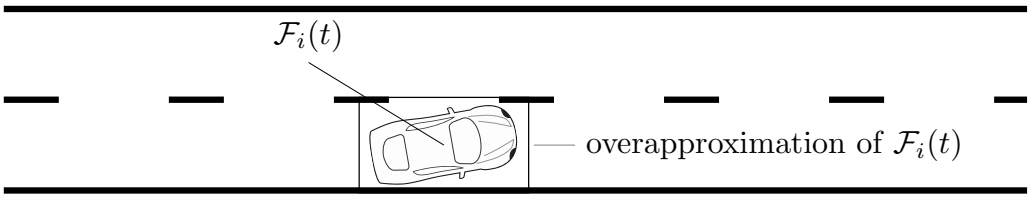


Figure 2.5.: Example for a lane-aligned, rectangular overapproximation of $\mathcal{F}_i(t)$.

emphasized: first, the flow function considers *multiple* vehicles. Second, while the typically nonlinear dynamics (2.1) models position dynamics due to (2.3), the dynamics (2.7) only models the evolution of *reference* positions; an affine system is chosen there for reasons of computational efficiency.

In on-road traffic, the interaction between vehicles is constrained by the existence of lanes, which allows for the separate consideration of longitudinal and lateral interaction between vehicles. This corresponds to abstracting from a set $\mathcal{F}_i(t)$ and to consider a lane-aligned, rectangular overapproximation instead, cf. Figure 2.5. For example, during lane keeping, only the distance to a leading vehicle constrains the action of a following vehicle, while during overtaking or lane changing, the distance to vehicles next to each involved vehicle and both their longitudinal and lateral motion must be considered. Assume:

Assumption 2.4 (Interaction between Vehicles)**.** *During the execution of a maneuver, non-cooperating vehicles only impact cooperating ones through their longitudinal motion. Lateral interaction is not admissible.*

In particular, this assumption ensures that no non-cooperating neighboring vehicle leaves its lane in the direction of an occupied neighboring lane, e.g. by merging into the safety gap between two cooperating vehicles.

In order to model these rules for interaction between vehicles, the general approach in the following is to define a hybrid automaton `HA` for all vehicles in $\mathcal{C}$ and $\mathcal{N}$ combined, where the temporal evolution of a certain cooperative behavior to be modeled is divided into sequential phases $\mathcal{Q}$. To each phase q, an invariant $\text{inv}(q)$ is associated, specifying admissible states x in that phase, while the discrete transitions $\Theta(q)$ and the corresponding guard maps $g(\theta)$, $\theta \in \Theta(q)$, define admissible transitions from one phase to another.

These choices must be made subject to two major requirements: First, the states and inputs must be constrained such that, based on Assumption 2.3, the reference trajectory is suitable for tracking purposes as discussed in Part II. Second, the invariants and transitions must ensure that no vehicle in $\mathcal{C}$ causes a collision. In the following, longitudinal and lateral collisions are considered separately as discussed in Section 2.3.1 and Section 2.3.2, respectively.

2.3.1. Longitudinal Collision Avoidance

This section aims at the derivation of constraints which, if considered in the definition of the invariants of a hybrid automaton when modeling a maneuver, ensure longitudinal

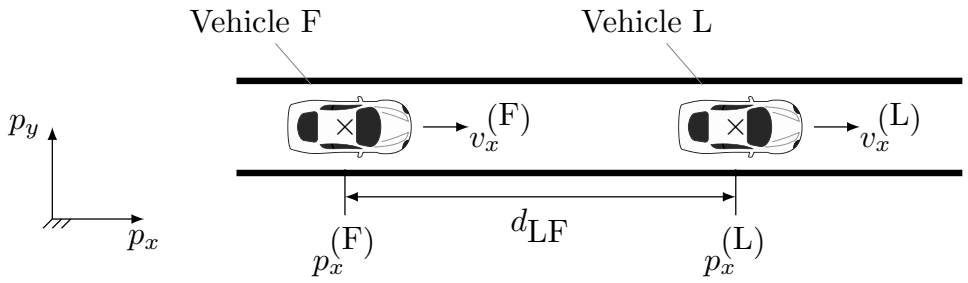


Figure 2.6.: Longitudinal distance between two vehicles.

collision avoidance between all considered vehicles. To that end, the constellation shown in Figure 2.6 is considered, where a vehicle L is driving ahead of a vehicle F. Denoting the longitudinal positions of these vehicles by $p_x^{(\mathrm{L})}$ and $p_x^{(\mathrm{F})}$, respectively, and the relative distance by $d_{\mathrm{LF}} := p_x^{(\mathrm{L})} - p_x^{(\mathrm{F})}$, a longitudinal collision between these two occurs if a negative relative velocity is maintained for too long; this results if F is driving faster than L. Because traffic rules forbid F to cause a collision by purposely tail-gating L, the following analysis focuses on the case where braking of L leads to negative relative velocity – for example in case of emergency braking.

In order to simplify the exposition of the following derivations, the vehicles' longitudinal acceleration/braking is considered as input, with identical and symmetric bounds:

$$|u_x^{(\mathrm{F})}| \leq u_{x,\max}, \quad |u_x^{(\mathrm{L})}| \leq u_{x,\max}.$$

The objective is to characterize an *invariant set* of relative distances and velocities, such that a minimum longitudinal safety distance $l_{x,\mathrm{safe}} > 0$ can be maintained by the vehicle F despite *all* acceleration/braking actions of vehicle L *for all* times. In this, it is assumed that F immediately notices when L brakes, such that there is no time delay in a reaction of F (the consideration of delays, while possible, would unnecessarily complicate the following exposition). Assume further that both F and L obey to velocity constraints which are imposed by traffic rules:

$$v_{x,\min} \leq v_x^{(\mathrm{F})} \leq v_{x,\max}, \quad v_{x,\min} \leq v_x^{(\mathrm{L})} \leq v_{x,\max}.$$

This constrains the accelerations in dependency on the velocity: if the velocity of a vehicle is extremal, it must not accelerate further beyond that velocity, but only in the opposite direction. Under these assumptions, it is irrelevant whether L is cooperating/autonomous or not because the necessary safety distance only depends on acceleration/braking capabilities of the two vehicles.

Let the dynamics of the relative position and the vehicle velocities be modeled by zero-order hold discretization with sampling time T_{s} of:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} d^{(\mathrm{LF})} \\ v_x^{(\mathrm{F})} \\ v_x^{(\mathrm{F})} \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} d^{(\mathrm{LF})} \\ v_x^{(\mathrm{F})} \\ v_x^{(\mathrm{F})} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u_x^{(\mathrm{L})} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x^{(\mathrm{F})}, \tag{2.12}$$

leading to:

$$x(t_{k+1}) = Ax(t_k) + Bu(t_k) + B_1 w(t_k), \tag{2.13}$$

with:

$$x := \begin{bmatrix} d^{(\mathrm{LF})} \\ v_x^{(\mathrm{L})} \\ v_x^{(\mathrm{F})} \end{bmatrix},\ u := u_x^{(\mathrm{F})},\ w := u_x^{(\mathrm{L})},\ A = \begin{bmatrix} 1 & T_\mathrm{s} & -T_\mathrm{s} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ B = \begin{bmatrix} -\frac{T_\mathrm{s}^2}{2} \\ 0 \\ T_\mathrm{s} \end{bmatrix},\ B_1 = \begin{bmatrix} \frac{T_\mathrm{s}^2}{2} \\ T_\mathrm{s} \\ 0 \end{bmatrix}. \tag{2.14}$$

It should be noted that this simple model actually reflects the purely longitudinal dynamics of more complex vehicle models such as the bicycle model as considered in Part II. The state is confined to the set

$$\mathcal{X} := \left\{ \begin{bmatrix} d^{(\mathrm{LF})} \\ v_x^{(\mathrm{L})} \\ v_x^{(\mathrm{F})} \end{bmatrix} | d^{(\mathrm{LF})} \geq l_{x,\mathrm{safe}}, v_{x,\min} \leq v_x^{(\mathrm{F})} \leq v_{x,\max} \right\}, \tag{2.15}$$

while the control inputs are taken from:

$$\mathcal{U} = \left\{ u_x^{(\mathrm{F})} |\ |u_x^{(\mathrm{F})}| \leq u_{x,\max} \right\}. \tag{2.16}$$

In order to model constraints on the disturbance input $u_x^{(\mathrm{L})}$, it should be noted that the interval $[v_{x,\min}, v_{x,\max}]$ is control invariant for the leading vehicle L, i.e., the vehicle can always choose $u_x^{(\mathrm{L})}$ such that it complies with the velocity constraints. Compatible inputs are encoded by:

$$\begin{aligned} \mathcal{W} := \Bigg\{ \begin{bmatrix} x \\ u_x^{(\mathrm{L})} \end{bmatrix} | \begin{bmatrix} v_x^{(\mathrm{L})} \\ u_x^{(\mathrm{L})} \end{bmatrix} \in [v_{x,\min}, v_{x,\max}] \times [-u_{x,\max}, u_{x,\max}], \\ \begin{bmatrix} v_x^{(\mathrm{L})} + T_\mathrm{s} u_x^{(\mathrm{L})} \\ u_x^{(\mathrm{L})} \end{bmatrix} \in [v_{x,\min}, v_{x,\max}] \times [-u_{x,\max}, u_{x,\max}] \Bigg\}, \end{aligned}$$

or equivalently,

$$\mathcal{W} = \left\{ \begin{bmatrix} x \\ u_x^{(\mathrm{L})} \end{bmatrix} |\ |u_x^{(\mathrm{L})}| \leq u_{x,\max}, \frac{v_{x,\min} - v_x^{(\mathrm{L})}}{T_\mathrm{s}} \leq u_x^{(\mathrm{L})} \leq \frac{v_{x,\max} - v_x^{(\mathrm{L})}}{T_\mathrm{s}}, v_{x,\min} \leq v_x \leq v_{x,\max} \right\}. \tag{2.17}$$

The objective of the remainder of the section is to compute a robust control invariant set $\mathcal{X}_\mathrm{safe}$, containing according to Definition 2.6 those states for which an admissible safety distance can be maintained despite all actions of the leading vehicle. If $v_{x,\min} = 0$, these actions comprise braking to standstill of L and consequently also F. Let

$$\mathcal{X}_\mathrm{T} = \left\{ x | d^{(\mathrm{LF})} \geq l_{x,\mathrm{safe}}, v_x^{(\mathrm{F})} = 0 \right\}. \tag{2.18}$$

Then, the computation of $\mathcal{X}_{\text{safe}}$ relies on the fact that if the following vehicle F can avoid a collision for *maximum braking* of the leading vehicle L, it can do so for all admissible accelerations. More formally:

Proposition 2.1. *The set*

$$\left\{ x \in \mathcal{X} | \exists j \in \mathbb{N}, u_j(\cdot), x_{j+1}(\cdot) : Ax_{j+1}(t_k) + Bu_j(t_k) + B_1 w_j(t_k) = x_{j+1}(t_{k+1}), \right.$$
$$x_{j+1}(t_0) = x, u_j(t_k) \in \mathcal{U}, x_{j+1}(t_j) \in \mathcal{X}_{\text{T}},$$
$$\left. w_j(t_k) = \max \left\{ -u_{x,\max}, \frac{v_{x,\min} - v_x^{(L)}(t_k)}{T_s} \right\} \forall k = 0, 1, 2, \ldots, j-1 \right\}$$

is robust control invariant under the dynamics (2.13).

Even though intuitive, Proposition 2.1 is elementary for the rest of this part and therefore proven in detail:

Proof. First, note that the target set $\mathcal{X}_{\text{T}}$ is a robust control invariant set: vehicle F is at standstill in a safe distance to the leading vehicle L, which can only increase the safety distance as it must not drive backwards. By definition, every set of states that can be robustly controlled to a robust control invariant set is also robust control invariant [23]. The general idea is to compute a sequence of j of such sets:

$$\left(\mathcal{K}_k(\mathcal{X}_{\text{T}})\right)_{k=1}^{j}$$

according to Section 2.5, which is simplified by the fact that not a hybrid automaton, but only a linear, time-invariant system is considered, such that phases and transitions do not appear in the computations. Each k-step controllable set is of the form:

$$\mathcal{K}_k = \{x | A_k x \leq b_k\}.$$

Next, it is shown that these sets are equivalent to the corresponding sets which result for maximum admissible braking of the leading vehicle. Computations are based on the recursion in Definition 2.5, requiring application of the `pre`-operator (2.10). In iteration k, `pre` reads:

$$\left\{ \begin{bmatrix} A_k \begin{bmatrix} 1 & T_{\text{s}} & -T_{\text{s}} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & A_k \begin{bmatrix} -T_{\text{s}}^2 \\ 0 \\ T_{\text{s}} \end{bmatrix} \\ 0 & A_u \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \leq b_k - \tilde{b} \right\} \cap \mathcal{X}. \tag{2.19}$$

The computation of $\tilde{b}$ requires to solve the optimization problem (2.11), where the optimal value of w depends on $A_{k[l,:]}$. These result from preceding iterations and can be explicitly computed. More specifically, given A_k such that $\operatorname{sgn} A_{k[l,1]} \leq 0$ and $\operatorname{sgn} A_{k[l,2]} \leq 0$ for each

row l, the same property holds for A_{k+1}. The transition from A_k to A_{k+1} consists of three steps: multiplication of A_k by A, projection, and intersection with $\mathcal{X}$. Clearly, based on A in (2.14), the signs of the first and second column remain unaltered for $A_k A$. The projection operation $\texttt{proj}_x$ is typically facilitated based on Fourier-Motzkin elimination [162], which is equivalent to multiplication with a matrix D of non-negative entries, such that the projection in (2.19) is equivalent to:

$$D \begin{bmatrix} A_k \begin{bmatrix} 1 & T_\mathrm{s} & -T_\mathrm{s} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & A_k \begin{bmatrix} -T_\mathrm{s}^2 \\ 0 \\ T_\mathrm{s} \end{bmatrix} \\ 0 & A_u \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \leq D \left(b_k - \tilde{b} \right),$$

Thus, the signs of the first two columns of the resulting matrix are non-positive. The intersection with the state constraints $\mathcal{X}$ only adds a negative entry to the first column, as with (2.15) in half-space representation,

$$\mathcal{X} = \left\{ x | \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix} \leq \begin{bmatrix} -l_{x,\mathrm{safe}} \\ -v_{x,\mathrm{min}} \\ v_{x,\mathrm{max}} \end{bmatrix} \right\},$$

while the added positive value in the third column is irrelevant for the optimization of (2.11).

The beginning of the induction is the target set (2.18). Re-writing equality constraints into two inequality constraints then leads to:

$$A_0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix}, \; b_0 = \begin{bmatrix} -l_{x,\mathrm{safe}} \\ 0 \\ 0 \end{bmatrix},$$

and therefore,

$$A_0 A = \begin{bmatrix} -1 & -T_\mathrm{s} & T_\mathrm{s} \\ 0 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Consequently,

$$A_{k[l,:]} B_1 \leq 0$$

for every row l and any step k. Combined with (2.11), this implies that

$$\arg\max A_{k[l,:]} B_1 w = \max \left\{ -u_{x,\mathrm{max}}, \frac{v_{x,\mathrm{min}} - v_x^{(\mathrm{L})}}{T_\mathrm{s}} \right\}$$

for every row l and any k, completing the proof.

□

While a valuable tool in the proof of Proposition 2.1, two major reasons prevent the application of the recursion in Definition 2.5 to the actual computation: first, it would, depending on the chosen number of steps, return a large number of polytopes, making it computationally expensive. Second – and even more severe – the state-dependent disturbance prevents the use of the standard tools from computational geometry for computation of the projection operation that is required to implement the `pre`-operator, cf. (2.10). On the other hand, it is possible to compute a robust control invariant set analytically:

Proposition 2.2 (Robust control invariant set $\bar{\mathcal{X}}_{\text{safe}}$)**.** *The set*

$$\bar{\mathcal{X}}_{safe} = \left\{ \begin{bmatrix} d^{(LF)} \\ v_x^{(L)} \\ v_x^{(F)} \end{bmatrix} \middle| d^{(LF)} + \frac{1}{2|u_{x,\min}|} \left({v_x^{(L)}}^2 - {v_x^{(F)}}^2 \right) \geq l_{x,safe} \right\}$$

is robust control invariant for the dynamics (2.13) *and the corresponding constraints* (2.15), (2.16)*, and* (2.17)*.*

Proposition 2.2 is proven in Appendix A. While the set $\bar{\mathcal{X}}_{\text{safe}}$ is non-convex, polytopes are preferable for computational reasons. This leads to:

Proposition 2.3 (Longitudinal Collision Avoidance)**.** *Given a robust control invariant set $\bar{\mathcal{X}}_{safe}$ according to Proposition 2.2, a polytope*

$$\mathcal{X}_{safe} = \{x | A_{safe} x \leq b_{safe}\} \subset \bar{\mathcal{X}}_{safe}, \tag{2.20}$$

and a current state $x \in \mathcal{X}_{safe}$, longitudinal collision between L and F can be avoided at all future times.

Proof. Clearly, $x(t_k) \in \mathcal{X}_{\text{safe}} \Rightarrow x(t_k) \in \bar{\mathcal{X}}_{\text{safe}}$. By definition of $\bar{\mathcal{X}}_{\text{safe}}$, there exists $u^{(\text{F})}(t_k)$ such that $x(t_{k+1}) \in \bar{\mathcal{X}}_{\text{safe}} \; \forall \; u_x^{(\text{L})}$. □

With $\bar{\mathcal{X}}_{\text{safe}}$ being a three-dimensional set, it is possible to obtain the polyhedral inner approximation $\mathcal{X}_{\text{safe}}$ in (2.20) manually by visual inspection. Longitudinal collisions in a platoon of multiple vehicles can obviously be avoided if Proposition 2.3 holds for according pairs of neighboring vehicles of the platoon.

2.3.2. Lateral Collision Avoidance

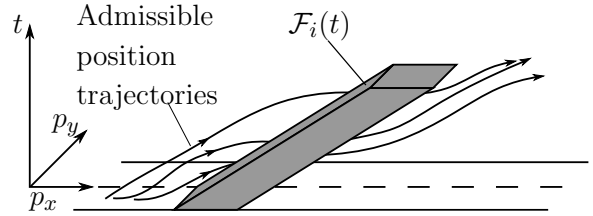
While the derivations of the previous section give conditions for longitudinal safety between any pair of vehicles and while non-cooperating vehicles will not cause a collision by lateral movement according to Assumption 2.4, the planning algorithm must actively ensure lateral collision avoidance for the cooperating vehicles. Many approaches to collision avoidance specify *forbidden regions* of positions to be avoided during the planning process. In the presence of moving obstacles, this leads to time-varying constraints as illustrated in the space-time diagram (cf. [16]) in Fig. 2.7a for an example overtaking maneuver: at

each time, the rectangular over-approximation of the space occupied by the vehicle which is being overtaken must be avoided by the overtaking vehicle.

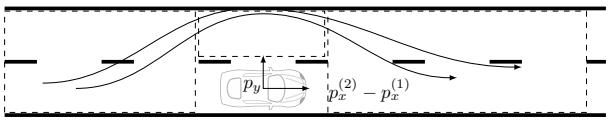
An alternative representation of the same collision avoidance constraints is shown in Fig. 2.7b, relying on *relative positions* (relative to a chosen vehicle) between the involved vehicles. While both representations of obstacle avoidance constraints encode equivalent information regarding the set of forbidden positions, the approach based on relative positions is simpler, as constraints are now time invariant. In addition – even though it would be possible to specify sets of forbidden relative positions in this approach – the notion of *admissible positions* as the complement of forbidden regions is more advantageous: If sets of states $\mathcal{X}_0$ and $\mathcal{X}_\mathrm{T}$ are defined which denote start and end of a maneuver, it is possible to partition a set of admissible positions into a sequence of convex sets which connects these initial and target sets (consider again Fig. 2.3!). This reduces the planning problem to the task of planning inside convex constraints. Each such convex set can be composed with further constraints on states, yielding the invariants of the phases of a maneuver. While the times of transitions between phases are unknown (except in the unrealistic case of known longitudinal velocities of all involved vehicles), their computation is simpler than general avoidance of forbidden sets at each time instance, as the corresponding *planning problem* as detailed in the next section is easier to solve. Further simplifications can be obtained by imposing constraints on the temporal order of phases, for example excluding a behavior where the overtaking vehicle drives next to the preceding vehicle, not overtaking, but falling back and coming up again.

2.4. Using Maneuvers for Planning

Cooperative trajectory planning, i.e., the computation of a reference trajectory for a group of cooperating vehicles, requires to account for many different constraints in high-dimensional state spaces. The maneuver concept as proposed in the previous sections is a structured way to encode these constraints. Yet, while a maneuver implicitly defines a *set* of admissible runs of the corresponding hybrid automaton `HA`, controlling a group $\mathcal{C}$ of autonomous vehicles requires to determine a *single* reference trajectory starting at a given initial state. In the following, an optimization-based procedure is used to approach this



(a) Time-varying constraints on positions.



(b) Time-invariant constraints (dashed) on relative positions for different phases.

Figure 2.7.: Constraints and example position trajectories during an overtaking maneuver with two vehicles.

challenge – not only for optimality, but to automatically ensure constraint compliance.

However, the formulation of an optimization problem is impeded by the fact that a hybrid automaton as in Definition 2.2 is an *uncertain system* due to the influence of the disturbance w, whose future values are unknown and beyond control. In this context, optimization is typically considered either in a stochastic [132] or a game-theoretic setting [12]. Both philosophies make assumptions on the future values of w in order to *predict* future values of the state x, based on the semantics as given in Definition 2.2. Predictions of a quantity $\bullet$ made for time step t_{k+j} based on information available at time t_k will be denoted by $\bullet[\![j|k]\!]$, $j \in \mathbb{N}$, as opposed to the actually resulting (but unknown at the moment) value $\bullet(t_{k+j})$.

In a stochastic setting, predictions are based on probabilistic information regarding future values of w, while the game-theoretic setting typically makes worst-case assumptions with respect to the effect of w on a cost function or constraints. Even though a first attempt towards a stochastic setting has been made in [42], obtaining stochastic information for behavior prediction of traffic participants is a field on its own [21] and beyond the scope of this thesis. Instead, a cautious, yet simple (with respect to assumptions on w) approach is chosen. Based on Assumption 2.4, it is noted that assumptions regarding the future cannot threaten safety as long as the safety distances as encoded by $\mathcal{X}_{\text{safe}}$ in Section 2.3.1 are maintained at any time. In the proof of Proposition 2.1, this requirement was shown to be equivalent to proper reaction to maximum braking of non-cooperating leading vehicles. Thus, preparing for the worst-case acceleration (in the sense of constraint compliance), i.e., letting:

$$w_{[i]}[\![j|k]\!] = \max\left\{-u_{x,\max}, \frac{v_{x,\min} - v_x^{(i)}[\![j|k]\!]}{T_\text{s}}\right\} \tag{2.21}$$

for all vehicles $i \in \mathcal{N}$ guarantees the ability to avoid longitudinal collisions. Because future values of $v_x^{(i)}$ for these vehicles only depend on their initial velocity and w, the maximum in (2.21) can be computed. The result is a behavior where a vehicle $i \in \mathcal{N}$ brakes as hard as possible until $v_{x,\min}$ is reached.

This insight allows to replace the unknown disturbance in (2.7) by a known, time-varying affine term $B_1 w[\![j|k]\!]$. Combining these terms in a sequence, where the jth element reads:

$$\tilde{a}(j) := a + B_1 w[\![j|k]\!], \tag{2.22}$$

allows to replace the flow function (2.12) by undisturbed dynamics, parametric in $\tilde{a}$:

$$\tilde{f}(x, u, \tilde{a}(j)) = Ax + Bu + \tilde{a}(j). \tag{2.23}$$

Then, the basis for computationally efficient planning is the well-known fact that a hybrid automaton as given by Definition 2.2 can be encoded as a system of affine inequalities [15]:

$$E_1 x[\![j|k]\!] + E_2 x[\![j+1|k]\!] + E_3 \beta[\![j+1|k]\!] + E_4 \beta[\![j|k]\!] + E_5 u[\![j|k]\!] \leq E_0, \tag{2.24}$$

using the so-called big-M method, cf. [151], with matrices E_i of appropriate dimensions, $i = 1, 2, \ldots, 5$, and vectors of auxiliary binary variables β. Regarding these, assume that the modeling process ensures the following:

Assumption 2.5. *There is a one-to-one correspondence between the phase $q(t_k)$ and $\beta(t_k)$.*

Besides constraints, an optimization-based approach also requires the definition of a cost function which assigns a cost value to each run. In the given context, not necessarily all entries of the state vector x are relevant for a cost function. This can, for example, apply to longitudinal positions of vehicles, as long as collision avoidance is ensured by the constraints. In order to select the states relevant to the cost function, $z \in \mathbb{R}^{n_z}$ and a full-rank matrix $C_z \in \mathbb{R}^{n_z \times n_x}$ are introduced, where $n_z < n_x$, such that:

$$z = C_z x.$$

Next, positive definite matrices $C_1 \in \mathbb{R}^{n_z \times n_z}$ and $D_1 \in \mathbb{R}^{n_u \times n_u}$ are introduced, penalizing both the predictions of the weighted magnitude of the control inputs u and the deviation of z from a constant reference value $\bar{z} \in \mathbb{R}^{n_z}$ (for example encoding the center of a lane as desired lateral position):

$$\sum_{j=0}^{H_{\text{plan}}} \left\| C_1 (z[\![j+1|k]\!] - \bar{z}) \right\|^2 + \left\| D_1 u[\![j|k]\!] \right\|^2 \tag{2.25}$$

over a *planning horizon* H_{plan}, based on information given at time instance t_k. Let only the states of the cooperating vehicles be considered in the cost function by appropriate choice of C_z; then, the matrices C_1 and D_1 can – for example – be used to encode the preferred driving style of the vehicles in $\mathcal{C}$. In order to enforce completion of the maneuver within the prescribed duration $T_{\text{s}} \cdot H_{\text{plan}}$, it is necessary that:

$$x[\![H_{\text{plan}}|k]\!] \in \mathcal{X}_{\text{T}}.$$

While (2.21) preserves safety, it is problematic because uncertainty in the predictions of future values of x is accumulated over the planning horizon as j increases: while the predictions $x[\![1|k]\!]$ only depend on $w[\![0|k]\!]$, the prediction of $x[\![H_{\text{plan}}|k]\!]$ depends on all $w[\![j|k]\!]$, $0 \leq j \leq H_{\text{plan}} - 1$. This accumulation of uncertainty requires to make according provisions, which can drastically reduce performance. Even more severe, if the assumption is that the leading vehicle brakes to standstill, then the only plan a following vehicle can make is to also stop, which would prevent execution of any maneuver.

This problem is typically counteracted by the concept of *feedback*, which makes use of the fact that the actual values $x(t_{k+j})$, $1 \leq j \leq H_{\text{plan}}$, become available one after another during execution of a plan which was initially made at time t_k. These values will typically deviate from the predicted values $x[\![j|k]\!]$ due to the uncertainty, in which case at time step t_{k+j}, all subsequent predictions based on $x[\![j|k]\!]$ a posteriori turn out to have been made on an incorrect basis. To correct this, a new plan is generated at time t_{k+j}, typically starting from less pessimistic initial conditions as were predicted. Note that unlike in model-predictive control [123], no receding, but rather a shrinking horizon is employed, as a finite-time problem is considered. This leads to the following problem definition:

Problem 2.1 (Planning Problem)**.** *For a maneuver scheduled to start at initial time t_k with planning horizon H_{plan}, given the momentary state $x(t_{k+i})$, $i = 0, 1, \ldots, H_{plan} - 1$, determine input signals $u[\![j|k+i]\!]$, $j = 0, 1, \ldots, H_{plan} - 1 - i$ as solution of:*

$$\min \sum_{j=0}^{H_{plan}-1-i} \left\| C_1(z[\![j+1|k+i]\!] - \bar{z}) \right\|^2 + \left\| D_1 u[\![j|k+i]\!] \right\|^2$$

subject to

$$E_1 x[\![j|k+i]\!] + E_2 x[\![j+1|k+i]\!] + E_3 \beta[\![j+1|k+i]\!] + E_4 \beta[\![j|k+i]\!] + E_5 u[\![j|k+i]\!] \leq E_0,$$

and $x[\![0|k+i]\!] = x(t_{k+i})$. The reference values result from the output function in (2.8)*:*

$$\bar{x}(t_{k+1+i}) = C x[\![1|k+i]\!] + D u[\![1|k+i]\!].$$

Problem 2.1 is a mixed-integer quadratic program, which can, e.g., be solved using software such as CPLEX [72] or GUROBI [63].

In addition to feedback, another, more application-specific means to limit the effect of uncertainty is to assume that a leading, non-cooperating vehicle, while braking with maximal deceleration, does not brake to stand-still in regular operation, but only to $v_{x,\min} \gg 0$. In a highway setting, this assumption is legitimate as minimum velocities are imposed by traffic rules in order to increase traffic flow. Note that this does not put safety at risk as long as safety distances according to Section 2.3.1 are maintained.

With a formulation of the planning problem at hand, the next chapter focuses on the approximate solution of this problem as well as the approximate computation of controllable sets, which have been introduced in this chapter as a concept that allows to quickly assess the feasibility of a maneuver.

3. Feasibility Assessment and Efficient Planning

In Chapter 2, the maneuver concept has been introduced, allowing for computationally efficient planning. This is beneficial for computing a plan if a *single* maneuver has already been selected. The process in the hierarchical framework in Figure 2.1, however, requires to very quickly assess the feasibility of *several* maneuvers in order to build a set of candidates from which the high-level planner then can choose one maneuver. This requirement is possibly not met by solely ensuring computational efficiency of the trajectory planning procedure, as it still requires to carry out this procedure for every single maneuver in order to identify the feasible ones. Considering that eventually, only one maneuver will be scheduled for execution while all others are discarded, this kind of approach is not even desirable.

The maneuver concept offers a solution to this problem, as it not only increases computational efficiency, but – being based on hybrid automata – also allows to make use of concepts from the well-established field of formal verification of hybrid systems [98]. As outlined in Section 2.1, these methods often rely on the computation of sets with certain properties; in the following, feasibility assessment is carried out relying on *controllable sets* as defined in Definition 2.5. Assume that during the recursive computation of such sets according to Definition 2.5 (i.e., the computation of an N-step controllable set for target phase q_T and target set $\mathcal{X}_\mathrm{T}$), a map $\texttt{post} : \mathcal{K}_j \to \mathcal{K}_{j-1}$ is established for each $j = 1, 2, \ldots, N$, which allows to determine the set to which each $\mathcal{K}_j^{(i)} \in \mathcal{K}_j$ is controllable. That is, given $\mathcal{K}_j^{(i_1)} \in \mathcal{K}_j$ and $\mathcal{K}_{j-1}^{(i_2)} \in \mathcal{K}_{j-1}$:

$$\texttt{post}\left(\mathcal{K}_j^{(i_1)}\right) = \mathcal{K}_{j-1}^{(i_2)} \Leftrightarrow \texttt{pre}_\mathrm{c}\left(\mathcal{K}_{j-1}^{(i_2)}\right) = \mathcal{K}_j^{(i_1)} \vee \texttt{pre}_\mathrm{d}\left(\theta, \mathcal{K}_{j-1}^{(i_2)}\right) = \mathcal{K}_j^{(i_1)}.$$

Based on `post`, define:

Definition 3.1 (Controllable Tube and Entry Set)**.** *Given controllable sets $\mathcal{K}_j$, where $j = 0, 1, 2, \ldots, N$, for a target set $\mathcal{X}_\mathrm{T}$ in target phase q_T, the* controllable tube $\mathcal{T} = \left(\mathcal{T}^{(0)}, \mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(N)}\right)$ *emanating from $\mathcal{K}_N^{(i)} \in \mathcal{K}_N$ is a sequence of controllable sets such that:*

$$\mathcal{T}^{(l)} = \texttt{post}\left(\mathcal{T}^{(l-1)}\right)$$

holds for $l = 1, 2, \ldots, N$. Especially, $\mathcal{T}^{(N)} = \mathcal{K}_0 \equiv \mathcal{X}_\mathrm{T}$ and $\mathcal{T}^{(0)} = \mathcal{K}_j^{(i)}$. In the following, $\mathcal{T}^{(0)}$ is referred to as the entry set *of the tube.*

This concept offers the advantage to explicitly characterize those states for which a solution to the first instance of Problem 2.1, i.e. $i = 0$, exists for a given maneuver. Given the current state of a group of vehicles, this allows to quickly assess the feasibility of a maneuver *prior to planning* by simply checking inclusion of the initial state:

$$x_0 \in \mathtt{proj}_x \mathcal{T}^{(0)} \tag{3.1}$$

for some entry set $\mathcal{T}^{(0)}$ of a controllable tube of the respective maneuver and the current state x_0 of the involved vehicles. For affine flow functions and polyhedral constraints as in Definition 2.2, controllable sets are typically described by linear inequalities, such that checking set inclusion reduces to a matrix-vector product. This is computationally very efficient and allows the high-level scheduling algorithm to quickly evaluate feasibility of different options (i.e., different maneuvers and/or different maneuver durations).

Furthermore, it becomes possible to ensure *recursive feasibility* of the planning procedure, which is understood here as the possibility to find a solution not only to the first planning instance in Problem 2.1, but also for future instances with $i > 0$. Note that this only pertains to a *finite* number of future problem instances, as an optimal control problem with shrinking horizon is solved – contrasting model-predictive control, where recursive feasibility ensures feasibility of an *infinite* number of problem instances [123] with constant prediction horizon length. Unlike in the case of undisturbed system dynamics, (3.1) is only sufficient to ensure the existence of a solution to Problem 2.1 at instance $i = 0$, but not at future instances. This is due to the fact that not all states which are admissible at the next time step $j+1$ can safely be steered to the target set under the influence of the disturbance [24]. Rather, in order to ensure recursive feasibility, it is necessary to enforce that at instance i, the first predicted state $x[\![j+1|k+i]\!]$ again lies in a controllable set despite the worst-case disturbances as represented by the sequence $\tilde{a}(\cdot)$:

Proposition 3.1 (Recursive Feasibility of Problem 2.1)**.** *Consider the setting of Problem 2.1. There, given controllable sets $\mathcal{K}_{H_{plan}-i}$, $i = 0, 1, \ldots, H_{plan} - 1$, the additional constraint*

$$x[\![j+1|k+i]\!] \in \mathtt{proj}_x \mathcal{K}_{H_{plan}-i}$$

ensures that a solution exists for all $i = 0, 1, \ldots, H_{plan} - 1$.

The proof follows immediately from Definition 2.5 and is therefore ommitted. Note that the realization of the planning problem corresponding to this proposition is not advantageous due to reasons of computational efficiency. Rather, (approximations of) controllable tubes should be employed, which are detailed lateron in this chapter. These are computationally more efficient, while Proposition 3.1 still holds.

Even though it is – in principle – possible to compute controllable sets (numerically) exactly, using for example methods as provided by the MPT-toolbox [67], this is impractical for two reasons: Firstly, for a fixed sequence $\beta(\cdot)$ (according to Assumption 2.5, corresponding to a fixed sequence of phases $q(\cdot)$), these computations are based on algorithms from

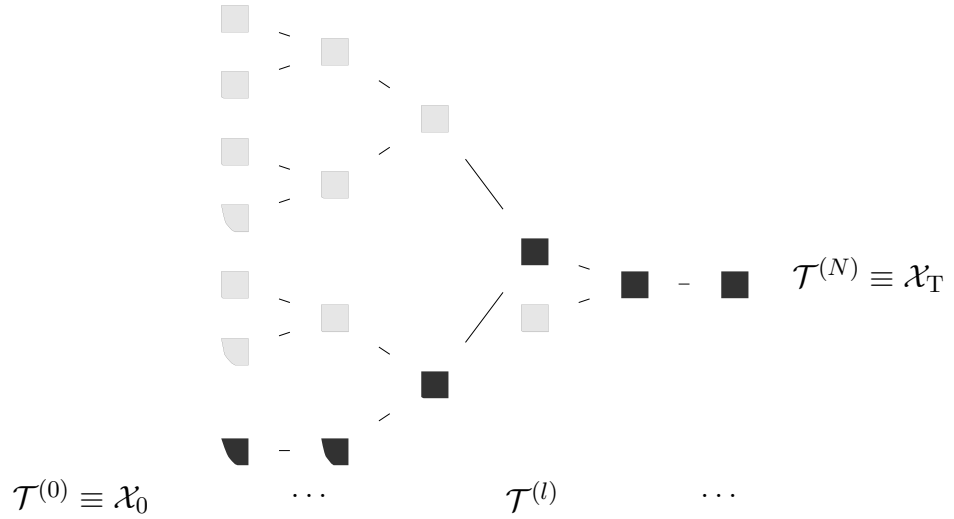


Figure 3.1.: An example for the resulting structure of sets which are controllable to target $\mathcal{X}_\mathrm{T}$ in different numbers of steps. The projection of a controllable tube is shown in black with entry set $\mathcal{X}_0$.

computational geometry (especially Minkowski addition of polytopes) which are known to scale badly with increasing state space dimensionality. Secondly, depending on H_plan and the number of phases and discrete transitions of a specific maneuver formulation, the number of polytopes to be computed may be large. This motivates *approximate* computations as detailed in Section 3.1. These approximations not only extend the applicability of controllable sets to more complex scenarios, but can also be used to drastically speed up the planning process itself, relying on interpolation of optimal trajectories as detailed in Section 3.2. Both sections rely on the fact that a controllable tube $\mathcal{T}$ provides information about the chronological sequence of phases during the completion of a maneuver:

$$q_{H_\mathrm{plan}+1}(\cdot) = \mathtt{proj}_q \mathcal{T}, \tag{3.2}$$

which in turn allows to infer the corresponding sequence $\beta(\cdot)$ in the planning Problem 2.1.

3.1. Inner Approximations of Controllable Sets and Tubes

As detailed in the following, the approximation of controllable sets relies on two concepts: First, in order to keep the number of sets to be computed sufficiently low, these are only computed for a subset of all admissible sequences (3.2). Second, for a given sequence, inner approximations of the controllable sets are computed such that these have a lower complexity of representation than the exact sets.

The approximation of controllable sets pertains to the problem of approximating 1) the solution set of optimization problems as well as 2) the projection of a set onto a lower-dimensional space. For convex problems, these sets are convex and can thus be

approximated by inscribing a simpler convex body in the more complex original set. This approach has been considered in different contexts, with a main difference being the shape of the simpler body used for approximation: in [161], ellipsoids are used, [160] relies on polytopes of pre-defined shape, while [14] relies on boxes. First attempts even exist for the approximation of non-convex sets [155, 156]. In this chapter, polytopic approximations are employed, motivated by the fact that these can approximate a convex body with arbitrary accuracy as opposed to less versatile set representations such as ellipsoids, simplices, boxes, or zonotopes. Approximations of controllable sets are defined recursively:

Proposition 3.2 (Approximations of Controllable Sets)**.** *Given are a positive integer N, the projection $\mathcal{A}_{N-1}$ of an approximation of a controllable set $\mathcal{K} \in \mathcal{K}_{N-1}$ of a maneuver as in Definition 2.5, a polytope $\mathcal{A}_0 \subseteq \mathcal{X}_\mathrm{T}$, m different states $x_0^{(l)} \in \mathbb{R}^{n_x}$, $l \in \{1, 2, \dots, m\}$, an integer j with $0 \leq j \leq N$, a* single *sequence $\beta_{N+1}(\cdot)$ of binary variables (and the sequence $q_{N+1}(\cdot)$ corresponding to it), m input sequences $u_N^{(l)}(\cdot)$, and sequences $\tilde{a}_N^{(l)}(\cdot)$ according to* (2.22)*. Then, define:*

$$\xi^{(l)}(\cdot) := x\left(\cdot; 0, N \cdot T_s, x_0^{(l)}, u_N^{(l)}(\cdot), \tilde{a}_N^{(l)}(\cdot)\right) \tag{3.3}$$

as short-hand for the solution of (2.23) *from time steps 0 to N for the listed parameters as given on the right-hand side, with $x_0^{(l)} = \xi^{(l)}(0)$. Let all $\xi^{(l)}(\cdot)$, $u_N^{(l)}(\cdot)$, $\tilde{a}_N^{(l)}(\cdot)$, as well as $\beta_{N+1}(\cdot)$ be admissible with respect to Problem 2.1 for $i = 0$ and let*

$$\texttt{convh}\left(\left\{\xi^{(1)}(1), \xi^{(2)}(1), \dots, \xi^{(m)}(1)\right\}\right) \subseteq \mathcal{A}_{N-1} \tag{3.4}$$

hold. Then, with

$$\mathcal{A}_N := \texttt{convh}\left(\left\{x_0^{(1)}, x_0^{(2)}, \dots, x_0^{(m)}\right\}\right), \tag{3.5}$$

it holds that $\{q_{N+1}(0)\} \times \mathcal{A}_N$ is an inner approximation of a N-step controllable set*. An* approximated controllable tube *is defined as*

$$\tilde{\mathcal{T}} := \left(\{q_{N+1}(j)\} \times \texttt{convh}\left(\left\{\xi^{(l)}(j)\right\}_{l=1}^{m}\right)\right)_{j=0}^{N} \tag{3.6}$$

Proof. Because the sequences $u_N^{(l)}(\cdot)$ and $\tilde{a}_N^{(l)}(\cdot)$ lead to admissible trajectories $\xi^{(l)}(\cdot)$ which by virtue of (3.4) pass through the projection $\mathcal{A}_{N-1}$ of the inner approximation of an $N-1$-step controllable set, for each $x_0^{(l)}$ there must exist some $i_l \in \{1, 2, \dots, |\mathcal{K}_N|\}$, such that $x_0^{(l)} \in \texttt{proj}_x \mathcal{K}_N^{(i_l)}$ for $\mathcal{K}_N^{(i_l)} \in \mathcal{K}_N$. All m trajectories are admissible for the same $\beta_{N+1}(\cdot)$, which reduces Problem 2.1 to a convex quadratic program [25]. The solution set of this problem class is convex, which implies that there exists a single i^* such that $x_0^{(l)} \in \texttt{proj}_x \mathcal{K}_N^{(i^*)}$ $\forall\, l \in \{1, 2, \dots, m\}$ and consequently also $\mathcal{A}_N \in \texttt{proj}_x \mathcal{K}_N^{(i^*)}$. Lifting to the q-x-space then gives the result. □

While it may be possible to derive sequences $\beta_{N+1}(\cdot)$ in an automated way, for example based on measurements taken from corresponding human driving maneuvers, in the following, such sequences are assumed to be given by the designer's decision. For a given sequence, the construction of an approximation requires to determine m points $x_0^{(l)}$, $l = 1, 2, \ldots, m$, with corresponding admissible input trajectories $u_N^{(l)}(\cdot)$. In [97], an approach to this problem is given which relies on iterative extension of a given initial set. This approach is used in the following, where Section 3.1.1 detailes the iterative procedure, while its initialization is described in Section 3.1.2.

3.1.1. Iterative Approximation

A detailed study of the problem of approximating convex bodies is provided by the book [97], which, however, seems to be overlooked by the control community, considering more recent, yet conceptually similar approaches such as the algorithm in [40] or the concept of *template polyhedra* [89]. In the following, the so-called *Estimate Refinement method* ([97, p. 258]) will be employed, whose basic idea is to extend an initial inner polytopic approximation $\mathcal{P}^{(0)}$ of a polytope $\mathcal{P} \subset \mathbb{R}^{n_x}$ by incrementally adding new vertices. In iteration n, a new polytope $\mathcal{P}^{(n+1)}$ results as convex hull of the existing set $\mathcal{P}^{(n)}$ and a new vertex $v^{(n)}$, the choice of which is an elementary aspect of each iteration. For completeness, the main elements of the method are restated here in a condensed form. It relies on the following concepts:

Definition 3.2 (Support function and points of tangency [97])**.** *The support function $g_{\mathcal{P}} : \mathbb{R}^{n_x} \to \mathbb{R}$ of a polytope $\mathcal{P} \subset \mathbb{R}^{n_x}$ is defined as:*

$$g_{\mathcal{P}}(p) = \max \left\{ p^{\mathsf{T}} x | x \in \mathcal{P} \right\}. \tag{3.7}$$

Denoting the boundary of $\mathcal{P}$ by $\partial\mathcal{P}$, corresponding maximizers are contained in the set of points of tangency

$$T_{\mathcal{P}}(p) = \left\{ x \in \partial\mathcal{P} | p^{\mathsf{T}} x = g_{\mathcal{P}}(p) \right\}.$$

Definition 3.3 (Hausdorff distance [97])**.** *The* Hausdorff distance *between two compact, non-empty sets $C_1 \subset \mathbb{R}^{n_x}$, $C_2 \subset \mathbb{R}^{n_x}$ is defined as:*

$$\delta^H(C_1, C_2) = \max \left\{ \sup \left\{ \rho(x, C_2) : x \in C_1 \right\}, \sup \left\{ \rho(x, C1) : x \in C_2 \right\} \right\}$$

with distance *ρ between a point x and a set C:*

$$\rho(x, C) := \inf \left\{ ||x - x'|| : x' \in C \right\}.$$

Termed an *adaptive* scheme in [97], the choice of a new point relies on information about the approximated set $\mathcal{P}$. This is possible even if no explicit characterization of $\mathcal{P}$ is known, as long as its support function can be evaluated. Given an initial approximation $\mathcal{P}^{(0)}$ and a maximum number $N_{\max}$ of vertices to be added to it, the major steps of the procedure

are summarized in Algorithm 3.1: in the nth iteration, at first, the facets, i.e., the faces of dimension $n_x - 1$, of the latest approximation $\mathcal{P}^{(n)}$ are computed. To that end, denote by $\mathtt{f}(\mathcal{P})$ the set of outward-pointing unit normals of the facets of polytope $\mathcal{P}$. Among all unit normal vectors of these facets, the one is selected which maximizes a lower bound on the Hausdorff distance between the approximation and the original (yet unknown) set:

$$p^{(n)} = \arg \max_{p \in \mathtt{f}(\mathcal{P}^{(n)})} \left\{ g_{\mathcal{P}}(p) - g_{\mathcal{P}^{(n)}}(p) \right\}. \tag{3.8}$$

Then, the new vertex $v^{(n)}$ is chosen as corresponding support vector:

$$v^{(n)} \in T_{\mathcal{P}} \left(p^{(n)} \right).$$

As the evaluation of the support function requires the solution of a linear program, the selection of a maximizer can be entrusted to the numerical optimization algorithm even if $T_{\mathcal{P}}$ is no singleton, because such algorithms are designed to return only one solution. Having determined $v^{(n)}$, it is then added to the set of vertices of $\mathcal{P}^{(n)}$, yielding the vertices of $\mathcal{P}^{(n+1)}$:

$$\mathtt{v}\left(\mathcal{P}^{(n)}\right) \cup \left\{ v^{(n)} \right\}.$$

From these, the facets can be constructed, which serve as the basis for the next iteration. The procedure enjoys several properties, e.g. upper and lower bounds on $\delta^H \left(\mathcal{P}^{(n)}, \mathcal{P} \right)$ can be derived (see [97, Chapter 8]). Most importantly, the following holds:

Proposition 3.3. *The sequence of generated polytopes $\mathcal{P}^{(n)}$ converges to the original set $\mathcal{P}$ in the Hausdorff distance([97, Theorem 8.5]):*

$$\lim_{n \to \infty} \delta^H \left(\mathcal{P}^{(n)}, \mathcal{P} \right) = 0,$$

from which convergence in the volumetric sense follows.

While a valuable insight, practical application of the procedure focuses on the computational complexity of the representation of $\mathcal{P}^{(n)}$, which is equivalent to the number of added vertices and iterations n. These are limited by $N_{\max}$, which must be chosen as a compromise between approximation quality and complexity of the set representation.

The computational complexity of Algorithm 3.1 is dominated by two tasks: First, the maximum in (3.8) is determined by enumeration of the values resulting for each facet normal $p \in \mathtt{f}\left(\mathcal{P}^{(n)}\right)$, every time requiring to determine $g_{\mathcal{P}}(p)$ and $g_{\mathcal{P}^{(n)}}(p)$. For $\mathcal{P}^{(n)} = \{x | A_n x \leq b_n\}$, computations are not necessary as $g_{\mathcal{P}^{(n)}}(p) = b_n$; for $g_{\mathcal{P}}(p)$, however, every evaluation of the support function corresponds to solving an optimization problem similar to Problem 2.1, but with given sequence $\beta(\cdot)$ (respectively $q(\cdot)$) and the support function (3.7) as a cost function (3.7), resulting in one linear program per facet.

Second, maintaining both vertex and facet representations and especially computing the facets from a vertex representation of a polytope in each iteration is computationally expensive. In this thesis, this motivates the use of a so-called *incremental convex hull* algorithm [113] in line 3, which allows to incrementally build on the results from previous iterations, i.e., already existing convex hulls of a set of points.

Input $\mathcal{P}^{(0)}$, $N_{\max}$
1: **for** $n = 0$ to $N_{\max} - 1$ **do**
2: $\Delta g \leftarrow 0$, $p \leftarrow 0$
3: $f_{\mathcal{P}} \leftarrow \mathtt{f}\left(\mathcal{P}^{(n)}\right)$ ▷ Compute facets
4: **for** $p \in f_{\mathcal{P}}$ **do**
5: **if** $\Delta g \leq g_{\mathcal{P}}(p) - g_{\mathcal{P}^{(n)}}(p)$ **then** ▷ Evaluate support functions
6: $p^{(n)} \leftarrow p$
7: $\Delta g \leftarrow g_{\mathcal{P}}(p) - g_{\mathcal{P}^{(n)}}(p)$
8: **end if**
9: **end for**
10: $v^{(n)} \leftarrow x \in T_{\mathcal{P}}\left(p^{(n)}\right)$ ▷ Obtain new vertex
11: $\mathcal{P}^{(n+1)} = \mathtt{convh}\left(\left\{v^{(n)}, \mathtt{v}\left(\mathcal{P}^{(n)}\right)\right\}\right)$
12: $n \leftarrow n + 1$
13: **end for**
Output $\mathcal{P}^{(N_{\max})}$

Algorithm 3.1.: Estimate refinement algorithm from [97, p. 258].

3.1.2. Initialization of the Approximation Scheme

Algorithm 3.1 requires an initial polytopic inner approximation $\mathcal{P}^{(0)} \subseteq \mathcal{P}$ of the unknown polytope $\mathcal{P}$. If a good approximation is the objective, it is generally desirable to make $\mathcal{P}^{(0)}$ as similar to $\mathcal{P}$ as possible; in the following, however, the objective is slightly different, aiming to maximize the approximation's volume while keeping the number of its vertices low.

Because the problem of maximizing the volume of a general polytope inscribed in another polytope is non-convex, it cannot be solved to global optimality in general. Instead, a possible approach [13] is to compute an inner approximation by choosing a set of *template vectors* [89] indicating directions along which one attempts to find the furthest point that still yields a solution to Problem 2.1. In this chapter, the following approach is used, based on a modified version of Problem 2.1:

Problem 3.1 (Computation of an initial inner approximation)**.** *Given a full-dimensional polytopic set $\mathcal{X}_{0,des} \subset \mathbb{R}^{n_x}$ with m vertices $\hat{x}_0^{(l)}$, $l \in \{1, 2, \ldots, m\}$, a weighting matrix $W \in \mathbb{R}^{n_x \times n_x}$, $W \geq 0$, a planning horizon $H_{plan} \in \mathbb{N}^+$, a single sequence $\hat{\beta}_{H_{plan}+1}(\cdot)$, and sequences $\tilde{a}_N^{(l)}(\cdot)$, determine sequences $x_{H_{plan}+1}^{(l)}(\cdot)$ and $u_{H_{plan}}^{(l)}(\cdot)$ as solution to the problem:*

$$\min \left(x^{(l)}(t_0) - \hat{x}_0^{(l)}\right)^{\mathsf{T}} W \left(x^{(l)}(t_0) - \hat{x}_0^{(l)}\right)$$

subject to (recall (2.24)):

$$E_1 x^{(l)}(t_k) + E_2 x^{(l)}(t_{k+1}) + E_3 \hat{\beta}(t_{k+1}) + E_4 \hat{\beta}(t_k) + E_5 u^{(l)}(t_k) \leq E_0$$

for all $k = 0, 1, \ldots, H_{plan} - 1$. *Then, obtain an initial approximation as follows:*

$$\mathcal{P}^{(0)} = \texttt{convh}\left(\left\{x^{(l)}(t_0)\right\}_{l=1}^{m}\right).$$

The sequence $\hat{\beta}_{H_{\text{plan}}+1}(\cdot)$ can be obtained by solving a single instance of Problem 3.1, with $\hat{\beta}_{H_{\text{plan}}}(\cdot)$ treated as variables; this results in a MIQP instead of a QP. The solution of Problem 3.1 allows the designer to specify desirable vertices by choice of $\mathcal{X}_{0,\text{des}}$. Alternatively, instead of solving m problems sequentially, a single (however large and computationally more expensive) MIQP could be solved.

3.2. Fast Planning using Approximated Controllable Sets

The basic idea to ensure fast planning is to add constraints to Problem 2.1 which confine the run of the corresponding hybrid automaton to a controllable tube. This determines the temporal sequence of phases according to (3.2), from which a speed-up of the planning procedure can be expected. Generally, two realizations are conceivable: On the one hand, the mixed-integer program in Problem 2.1 can be reduced to a quadratic program by fixing the sequences $\beta(\cdot)$ resp. $q(\cdot)$. While this would lead to a less complex problem class, it would still require optimization online, i.e. during the operation of a vehicle. On the other hand – and this approach is pursued in the following – it is possible to avoid online optimization, relying on computations which have been carried out offline and from which a (not necessarily optimal) plan can be obtained online, depending on the initial state as a parameter.

An approach which (independently of the application considered here) even obtains optimal solutions in this way is based on multi-parametric programming [24], which, however, is known to be tractable only for relatively low-dimensional systems due to its computational complexity. In cooperative driving, in contrast, *groups* of vehicles are considered, which typically leads to high-dimensional state spaces, such that exact multi-parametric programming is not applicable. The exact computation of controllable sets is subject to similar limitations. Therefore, the following procedure relies on approximations of controllable sets as described in the previous section. A conceptually similar approach is described in [13], yet not in the context of controllable sets for hybrid systems. It may also be interpreted as a form of approximate dynamic programming as in [17]. The following approach relies on *barycentric coordinates*:

Definition 3.4 (Barycentric Coordinates [76])**.** *Given a polytope* $\mathcal{P} \subset \mathbb{R}^{n_x}$ *with* m *vertices contained in* $\mathtt{v}(\mathcal{P}) = \{v_1, v_2, \ldots, v_m\}$ *and an interior point* $x \in \mathtt{int}(\mathcal{P})$, barycentric coordinates $\lambda \in \mathbb{R}^m$ *fulfill:*

$$x = \sum_{v_i \in \mathtt{v}(\mathcal{P})} \lambda_{[i]} v_i, \qquad \lambda_{[i]} \geq 0 \ \forall i = 1, 2, \ldots, m, \qquad 1 = \sum_{i=1}^{m} \lambda_{[i]}.$$

Note that depending on the polytope $\mathcal{P}$, barycentric coordinates may not be unique. The computation of trajectories relies on interpolation:

Proposition 3.4 (Trajectory Interpolation)**.** *Given an approximated controllable tube $\tilde{\mathcal{T}}$ as in Proposition 3.2 along with its determining quantities, let:*

$$\hat{X}(t_k) := \begin{bmatrix} \xi^{(1)}(t_k) & \xi^{(2)}(t_k) & \dots & \xi^{(m)}(t_k) \end{bmatrix},$$
$$\hat{U}(t_k) := \begin{bmatrix} u^{(1)}(t_k) & u^{(2)}(t_k) & \dots & u^{(m)}(t_k) \end{bmatrix}.$$

Then, for any $x \in \texttt{proj}_x \tilde{\mathcal{T}}^{(0)}$ there exist barycentric coordinates $\lambda \in \mathbb{R}^m$, $\lambda_{[i]} \geq 0 \ \forall i = 1, 2, \dots, m$, $\sum_{i=1}^{m} \lambda_{[i]} = 1$, such that the input sequence

$$u_N(\cdot) = \left(\hat{U}(t_k)\lambda\right)_{k=0}^{N-1}$$

is admissible and leads to an admissible trajectory $\xi(\cdot)$ as in (3.3), with

$$u_N(t_k) \in \mathcal{U},$$
$$\xi(t_k) \in \texttt{proj}_x \tilde{\mathcal{T}}^{(k)}$$

for all $k \in \{0, 1, \dots, N\}$.

Proof. Suppose that $x(t_k) \in \texttt{proj}_x \tilde{\mathcal{T}}^{(k)}$. Existence of λ such that $x(t_k) = \hat{X}(t_k)\lambda$ follows directly from convexity. Defining

$$F_k := \begin{bmatrix} \tilde{a}^{(1)}(k) & \tilde{a}^{(2)}(k) & \dots \tilde{a}^{(m)}(k) \end{bmatrix} \in \mathbb{R}^{n_x \times m},$$

it holds for the affine dynamics underlying Proposition 3.2 that

$$\hat{X}(t_{k+1}) = A\hat{X}(t_k) + B\hat{U}(t_k) + F_k,$$

which, by multiplying by λ and defining:

$$x(t_{k+1}) := \hat{X}(t_{k+1})\lambda,$$
$$u(t_k) := \hat{U}(t_k)\lambda,$$
$$\tilde{a}(t_k) := F_k\lambda,$$

reduces to affine dynamics (2.7). Due to convexity of $\mathcal{U}$ and $\texttt{proj}_x \tilde{\mathcal{T}}^{(k+1)}$, both $x(t_{k+1}) \in \texttt{proj}_x \tilde{\mathcal{T}}^{(k+1)}$ and $u(t_k) \in \mathcal{U}$ hold. Since $x = x(t_0) \in \texttt{proj}_x \tilde{\mathcal{T}}^{(0)}$, induction over k leads to the result. □

Proposition 3.4 suggests the procedure summarized in Algorithm 3.2 for online trajectory planning at a certain time instance, given an approximated controllable tube $\tilde{\mathcal{T}}$ which has been selected before by the high-level controller: after having extracted the time-varying matrices of state and input trajectories $\hat{X}(\cdot)$ and $\hat{U}(\cdot)$ associated with the vertices of the entry set $\tilde{\mathcal{T}}^{(0)}$, barycentric coordinates are computed for the given initial state x_0 as outlined in the following. These coordinates allow to interpolate between the vertex trajectories in order to obtain state and input trajectories for the maneuver starting in x_0, from which a reference trajectory can then be obtained based on the function h as in Definition 2.2.

Input x_0, $\tilde{\mathcal{T}}$, H_{plan}
1: $[\hat{X}(\cdot), \hat{U}(\cdot)] \leftarrow \text{GetVertexTrajectories}\big(\tilde{\mathcal{T}}\big)$
2: $\lambda \leftarrow \text{ComputeLambda}\big(x_0, \tilde{\mathcal{T}}\big)$ ▷ cf. Section 3.2.2
3: $x(t_0) \leftarrow x_0$
4: **for** $k = 0$ to $H_{\text{plan}} - 1$ **do**
5: $\quad x(t_{k+1}) \leftarrow \hat{X}(t_{k+1})\lambda$
6: $\quad u(t_k) \leftarrow \hat{U}(t_k)\lambda$
7: **end for**
Output $x_{H_{\text{plan}}+1}(\cdot)$, $u_{H_{\text{plan}}}(\cdot)$

Algorithm 3.2.: Schematic: Online trajectory planning.

Employing Algorithm 3.2 according to the receding horizon as in the planning problem 2.1 requires to run this at every time instance until completion of a maneuver, rendering all but the first step of the iteration starting in line 4 obsolete.

The computation of barycentric coordinates for a given state x_0 is a central element of Algorithm 3.2. Even though barycentric coordinates have found wide application in linear parameter-varying (LPV) control (cf. [150] for an overview), usually only simplified settings are considered where coordinates are computed for hyperrectangles and not for general polytopes as in this work. While it is always possible to compute λ by solving a linear program, the approach pursued here explicitly aims at avoiding online optimization in order to ensure fast planning. In the following, two possibilities are detailed which provide expressions which do not require iterative numerical computations and can therefore be evaluated efficiently online.

3.2.1. Triangulation

The first option, for example used in [13, 75, 128], relies on *triangulation* of general polytopes into *simplices* according to the following definitions:

Definition 3.5 (Simplex [162])**.** *A* simplex *in* $\mathbb{R}^{n_x}$ *is a polytope of dimension* n_x *with* $n_x + 1$ *vertices.*

Definition 3.6 (Triangulation [24])**.** *Given a finite set of points* $\mathcal{V} = \{v_1, v_2, \ldots, v_{n_v}\} \subset \mathbb{R}^{n_x}$ *and a matrix* $V = \begin{bmatrix} v_1 & v_2 & \ldots & v_{n_v} \end{bmatrix}^\mathsf{T}$*,* a triangulation *of* $\mathcal{V}$ *is a finite set* $\texttt{tri}\,(\mathcal{V}) = \{S_1, S_2, \ldots, S_{n_s}\}$ *of index sets, such that*

1. $\mathcal{S}_i := \texttt{convh}\left(\left\{\left(V_{[j,:]}\right)^\mathsf{T} | j \in S_i\right\}\right)$ *defines a full-dimensional simplex for all* $i = 1, 2, \ldots, n_s$*,*
2. $\texttt{convh}\,(\mathcal{V}) \equiv \cup_{i=1}^{n_s} \mathcal{S}_i$ *and* $\texttt{int}\,(\mathcal{S}_i) \cap \texttt{int}\,(\mathcal{S}_j) = \emptyset$ *for* $i \neq j$*.*

An example for a triangulation of a polyhedron in a two-dimensional space is illustrated in Fig. 3.2. In general, different triangulations of a polytope exist, such as different methods

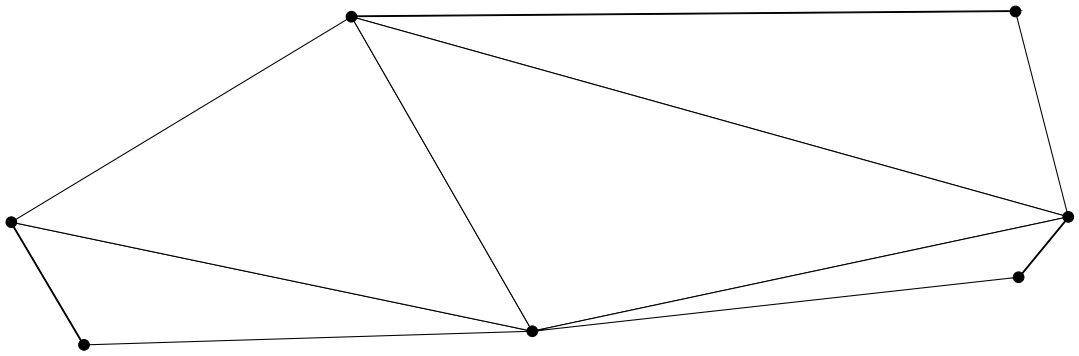


Figure 3.2.: Triangulation of an example polytope.

for its computation [139, Chpt. 27, 29]. In the following, a *Delaunay triangulation* is employed [139, Chpt. 27], which can be computed relying on the qhull-library [11].

For a simplex $\mathcal{S}_i$ as introduced above and an interior point $x_0 \in \texttt{int}(\mathcal{S}_i)$, barycentric coordinates can be computed by:

$$\lambda = \begin{bmatrix} 1_{1\times n_x+1} \\ \left(V_{[S_i,:]}\right)^\mathsf{T} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ x_0 \end{bmatrix}. \tag{3.9}$$

Note that for online application, the matrix inverse in (3.9) can be pre-computed offline for each simplex, such that online computations are limited to identifying the simplex containing the current state x_0 and executing the above matrix-vector-product. The number of simplices resulting from the triangulation, however, is reported to be of the order $n_v^{\lfloor \frac{n_x+1}{2} \rfloor}$ in the worst case, depending on the dimension n_x of the state space, the number of vertices, and the shape of the triangulated set [58].

3.2.2. Warren's Coordinates

The second option for the computation of coordinates λ for a given initial state does not rely on triangulation of the considered polytope, but directly assigns coordinates to it. Generally, barycentric coordinates are non-unique, such that different approaches to this task exist, cf. e.g. [144] for polygons. In this section, an approach based on [146, 147], and mainly [76] is employed. The basic theory is outlined in [146] and extended to arbitrary convex sets in [147], however being vague about details required for handling so-called non-simple polytopes, which often arise in practically relevant problems. Nonetheless, it has found application to approximate explicit MPC [75]. These details regarding non-simple polytopes are then given in [76], which, according to [52], provides a "cleaner solution" to this very topic than [146, 147]. While full generality is claimed, the focus of the derivation in [76] is on two- and three-dimensional sets. For completeness, this section states the procedure independently of the dimensionality of a polytope and contributes in addition a consideration of questions of online computation.

General Procedure

A fundamental concept in the procedure devised by [76] is the *polar dual* of a polyhedron $\mathcal{P}$:

Definition 3.7 (Polar Dual [76])**.** *For a convex polyhedron $\mathcal{P} = \{x|Ax \leq b\} \subset \mathbb{R}^{n_x}$, $A \in \mathbb{R}^{n_A \times n_x}$, $b \in \mathbb{R}^{n_A}$, that contains the origin and whose n_v vertices are arranged row-wise in a matrix $V \in \mathbb{R}^{n_v \times n_x}$ such that $\mathcal{P} = \texttt{convh}\left(\left\{\left(V_{[i,:]}\right)^\mathsf{T} | i = 1, 2, \ldots, n_v\right\}\right)$, the* polar dual *is a convex polyhedron of the form*

$$\texttt{d}(\mathcal{P}) = \left\{y | y^\mathsf{T} \cdot x \leq 1 \ \forall x \in \mathcal{P}\right\}$$

which contains the origin in its interior. Also, there is a one-to-one correspondence between the matrix V of vertices of the primal and the facets of the dual according to:

$$\begin{aligned}\texttt{d}(\mathcal{P}) &= \{y | Vy \leq 1_{n_v \times 1}\}, \\ \texttt{d}(\mathcal{P}) &= \texttt{convh}\left(\left\{\frac{A_{[j,:]}}{b_{[j]}} | \ j \in \{1, 2, \ldots, n_A\}\right\}\right).\end{aligned}$$

The procedure devised by [76] is motivated by the fact that for a general polytope, barycentric coordinates can be expressed as ratios of volumes pertaining to the polar dual of the polytope, as outlined in the following. For a given polytope $\mathcal{P} \subset \mathbb{R}^{n_x}$ and a point $x_1 \in \texttt{int}(\mathcal{P})$, the objective is to compute barycentric coordinates according to Definition 3.4. Without loss of generality, it is assumed that $\mathcal{P}$ is full-dimensional; otherwise, projection on its affine hull can ensure this condition (which is also applied for points on the boundary of $\mathcal{P}$). Given both a vertex- and a halfspace-representation of $\mathcal{P}$ as:

$$\mathcal{P} = \texttt{convh}\left(\{v_1, v_2, \ldots, v_{n_v}\}\right) = \{x | Ax \leq b\},$$

the first step in the procedure is to shift it by x_1, defining $\bar{v}_i := v_i - x_1$ for $i = 1, 2, \ldots, n_v$, and $\bar{x} := x - x_1$, such that:

$$\bar{\mathcal{P}} := \texttt{convh}\left(\{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_{n_v}\}\right) = \{\bar{x} | A(\bar{x} + x_1) \leq b\}. \tag{3.10}$$

This ensures that $\bar{\mathcal{P}}$ contains the origin. Then, define:

$$\begin{aligned}\bar{V} &:= \begin{bmatrix}\bar{v}_1 & \bar{v}_2 & \ldots & \bar{v}_{n_v}\end{bmatrix}^\mathsf{T} \in \mathbb{R}^{n_v \times n_x}, \\ \bar{b} &:= b - Ax_1.\end{aligned}$$

For the following procedure, the polar dual of $\bar{\mathcal{P}}$ must be obtained in both vertex- and halfspace-representation:

$$\bar{\mathcal{D}} := \texttt{d}\left(\bar{\mathcal{P}}\right) = \left\{y | \hat{A}y \leq \hat{b}\right\} = \texttt{convh}\left(\left\{\bar{\hat{v}}_1, \bar{\hat{v}}_2, \ldots, \bar{\hat{v}}_{n_A}\right\}\right), \tag{3.11}$$

which is not difficult, as according to Definition 3.7,

$$\hat{A} = \bar{V}, \quad \hat{b} = 1_{n_v \times 1}, \quad \bar{\hat{v}}_j^\mathsf{T} = \frac{A_{[j,:]}}{\bar{b}_{[j]}}, \quad j \in \{1, 2, \ldots, n_A\}.$$

Next, define

$$\hat{V} := \begin{bmatrix} \bar{\hat{v}}_1 & \bar{\hat{v}}_2 & \dots & \bar{\hat{v}}_{n_A} \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^{n_A \times n_x}.$$

Given this data, for the ith facet of $\bar{\mathcal{D}}$ as defined by the corresponding normal vector $\left(\hat{A}_{[i,:]}\right)^{\mathsf{T}}$ and $\hat{b}_{[i]} = 1$, $i = 1, 2, \dots, n_v$, the incident vertices are determined. The corresponding index set is defined as:

$$\begin{aligned} \mathcal{F}_i :&= \left\{ j \in \mathbb{N}^+ | \hat{V}_{[j,:]} \left(\hat{A}_{[i,:]}\right)^{\mathsf{T}} = 1 \right\} \\ &= \left\{ j \in \mathbb{N}^+ | \hat{V}_{[j,:]} \bar{v}_i = 1 \right\}. \end{aligned}$$

Based on these index sets, a triangulation

$$\bar{T}_i = \texttt{tri}\left(\left\{ \left(\hat{V}_{[j,:]}\right)^{\mathsf{T}} | j \in \mathcal{F}_i \right\} \right) = \{S_1, S_2, \dots, S_{s_i}\} \tag{3.12}$$

of each facet can be computed, consisting of s_i simplices in $\mathbb{R}^{n_x - 1}$. The convex hull of the simplex in $\mathbb{R}^{n_x - 1}$ corresponding to $S_j \in \bar{T}_i$:

$$\mathcal{S}_j := \texttt{convh}\left(\left\{ \left(\hat{V}_{[i,:]}\right)^{\mathsf{T}} | i \in S_j \right\} \right),$$

and the origin:

$$\texttt{convh}\left(\mathcal{S}_j \cup \{0\}\right) \tag{3.13}$$

defines a simplex in $\mathbb{R}^{n_x}$. Its volume is given by:

$$\texttt{vol}\left(\texttt{convh}\left(\mathcal{S}_j \cup \{0\}\right)\right) = \left| \frac{1}{n_x!} \det \hat{V}_{[S_j,:]} \right|. \tag{3.14}$$

Summation of the corresponding volumes for all simplices defined by a partition gives the so-called weight ω_i of vertex $\bar{v}_i$:

$$\omega_i := \sum_{S \in \bar{T}_i} \left| \frac{1}{n_x!} \det \hat{V}_{[S,:]} \right|. \tag{3.15}$$

From this, the coordinates $\lambda_{[i]}$ can be computed as:

$$\lambda_{[i]} = \frac{\omega_i}{\sum_{j=1}^{n_v} \omega_j},$$

where the denominator $\sum_{j=1}^{n_V} \omega_j$ is the volume of the dual polytope $\bar{\mathcal{D}}$. The procedure (starting after the pre-processing of the input polytope $\mathcal{P}$ as in (3.10)) is summarized in Algorithm 3.3a.

Online Computation

The computations outlined in the previous section operate on a polytope $\bar{\mathcal{P}}$ which results from shifting a set $\mathcal{P}$ by a given point x_1 for which barycentric coordinates λ are sought. In the considered context, $\mathcal{P}$ corresponds to the entry set of a controllable tube and x_1 to the momentary state of the vehicles involved in the corresponding maneuver. This state is not known offline, but becomes available only at the time a maneuver is to be planned. Therefore, the operations from the previous sections would have to be performed online. Especially the computation of a triangulation can be computationally demanding, such that no conceptual advantage against a procedure relying on online optimization would be gained. The following section demonstrates how to resolve this problem by modifying the algorithm such that it can be applied online at low computational cost.

Having determined barycentric coordinates λ and all related quantities for some point $x_1 \in \mathcal{P}$ offline as outlined in the previous section and given some initial state $x_2 \in \mathcal{P}$ for which barycentric coordinates $\tilde{\lambda}$ are sought, the basic procedure is to shift $\bar{\mathcal{P}}$ by $x_2 - x_1$ and to then establish a relation to the results obtained offline for $\bar{\mathcal{P}}$ which can be evaluated more easily online. This approach requires to investigate the impact of a shift on 1) the dual polytope (3.11), 2) the triangulations (3.12) of the dual's facets, and 3) the weights (3.15).

Proposition 3.5. *Given a vertex-representation $\hat{V} \in \mathbb{R}^{n_A \times n_x}$ of the dual of a polytope $\mathcal{P}$ shifted by $x_1 \in \mathtt{int}(\mathcal{P})$, triangulations $\bar{T}_i$, $i = 1, 2, \ldots, n_v$, of the dual polytope's facets, and a point $x_2 \in \mathtt{int}(\mathcal{P})$, defining:*

$$\sigma_j := \frac{1}{1 - \bar{\hat{v}}_j^{\mathsf{T}} \cdot (x_2 - x_1)}$$

for all $j = 1, 2, \ldots, n_A$ allows to compute modified weights

$$\tilde{\omega}_i := \sum_{S \in \bar{T}_i} \left| \frac{1}{n_x!} \det \hat{V}_{[S,:]} \right| \cdot \prod_{k \in S} \sigma_k.$$

Then, barycentric coordinates $\tilde{\lambda}$ for $x_2 \in \mathcal{P}$ are given by:

$$\tilde{\lambda}_{[i]} = \frac{\tilde{\omega}_i}{\sum_{j=1}^{n_v} \tilde{\omega}_j}$$

for $i = 1, 2, \ldots, n_v$ such that $x_2 = V^{\mathsf{T}} \tilde{\lambda}$, $\tilde{\lambda}_{[i]} \geq 0$, and $\sum_{i=1} \tilde{\lambda}_{[i]} = 1$.

Proof. Application of Algorithm 3.3a to $\mathcal{P}$ and $x_2 \in \mathcal{P}$ would result in a shifted primal

$$\tilde{\mathcal{P}} := \mathtt{convh}\left(\{v_1 - x_2, v_2 - x_2, \ldots, v_{n_v} - x_2\}\right) = \{\tilde{x} | A(\tilde{x} + x_2) \leq b\},$$

where $\tilde{x} := x - x_2$, with the corresponding dual

$$\tilde{\mathcal{D}} := \mathtt{d}\left(\tilde{\mathcal{P}}\right) = \mathtt{convh}\left(\left\{\tilde{\hat{v}}_1, \tilde{\hat{v}}_2, \ldots, \tilde{\hat{v}}_{n_A}\right\}\right)$$

and triangulations $\tilde{T}_i$ of all of the dual's facets, $i = 1, 2, \ldots, n_v$. According to Definition 3.7, with $\tilde{b} := b - Ax_2$, the dual's vertices result from:

$$\tilde{\hat{v}}_j = \frac{\left(A_{[j,:]}\right)^\mathsf{T}}{\tilde{b}_{[j]}}, \quad j = 1, 2, \ldots, n_A.$$

Equivalently, $\tilde{\mathcal{P}}$, $\tilde{\mathcal{D}}$, and $\tilde{T}_i$ can be expressed in terms of $\bar{\mathcal{P}}$, $\bar{\mathcal{D}}$, and $\bar{T}_i$:

$$\begin{aligned}\tilde{\mathcal{P}} &= \texttt{convh}\left(\{v_1 - x_1 + x_1 - x_2, v_2 - x_1 + x_1 - x_2, \ldots, v_{n_V} - x_1 + x_1 - x_2\}\right) \\ &= \texttt{convh}\left(\{\bar{v}_1 + x_1 - x_2, \bar{v}_2 + x_1 - x_2, \ldots, \bar{v}_{n_V} + x_1 - x_2\}\right)\end{aligned}$$

and, as

$$\tilde{b} = b - Ax_2 = b - A(x_2 - x_1 + x_1) = \bar{b} - A(x_2 - x_1),$$

it holds according to Definition 3.7 that

$$\tilde{\hat{v}}_j = \frac{A_{[j,:]}^\mathsf{T}}{\bar{b}_{[j]}} \frac{1}{1 - \frac{A_{[j,:]}}{\bar{b}_{[j]}} \cdot (x_2 - x_1)} = \frac{1}{1 - \bar{\hat{v}}_j^\mathsf{T} \cdot (x_2 - x_1)} \bar{\hat{v}}_j = \sigma_j \bar{\hat{v}}_j.$$

This allows to relate the matrix of vertices of $\tilde{\mathcal{D}}$ to that of $\bar{\mathcal{D}}$:

$$\tilde{\hat{V}} := \begin{bmatrix}\tilde{\hat{v}}_1 & \tilde{\hat{v}}_2 & \ldots & \tilde{\hat{v}}_{n_A}\end{bmatrix}^\mathsf{T} = \texttt{diag}\left(\sigma_1, \sigma_2, \ldots, \sigma_{n_A}\right) \hat{V}, \tag{3.16}$$

which reveals that $\tilde{\mathcal{D}}$ – for given x_2 – results as a linear transformation of the polar dual $\bar{\mathcal{D}}$. Because

$$x_1, x_2 \in \texttt{int}(\mathcal{P}) \Rightarrow x_2 - x_1 \in \texttt{int}(\bar{\mathcal{P}})$$

and because $\bar{P}$ contains the origin, it holds that

$$\frac{A_{[j,:]}}{\bar{b}_{[j]}} \cdot (x_2 - x_1) < 1 \quad \Rightarrow \quad \sigma_j > 0, \; j = 1, 2, \ldots, n_A.$$

Thus, (3.16) is a linear transformation that scales the vertices $\tilde{\hat{v}}_j$ of the dual along rays defined by $\bar{\hat{v}}_j$ as illustrated in Fig. 3.3. Therefore, in general, shifting $\mathcal{P}$ scales the corresponding polar duals. As scaling is a linear transformation, vertices are mapped to vertices and facets remain facets; this especially implies that

$$\tilde{T}_i \equiv \bar{T}_i,$$

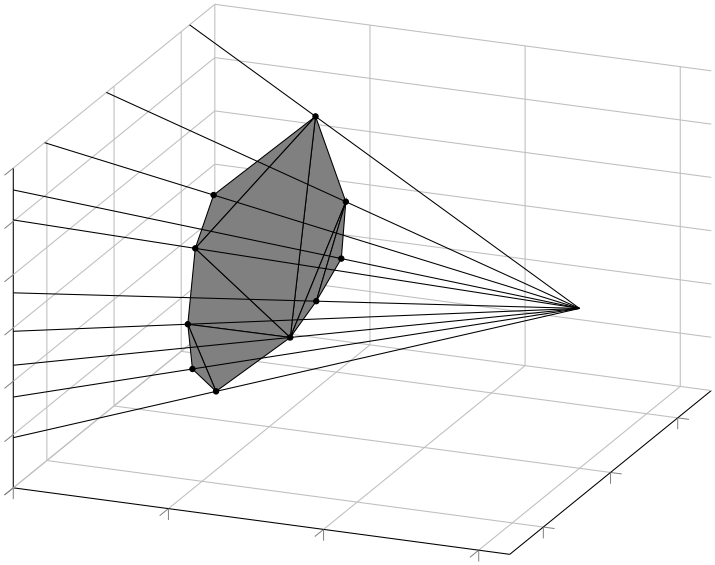


Figure 3.3.: Example: scaling the triangulation of a facet of a dual polytope along rays corresponding to $\bar{\tilde{v}}_j$ (bullets mark vertices $\tilde{\hat{v}}_j$).

as these only contain indices of the corresponding vertices. The volume of the simplices (3.13), however, is changed by scaling according to:

$$\begin{aligned}
\texttt{vol}\left(\texttt{convh}\left(\tilde{\mathcal{S}}_j \cup \{0\}\right)\right) &= \frac{1}{n_x!}\left|\det \tilde{\hat{V}}_{[S_j,:]}\right| \\
&= \frac{1}{n_x!}\left|\det\left(\texttt{diag}\left(\sigma_1, \sigma_2, \ldots, \sigma_{n_A}\right) \cdot \hat{V}\right)_{[S_j,:]}\right| \\
&= \frac{1}{n_x!}\left|\det\left(\texttt{diag}\left(\sigma_k\right)_{k\in S_j}\right) \cdot \det \hat{V}_{[S_j,:]}\right| \\
&= \texttt{vol}\left(\texttt{convh}\left(\mathcal{S}_j \cup \{0\}\right)\right) \cdot \prod_{k\in S_j} \sigma_k \\
&= \tilde{\omega}_i.
\end{aligned}$$

Thus, the volume of a scaled simplex can be related to the volume of the initial simplex, which allows to compute the weights $\tilde{\omega}_i$ and coordinates $\tilde{\lambda}_i$ as claimed.

□

The online computation of Warren's coordinates as outlined in this section is summarized in Algorithm 3.3b.

Given these algorithms for approximation of controllable sets and solutions of the planning problem, it becomes possible to assess the efficacy of the maneuver-based planning method in an example. This is provided in the following chapter and serves as the basis for a discussion of the approach.

Input $\hat{V}$, $\hat{A}$, n_v, n_A, n_x

$\omega \leftarrow 0$
for $i \in \{1, 2, \ldots, n_v\}$ **do**
  $\mathcal{F}_i \leftarrow \emptyset$, $\bar{T}_i \leftarrow \emptyset$, $\omega_i \leftarrow 0$, $\lambda_{[i]} \leftarrow 0$
  for $j \in \{1, 2, \ldots, n_A\}$ **do**
    if $\hat{V}_{[j,:]}\hat{A}^{\mathsf{T}}_{[i,:]} == 1$ **then**
      $\mathcal{F}_i \leftarrow \mathcal{F}_i \cup \{j\}$
    end if
  end for
  $\bar{T}_i \leftarrow \texttt{tri}\left(\left\{\left(\hat{V}_{[k,:]}\right)^{\mathsf{T}} | k \in \mathcal{F}_i\right\}\right)$
  for $S \in \bar{T}_i$ **do**
    $\omega_i \leftarrow \omega_i + \left|\frac{1}{n_x!} \det \hat{V}_{[S,:]}\right|$
  end for
  $\omega \leftarrow \omega + \omega_i$
end for
for $i \in \{1, 2, \ldots, n_v\}$ **do**
  $\lambda_{[i]} \leftarrow \frac{\omega_i}{\omega}$
end for
Output λ

(a) Offline computation [76].

Input $\hat{V}$, x_1, x_2, n_v, n_A, $\bar{T}_i$, $i = 1, 2, \ldots, n_v$

$\tilde{\omega} \leftarrow 0$
for $i \in \{1, 2, \ldots, n_A\}$ **do**
  $\sigma_i \leftarrow \frac{1}{1 - \hat{V}_{[i,:]} \cdot (x_2 - x_1)}$
end for
for $i \in \{1, 2, \ldots, n_v\}$ **do**
  $\tilde{\omega}_i \leftarrow 0$
  for $S \in \bar{T}_i$ **do**
    $\vartheta \leftarrow \text{LoadVolume}(S)$
    $\tilde{\omega}_i \leftarrow \tilde{\omega}_i + \vartheta \cdot \prod_{k \in S} \sigma_k$
  end for
  $\tilde{\omega} \leftarrow \tilde{\omega} + \tilde{\omega}_i$
end for
for $i \in \{1, 2, \ldots, n_v\}$ **do**
  $\tilde{\lambda}_{[i]} \leftarrow \frac{\tilde{\omega}_i}{\tilde{\omega}}$
end for
Output $\tilde{\lambda}$

(b) Extension of [76] to online computation.

Algorithm 3.3.: Online and offline versions of Warren's algorithm.

4. Examples and Discussion

The purpose of this chapter is to demonstrate the efficacy of the devised maneuver concept. To that end, at first, a simple example for the modeling of a maneuver is given, followed by a comparison of the optimization-based planning procedure and the interpolation-based one in terms of computation times. Then, in Section 4.2, the interaction with the high-level planner is exemplified. In the following, assume for simplicity:

Assumption 4.1 (Homogeneous Vehicle Dynamics)**.** *All vehicles have the same dynamics.*

While this assumption is clearly unrealistic, it is possible to extend the framework to heterogeneous dynamics, e.g. by assuming the existence of classes of similar vehicle dynamics and formulating maneuvers for combinations of different classes. While this extension is conceptually simple, it increases the number of possible maneuvers in the maneuver library (and therefore, the required design effort) such that it is deferred to future work.

4.1. Example I: A Cooperative Overtaking Maneuver

In the following, a simple overtaking maneuver as illustrated in Fig. 4.1 will be considered, where Vehicle 1 is scheduled to overtake Vehicle 2. This must be enabled by Vehicle 2 and the oncoming Vehicle 3. Focusing on modeling of this maneuver and a demonstration of the computational advantage of the interpolation-based approach over the optimization-based one, non-cooperating traffic participants are not considered until Section 4.2 for clarity of exposition.

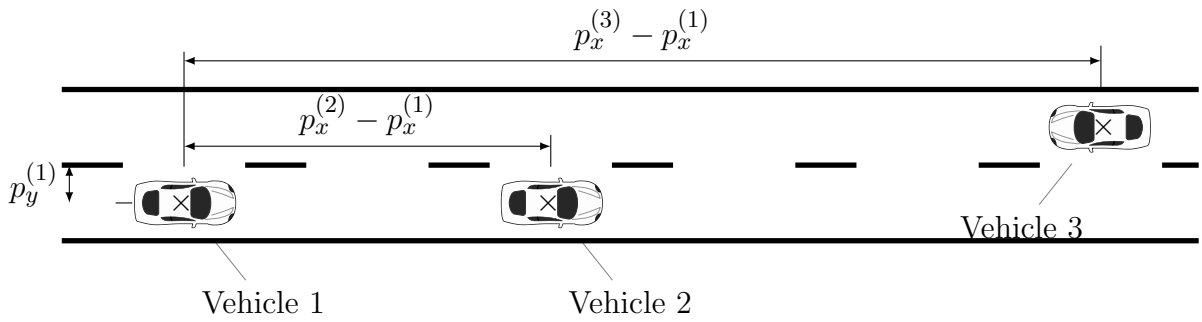


Figure 4.1.: Example Maneuver: cooperative overtaking with oncoming traffic.

4.1.1. Maneuver Formulation

Choosing the same model for all three vehicles for the generation of reference trajectories, the dynamical state of the ith vehicle, $i \in \{1, 2, 3\}$, is described by its longitudinal position and velocity, $p_x^{(i)}$ and $v_x^{(i)}$, respectively, and its lateral position and velocity, $p_y^{(i)}$ and $v_y^{(i)}$. In line with Section 2.3.2, the relative positions:

$$p_{\mathrm{rel}}^{(2)} := p_x^{(2)} - p_x^{(1)},\ p_{\mathrm{rel}}^{(3)} := p_x^{(3)} - p_x^{(1)}$$

are introduced. In addition, vehicles 2 and 3 are constrained to keep their lanes during the maneuver, such that the lateral velocities $v_y^{(2)}$ and $v_y^{(3)}$ are set to zero, leading to constant lateral positions $p_y^{(2)}$ and $p_y^{(3)}$. These simplifications reduce the dimensionality of the continuous state space $\mathbb{R}^{n_x}$ of the hybrid automaton to be defined. The resulting state vector reads:

$$x = \begin{bmatrix} p_{\mathrm{rel}}^{(2)} & p_{\mathrm{rel}}^{(3)} & p_y^{(1)} & v_x^{(1)} & v_x^{(2)} & v_x^{(3)} & v_y^{(1)} \end{bmatrix}^{\mathsf{T}}.$$

As inputs, the accelerations in the different directions are chosen, i.e., $u_x^{(i)} := \dot{v}_x^{(i)}$ and $u_y^{(i)} := \dot{v}_y^{(i)}$, such that:

$$u = \begin{bmatrix} u_x^{(1)} & u_x^{(2)} & u_x^{(3)} & u_y^{(1)} \end{bmatrix}^{\mathsf{T}}.$$

The flow function then results from zero-order hold discretization of:

$$\dot{x} = A_{\mathrm{c}} x + B_{\mathrm{c}} u, \tag{4.1}$$

with

$$A_{\mathrm{c}} = \begin{bmatrix} & -1 & 1 & 0 & 0 \\ 0_{3\times 3} & -1 & 0 & 1 & 0 \\ & 0 & 0 & 0 & 1 \\ & 0_{4\times 7} & & & \end{bmatrix},\quad B_{\mathrm{c}} = \begin{bmatrix} 0_{3\times 4} \\ I_{4\times 4} \end{bmatrix}.$$

This model uses double integrator dynamics for both longitudinal and lateral dynamics. Despite its simplicity, it has been successfully employed in planning problems, e.g. by [127, 120, 46]. This has several reasons: on the one hand, proper choice of the state and input constraints can capture many behaviours of more complex models. For example, it is possible to introduce coupling between longitudinal and lateral dynamics [120] or to approximate the so-called friction circle [101]. Furthermore, plans are to be obtained for comfortable on-road driving, which occurs in state space regions where more complex vehicle models are only mildly nonlinear, qualitatively speaking. Most importantly, in this thesis, obtained plans are not directly applied as control inputs to the vehicle, but serve as references for a tracking controller. As described in Part II, it is possible to quantify the maximum tracking error for a given model, which can then be accounted for during the maneuver formulation by providing safety distances accordingly.

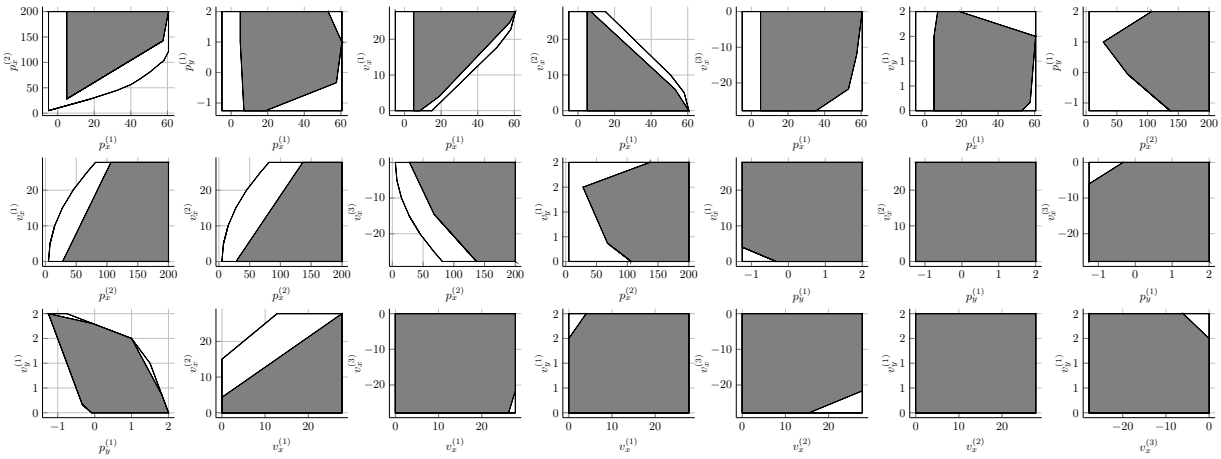


Figure 4.2.: Projections of an example 5-step controllable set: exact set (white) and approximation (gray).

The inputs are subject to the constraints:

$$u_{x,\min} \leq u_x^{(i)} \leq u_{x,\max},\ u_{y,\min} \leq u_y^{(1)} \leq u_{y,\max},$$

and the velocities are chosen such that the vehicles cannot reverse their driving direction:

$$0 \leq v_x^{(1)} \leq v_{x,\max},\ 0 \leq v_x^{(2)} \leq v_{x,\max},\ \ -v_{x,\max} \leq v_x^{(3)} \leq 0.$$

The overtaking maneuver is divided into three phases $\mathcal{Q} = \{q_1, q_2, q_3\}$: in the first, Vehicle 1 is behind Vehicle 2, in the second, Vehicle 1 overtakes and is somewhere next to Vehicle 2, and in the third phase, Vehicle 1 drives in between Vehicle 2 and the oncoming Vehicle 3, always maintaining pre-specified, constant safety distances in longitudinal and lateral direction, $l_{x,\text{safe}}$ and $l_{y,\text{safe}}$, respectively, which are assumed to be chosen such that bounded tracking errors are accounted for. Transitions are only allowed from phase 1 to phase 2 and from phase 2 to phase 3. Note that this does not allow to fall back to a prior phase, which may limit flexibility of the maneuver, but decreases computational complexity by reducing the number of admissible phase transitions. The relative positions are bounded from above and below by $p_{x,\text{rel,max}}$ and $p_{x,\text{rel,min}}$ respectively $p_{y,\text{rel,max}}$ and $p_{y,\text{rel,min}}$. For each phase, a definition of different invariant and guard sets is given in Table 4.1. Since location q_3 is the target phase of the automaton, no transitions exist from here. Once the state of the hybrid automaton has reached the target set:

$$\mathcal{X}_\text{T} = \left\{ x | x \in \text{inv}(q_3),\ 1.1 p_y^{(2)} \leq p_y^{(1)} \leq 0.9 p_y^{(2)},\ |v_y^{(1)}| \leq 0.1 \right\}$$

in phase q_3, the maneuver is completed. For numerical values of the employed parameters, cf. Table A.2.

Location	Invariant	Guard
q_1	$-\left(p_x^{(2)} - p_x^{(1)}\right) \leq l_{x,\text{safe}},$ $p_x^{(2)} - p_x^{(1)} \leq p_x^{(3)} - p_x^{(1)} - l_{x,\text{safe}},$ $p_x^{(3)} - p_x^{(1)} \leq p_{x,\text{rel,max}},$ $p_{y,\text{rel,min}} \leq p_y^{(1)} \leq p_{y,\text{rel,max}},$ $0 \leq v_y^{(1)} \leq v_{y,\max}.$	$g(\theta_{1,2}) = \{x \mid p_x^{(2)} - p_x^{(1)} \leq l_{x,\text{safe}}\}$
q_2	$p_{x,\text{rel,min}} \leq p_x^{(2)} - p_x^{(1)} \leq l_{x,\text{safe}},$ $l_{x,\text{safe}} \leq p_x^{(3)} - p_x^{(1)} \leq p_{x,\text{rel,max}},$ $p_y^{(2)} + l_{y,\text{safe}} \leq p_y^{(1)} \leq p_{y,\text{rel,max}},$ $v_{y,\min} \leq v_y^{(1)} \leq v_{y,\max}.$	$g(\theta_{2,3}) = \{x \mid p_x^{(2)} - p_x^{(1)} \leq -l_{x,\text{safe}}\}.$
q_3	$p_{x,\text{rel,min}} \leq p_x^{(2)} - p_x^{(1)} \leq -l_{x,\text{safe}},$ $l_{x,\text{safe}} \leq p_x^{(3)} - p_x^{(1)} \leq p_{x,\text{rel,max}},$ $p_{y,\min} \leq p_y^{(1)} \leq p_{y,\max},$ $v_{y,\min} \leq v_y^{(1)} \leq 0.$	—

Table 4.1.: Definition of the locations used to model the maneuver.

4.1.2. Example Results: Controllable Sets and Trajectories

In order to set up a planning problem based on this maneuver, the parameters of the cost function (2.25) must be chosen. The matrix C_z is defined as:

$$C_z = \begin{bmatrix} 0_{5\times 2} & I_{5\times 5} \end{bmatrix},$$

while $C_1 = I$, $D_1 = I$. As reference values in $\bar{z}$, a longitudinal speed of $20\,\mathrm{m\,s^{-1}}$ is defined for all vehicles, while the reference for the lateral speed and position of the overtaking vehicle is zero. These values, along with a planning horizon and constraints (2.24) derived from the maneuver definition, define the planning problem.

The given definition of the hybrid automaton for the overtaking maneuver allows to determine controllable sets for the target set $\mathcal{X}_\mathrm{T}$, containing states for which a solution to the planning problem exists. An example projection of such a 7-dimensional set on $\mathbb{R}^2$ is shown in Fig. 4.2, along with a projection of an approximation obtained by the procedure described in Section 3.1. The quality of a given approximation can be quantified by the

difference in volume or the Hausdorff distance to the original set; however, in light of Proposition 3.3, such values are only of interest during computation of the approximation, as it can be made arbitrarily precise at the price of an increase in complexity. This indicates a trade-off, to which a compromise must be found during the design phase: while complexity should be low, the volume of the approximated controllable set should be as large as possible, where the maximum is given by the exact controllable set.

Despite the need to compromise, approximations are nonetheless necessary: because the number of facets of j-step controllable sets and therefore computation time grow exponentially as j increases, an exact (within numerical tolerances) computation of controllable sets is generally only possible for a few steps. Because controllable sets are intended for feasibility assessment of planning problems, this also limits the planning problems to only a few time steps. Maneuvers require a certain amount of time for completion; if this time span is divided into only a few steps, a long sampling time T_s results, which can threaten safety because the inter-sample behavior is not accounted for in the discrete-time planning problem. This issue is exemplified in Fig. 4.3 and Fig. 4.4, where position and velocity trajectories for the three vehicles resulting during an overtaking maneuver are shown. The left plot in Fig. 4.3 is based on a sampling time $T_\mathrm{s} = 1\,\mathrm{s}$ for zero-order hold discretization of (4.1), which allows to complete the maneuver in 5 steps (being the maximum number of steps for which exact controllable sets could be computed in this case, cf. Figure 4.2). The right plot in Fig. 4.3, in contrast, results from approximations as described in Section 3.1, which allow for longer planning horizons and shorter sampling times because the number of time steps is less critical in the approximation scheme.

Despite this difference, in both cases, Vehicle 1 overtakes Vehicle 2 and avoids collision with Vehicle 3 as intended, illustrating the capability of the maneuver concept to reliably ensure execution of a maneuver. This is a major advantage over approaches which model vehicles by bounding boxes, e.g. [120]; in these approaches, a certain driving behavior can only be *stimulated* by tuning of the cost function and proper choice of reference values, but it cannot be guaranteed to result for all admissible initial states. Furthermore, these admissible initial states cannot even be computed in advance in the bounding box approach. Thus, when planning online, it is neither known whether an intended maneuver is feasible for the given initial state, nor if the planning algorithm will return a corresponding reference trajectory. The maneuver approach resolves both issues.

These drawbacks of the bounding box approach make it difficult to compare its computation times to those of the maneuver approach because it will often return results which do not comply with the desired behavior of the vehicles. A better indicator of computational complexity is the combinatorial complexity of the binary variables used in a specific planning problem formulation. Note, however, that combinatorial complexity does not completely determine the resulting computation times, as it is sometimes possible to obtain reductions by *adding* further binary variables [151]. Also, highly performant solvers such as [72, 63] make use of heuristics, whose success may differ for two different problem formulations.

The bounding box approach requires one binary variable per side of the bounding box per vehicle. For the considered example scenario, a number of five binary variables (three

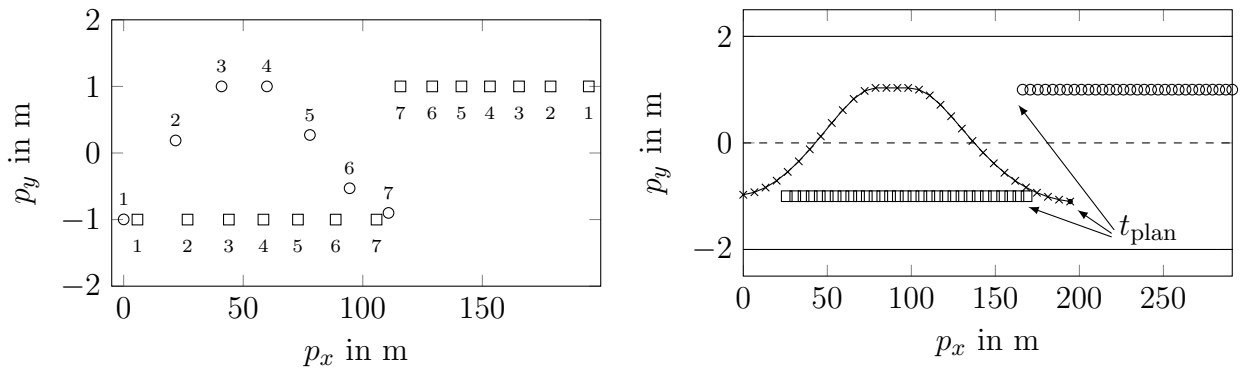


Figure 4.3.: Example position trajectories of the three vehicles during overtaking: optimization (left) vs. approximation (right) (numbers indicate time steps).

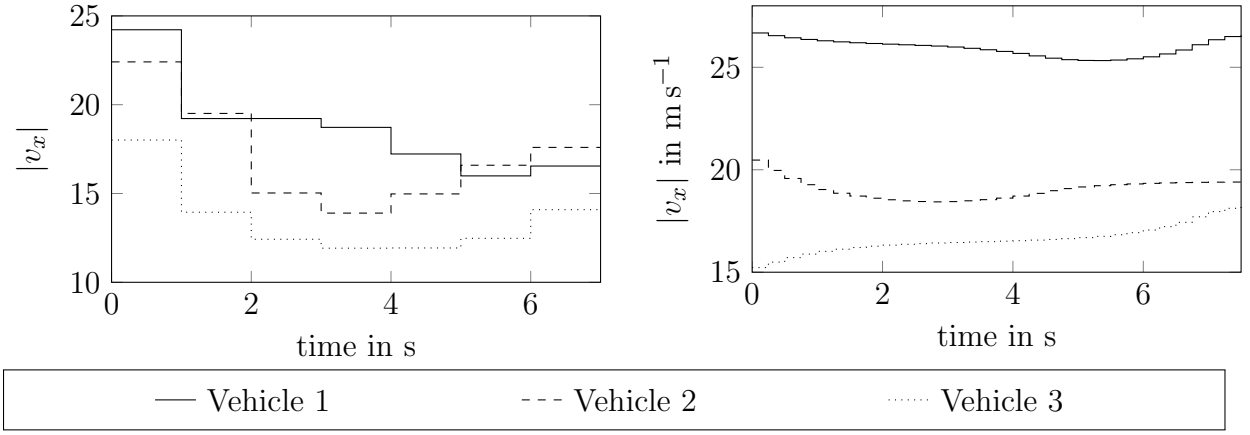


Figure 4.4.: Example velocity magnitude trajectories.

for bounding vehicle 2, two for vehicle 3) is reasonable. Since no temporal constraints exist among these variables (arbitrary switches between two consecutive time instants are allowed), a total of $2^{(5 \cdot H_{\text{plan}})}$ different combinations exists. The proposed method, in contrast, only requires two binary variables to implement the optimal control problem via (2.24), which are in addition connected by temporal constraints, namely, the switching order imposed by the set of allowed transitions Θ. Since the sequence in which the locations are traversed is known, only the switching times must be determined. For at most two switching times, there exist $\begin{pmatrix} H_{\text{plan}} \\ 2 \end{pmatrix}$ possibilities, which is significantly less than the above mentioned $2^{(5 \cdot H_{\text{plan}})}$ combinations.

The cooperative nature of the maneuver becomes clear by the fact that – even though the vehicles try to maintain their reference speed – Vehicle 2 and Vehicle 3 brake cooperatively in order to facilitate overtaking of Vehicle 1, cf. Figure 4.4. In this example, a rather tight longitudinal safety distance of $l_{x,\text{safe}} = 5\,\text{m}$ was prescribed between the vehicle center points, which could of course be increased.

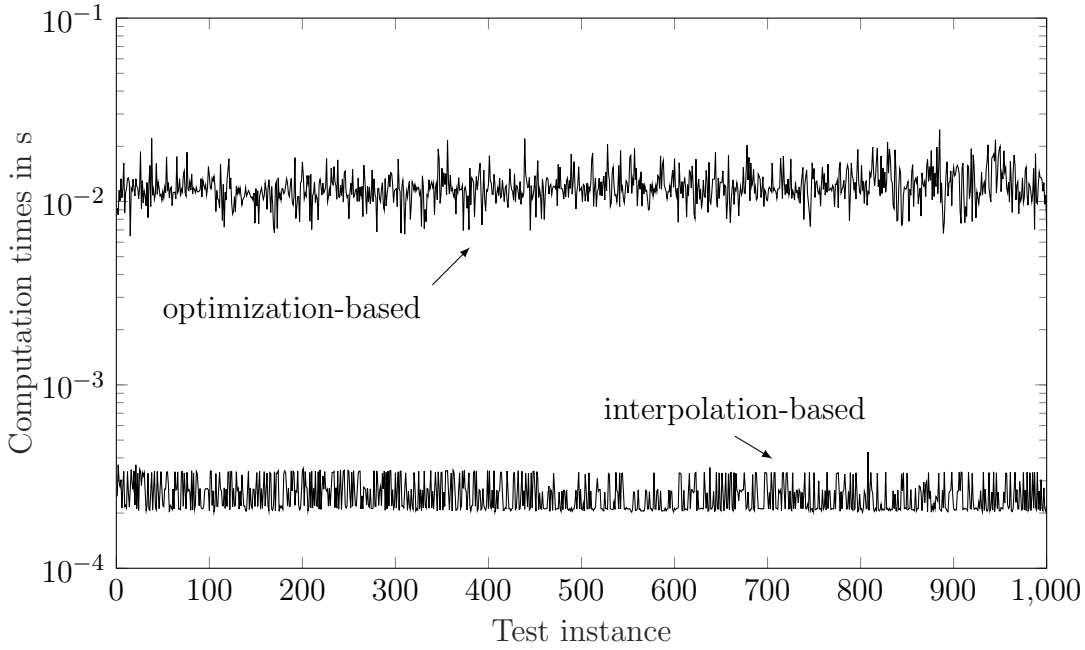


Figure 4.5.: Comparison of run-times.

While a comparison between the bounding box and the maneuver approach is methodically difficult, it is simple to compare computation times for the optimization-based and the approximated solution of a maneuver-based planning problem. This comparison was carried out for a large number of initial states randomly chosen from an entry set obtained as solution to Problem 3.1.

Computation of the barycentric coordinates λ was facilitated based on triangulation as discussed in Section 3.2.1, where a total of $n_s = 4858$ simplices resulted from triangulation of the considered entry set. The resulting run-times in milliseconds are shown in Fig. 4.5. The scenario was implemented in Matlab using Yalmip [96] and was run on a PC with an Intel i7 CPU (8 cores, each 3.4 GHz) and 16 GB RAM. Clearly, owing to the non-real time nature of the employed operating system, these run-times mainly serve as an illustrative indicator. While the optimization-based procedure (using Gurobi [63]) requires about 12.11 ms on average, the interpolation-based method only requires a mean of about 0.25 ms.

4.2. Example II: Highway Entry of Autonomous Vehicles

The example in the previous section focused on a single maneuver without considering non-cooperating traffic participants. This section, in contrast, demonstrates the ability of the maneuver concept to enable safe interaction between cooperating and non-cooperating traffic participants as well as efficient scheduling of different maneuvers. To that end, the scenario shown in Fig. 4.6 is considered, in which autonomous vehicles are driving on a highway (left lane in driving direction), while other autonomous vehicles are trying to

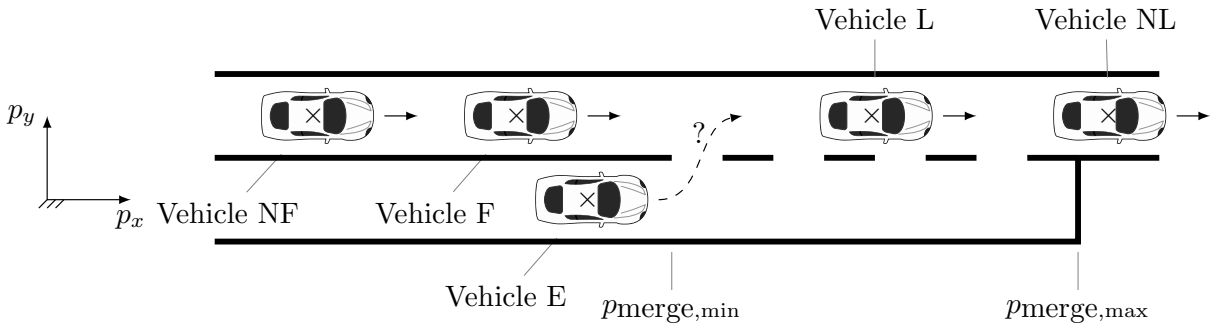


Figure 4.6.: Highway Entry: Where and when to merge?

merge from a merging lane (right lane). The following rules are imposed: a minimal velocity $v_{x,\text{min,left}} > 0$ is enforced on the highway (as is common e.g. on German highways). The merge lane enables vehicles to accelerate such that, at the end of the merging process, this constraint is fulfilled. Merging must take place between longitudinal coordinates $p_{\text{merge,min}}$ and $p_{\text{merge,max}}$. If it should be impossible because no gap opens, a vehicle must stop on the merge lane in a position that allows to sufficiently accelerate prior to merging later on.

The scenario is to be controlled by the hierarchical framework as depicted in Figure 2.1. Regarding the infrastructure setup, similar assumptions are made as in [159] for intersection control: At first, to limit the scope of the example, assume that only autonomous vehicles are driving on the road. Then, assume that the high-level controller is part of the infrastructure, i.e., a road side unit, and controls a certain section (the *control zone*) of the road up- and downstream (1.5 km in each direction). Events outside this zone will not be considered for simplicity. This setting allows to focus on the basic functionality of the approach without considering questions of decentralized computations, inter-vehicle communication, or changing road topology.

Focusing on feasibility, the high-level controller implements the following simple rules for maneuver scheduling: all vehicles on the merge lane except for the first one are to keep the lane. For the first vehicle, it is assessed whether a feasible merging maneuver exists in the maneuver library of the high-level controller, depending on the vehicles currently on the highway. If this is the case, the maneuver with shortest duration H_{plan} is executed. Otherwise, the vehicle must keep the lane until merging becomes possible. Note that both the gap into which and the time at which merging occurs are determined online by the high-level controller, i.e., these are not fixed a priori.

In order to compute reference trajectories for a vehicle i, affine system equations are used:

$$\dot{x}^{(i)}(t) = A_{\text{c}} x^{(i)}(t) + B_{\text{c}} u^{(i)}(t), \tag{4.2}$$

with the state and input vectors:

$$x^{(i)} = \begin{bmatrix} p_x^{(i)} & p_y^{(i)} & v_x^{(i)} & v_y^{(i)} \end{bmatrix}^{\mathsf{T}}, \; u^{(i)} = \begin{bmatrix} u_x^{(i)} & u_y^{(i)} \end{bmatrix}^{\mathsf{T}}$$

Table 4.2.: Parameter values used in the case study.

$u_{x,\min} = -3\,\mathrm{m\,s^{-1}}$	$p_{\mathrm{merge},\min} = 200\,\mathrm{m}$	$v_{x,\min} = 0\,\mathrm{m\,s^{-1}}$
$u_{x,\max} = 3\,\mathrm{m\,s^{-1}}$	$p_{\mathrm{merge},\max} = 400\,\mathrm{m}$	$v_{x,\max} = 33.3\,\mathrm{m\,s^{-1}}$
$u_{y,\min} = -3\,\mathrm{m\,s^{-1}}$	$v_{x,\mathrm{min,left}} = 22.2\,\mathrm{m\,s^{-1}}$	$v_{y,\min} = 0\,\mathrm{m\,s^{-1}}$
$u_{y,\max} = 3\,\mathrm{m\,s^{-1}}$	$p_{x,\min} = -\infty\ \mathrm{m}$	$v_{y,\max} = 5.56\,\mathrm{m\,s^{-1}}$
$l_{x,\mathrm{safe}} = 5\,\mathrm{m}$	$p_{x,\max} = \infty\ \mathrm{m}$	$T_\mathrm{s} = 0.5\,\mathrm{s}$

and – just as in Section 4.1 – double integrator dynamics in longitudinal and lateral direction, such that:

$$A_\mathrm{c} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ & & 0_{2\times 4} & \end{bmatrix},\ B_\mathrm{c} = \begin{bmatrix} 0_{2\times 2} \\ I_{2\times 2} \end{bmatrix}. \tag{4.3}$$

Zero-order hold discretization using sampling time T_s – which also determines the frequency at which the high-level controller operates – and appropriate, maneuver-dependent combination with the dynamics of other vehicles then give the flow function of the corresponding hybrid automaton as in Definition 2.2. States and inputs are constrained by polyhedral sets:

$$x^{(i)} \in \mathcal{X}^{(i)},\ u^{(i)} \in \mathcal{U}^{(i)}. \tag{4.4}$$

In the following, let

$$\mathcal{U}^{(i)} = [u_{x,\min}, u_{x,\max}] \times [u_{y,\min}, u_{y,\max}]$$

and

$$\mathcal{X}^{(i)} = [p_{x,\min}, p_{x,\max}] \times [p_{y,\min}, p_{y,\max}] \times [v_{x,\min}, v_{x,\max}] \times [v_{y,\min}, v_{y,\max}].$$

Parameter values used in this case study are given in Tab. 4.2.

4.2.1. Maneuver Formulations

In order to enable collision-free interaction of all vehicles on the highway and the merge lane, the high-level controller may choose between several maneuvers, which are formulated in the following.

Lane Change: Single Vehicle

The first maneuver defines the admissible behavior of a single vehicle during merging (which reduces to a simple lane change in the absence of other vehicles). Despite its simplicity, this maneuver is important because it allows to characterize those longitudinal positions and

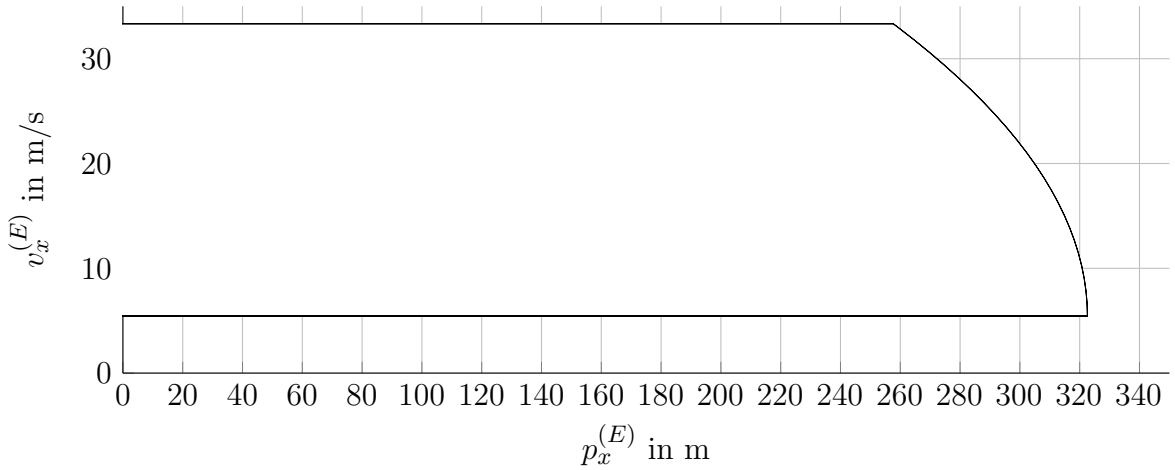


Figure 4.7.: Projection of the control invariant set $\mathcal{X}_{\text{merge}}$.

velocities of a vehicle on the merge lane for which merging is *physically* possible, which is also a necessary condition for successful completion of all merging maneuvers which involve other vehicles. Starting on the merge lane with zero lateral velocity:

$$\mathcal{X}_0 := \left\{ x^{(i)} \in \mathcal{X}^{(i)} | p_y^{(i)} = p_{y,\min}, v_y^{(i)} = 0, p_{\text{merge,min}} \leq p_x^{(i)} \leq p_{\text{merge,max}} \right\},$$

the maneuver is complete if the vehicle has merged on the highway before reaching the end of the merge lane:

$$\mathcal{X}_{\text{T}} := \left\{ x^{(i)} \in \mathcal{X}^{(i)} | p_y^{(i)} = p_{y,\max}, v_y^{(i)} = 0, p_{\text{merge,min}} \leq p_x^{(i)} \leq p_{\text{merge,max}} \right\}. \tag{4.5}$$

The maneuver can be modeled to consist of only one phase q_0 with $\text{inv}(q_0) = \mathcal{X}^{(i)}$, such that no guards and transitions must be defined. For this maneuver, the computation of controllable sets based on the recursion in Definition 2.5 can be observed to converge to a set $\mathcal{X}_{\text{merge}}$, i.e.,

$$\mathcal{X}_{\text{merge}} := \mathcal{K}_j, \text{ where } \texttt{pre}(\mathcal{K}_j) = \mathcal{K}_j, \ j \in \mathbb{N}.$$

This makes $\mathcal{X}_{\text{merge}}$ a control invariant set. Its projection on the space of longitudinal position and velocity is shown in Fig. 4.7. Enforcing $x \in \mathcal{X}_{\text{merge}}$ as long as no merging maneuver is scheduled has the effect that vehicles will automatically adapt their speed on the merge lane and even come to standstill if required in order to be able to merge sometime in the future; no special braking maneuver is required.

Lane Keeping: Single Vehicle

Another simple maneuver defines the behavior of a single vehicle which is driving on a lane without any other vehicles in the closer surrounding. The corresponding hybrid automaton contains only one phase q_0 and therefore no guards or transitions. Inputs,

states, dynamics, and input and state constraints correspond to those defined above, while the invariant (equivalent to the initial set $\mathcal{X}_0$) depends on the lane the vehicle is driving on: on the left lane,

$$\text{inv}(q_0) = \left\{ x \in \mathcal{X}^{(i)} | v_y^{(i)} = 0,\ p_y^{(i)} = p_{y,\max},\ v_x^{(i)} \geq v_{x,\min,\text{left}} \right\},$$

while on the right (merge) lane, the vehicle's state must also lie in the control invariant set $\mathcal{X}_{\text{merge}}$ from (4.5):

$$\text{inv}(q_0) = \left\{ x \in \mathcal{X}^{(i)} \cap \mathcal{X}_{\text{merge}},\ v_y^{(i)} = 0,\ p_y^{(i)} = p_{y,\min} \right\},$$

where it must be ensured by the designer that the intersection of the sets is non-empty.

Lane Keeping: Two Vehicles

A slightly more complex maneuver defines the behavior of a vehicle E driving behind a non-cooperating vehicle NL, while both keep the lane (similar to Fig. 2.6, with E $\triangleq$ F and NL $\triangleq$ L). The hybrid automaton of this maneuver also consists of only one phase q_0 and no transitions or guard sets. The dynamics now also incorporates those of NL, leading to the combined state vector $x = \left[x^{(\text{E})^\mathsf{T}} \quad x^{(\text{NL})^\mathsf{T}} \right]^\mathsf{T}$, while the input vector only consists of $u_x^{(\text{E})} \in \mathcal{U}^{(\text{E})}$ because the inputs $u_x^{(\text{NL})} \in \mathcal{U}^{(\text{NL})}$ of NL cannot be controlled within the maneuver, such that $\mathcal{U} = \mathcal{U}^{(\text{E})}$ and $\mathcal{W} = \mathcal{X}^{(\text{E})} \times \mathcal{X}^{(\text{NL})} \times \mathcal{U}^{(\text{NL})}$. For planning, conservative assumptions for the behavior of NL according to Sec. 2.3.1 are used. Similar to the single vehicle's lane keeping maneuver, the invariant depends on the lane the vehicles are driving on. Generally, $\mathcal{X}_0 = \text{inv}(q_0)$, with

$$\text{inv}(q_0) = \left\{ x \in \mathcal{X}^{(\text{E})} \times \mathcal{X}^{(\text{NL})},\ v_y^{(\text{E})} = 0, \left[p_x^{(\text{NL})} - p_x^{(\text{E})} \quad v_x^{(\text{E})} \quad v_x^{(\text{NL})} \right]^\mathsf{T} \in \mathcal{X}_{\text{safe}},$$
$$p_y^{(i)} = \begin{cases} p_{y,\max} \text{ (left)} \\ p_{y,\min} \text{ (right)} \end{cases},\ v_x^{(\text{E})} \geq \begin{cases} v_{x,\min,\text{left}} \text{ (left)} \\ v_{x,\min} \text{ (right)} \end{cases} \right\},$$

where "left" and "right" refer to the respective lanes.

Cooperative Merging

The very core of the maneuver library in this example is a cooperative merging maneuver. It defines roles for five vehicles, cf. Fig. 4.6: the non-cooperating leading and following vehicles NL and NF, respectively, the vehicle E which is to merge, and the vehicles cooperating with it, L and F. Note that E is only allowed to merge between F and L, while NF and NL are incorporated into the maneuver formulation in order to model the interaction between the cooperating vehicles and their surrounding according to Sec. 2.3.1, i.e., $\mathcal{C} = \{\text{E}, \text{F}, \text{L}\}$, and $\mathcal{N} = \{\text{NF}, \text{NL}\}$. The associated hybrid automaton consists of three phases $\mathcal{Q} = \{q_0, q_1, q_2\}$ and has the topology as shown in Fig. 2.4, i.e., allows transitions from q_0 to q_1 and from q_1 to q_2. The phases correspond to: 1) E driving on the right lane

with zero lateral velocity, 2) E changing lanes, not having reached the end of the acceleration lane $p_{\text{merge,max}}$ yet, and 3) E having passed $p_{\text{merge,max}}$ (which is only admissible after having merged onto the highway).

Based on (4.2), the continuous state vector of the automaton combines the states of the involved vehicles in:

$$x = \begin{bmatrix} x^{(\text{NF})^\mathsf{T}} & x^{(\text{F})^\mathsf{T}} & x^{(\text{L})^\mathsf{T}} & x^{(\text{NL})^\mathsf{T}} & x^{(\text{E})^\mathsf{T}} \end{bmatrix} \in \mathcal{X} := \mathcal{X}^{(\text{NF})} \times \mathcal{X}^{(\text{F})} \times \mathcal{X}^{(\text{L})} \times \mathcal{X}^{(\text{NL})} \times \mathcal{X}^{(\text{E})}. \tag{4.6}$$

The corresponding input vector is:

$$u = \begin{bmatrix} u^{(\text{F})^\mathsf{T}} & u^{(\text{L})^\mathsf{T}} & u^{(\text{E})^\mathsf{T}} \end{bmatrix}^\mathsf{T} \in \mathcal{U} := \mathcal{U}^{(\text{F})} \times \mathcal{U}^{(\text{L})} \times \mathcal{U}^{(\text{E})}. \tag{4.7}$$

Note that the inputs of the non-cooperating vehicles NF and NL are not contained because they are beyond control. Rather,

$$w = \begin{bmatrix} u_x^{(\text{NF}^\mathsf{T})} & u_x^{(\text{NL}^\mathsf{T})} \end{bmatrix}^\mathsf{T},$$

with

$$\begin{bmatrix} x \\ w \end{bmatrix} \in \mathcal{W} := \mathcal{X} \times \mathcal{U}^{(\text{NF})} \times \mathcal{U}^{(\text{NL})},$$

where conservative predictions according to Section 2.3.1 are used for w. The flow function in Definition 2.2 results from combining each vehicle's dynamics (4.2) appropriately according to the state vector (4.6), the input vector (4.7), and the conservative predictions. The state constraints in each phase combine: 1) general constraint sets $\mathcal{X}^{(i)}$ on the dynamics of single vehicles with 2) safety constraint sets $\mathcal{X}_{\text{safe}}$ in the state spaces of pairs of vehicles, and 3) location-dependent constraints. Let:

$$\mathcal{X}_Y = \left\{ x \in \mathcal{X} | v_y^{(i)} = 0,\ p_y^{(i)} = p_{y,\max} \right\},\ i \in \{\text{NF}, \text{F}, \text{L}, \text{NL}\},$$

and define $\tilde{\mathcal{X}} := (\mathcal{X} \times \mathcal{X} \times \mathcal{X} \times \mathcal{X} \times \mathcal{X}) \cap \mathcal{X}_Y$. Denote the projection on the state space of vehicle i by $\texttt{proj}_{(i)}$, with $i \in \{\text{NF}, \text{F}, \text{L}, \text{NL}\}$. Then:

$$\begin{aligned}
\text{inv}(q_0) = \big\{ & x | x \in \tilde{\mathcal{X}},\ \texttt{proj}_{(\text{NL,L})}(x) \in \mathcal{X}_{\text{safe}}, \\
& \texttt{proj}_{(\text{L,F})}(x) \in \mathcal{X}_{\text{safe}},\ \texttt{proj}_{(\text{F,NF})}(x) \in \mathcal{X}_{\text{safe}}, \\
& v_x^{(\text{NF})} = v_{x,\max},\ v_x^{(\text{NL})} = v_{x,\min} \big\}, \\
\text{inv}(q_1) = \big\{ & x | x \in \tilde{\mathcal{X}},\ \texttt{proj}_{(\text{NL,L})}(x) \in \mathcal{X}_{\text{safe}}, \\
& \texttt{proj}_{(\text{L,E})}(x) \in \mathcal{X}_{\text{safe}},\ \texttt{proj}_{(\text{E,F})}(x) \in \mathcal{X}_{\text{safe}}, \\
& \texttt{proj}_{(\text{F,NF})}(x) \in \mathcal{X}_{\text{safe}}, p_x^{(\text{E})} \geq p_{\text{merge,min}}, \\
& v_x^{(\text{NF})} = v_{x,\max},\ v_x^{(\text{NL})} = v_{x,\min} \big\}, \\
\text{inv}(q_2) = \big\{ & x | x \in \text{inv}(q_1),\ p_y^{(\text{E})} = p_{y,\max},\ v_y^{(\text{E})} = 0 \big\}.
\end{aligned}$$

The initial set is

$$\mathcal{X}_0 = \{x \in \mathrm{inv}(q_0) : p_y^{(\mathrm{E})} = p_{y,\min}, v_y^{(\mathrm{E})} = 0\},$$

and the target set reads:

$$\mathcal{X}_\mathrm{T} = \left\{ x | p_y^{(\mathrm{E})} = p_{y,\max},\ v_x^{(\mathrm{E})} \geq v_{x,\min,\mathrm{left}}, v_y^{(\mathrm{E})} = 0 \right\},$$

which enforces that E is driving on the left lane with zero lateral velocity and longitudinal velocity above $v_{x,\min,\mathrm{left}}$. The guard sets corresponding to the transitions are:

$$g(\theta_{0,1}) = \left\{ x \in \tilde{\mathcal{X}} | p_y^{(\mathrm{E})} > p_{y,\min} \right\},\ g(\theta_{1,2}) = \left\{ x \in \tilde{\mathcal{X}} | p_x^{(\mathrm{E})} > p_{\mathrm{merge,max}} \right\}.$$

From this cooperative merging maneuver, other merging maneuvers can be derived by omitting vehicles and their states from the hybrid automaton and adapting its formulation accordingly. This allows to cover situations in which fewer vehicles are present than depicted in Fig. 4.6, thus making the maneuver library more flexible. For example, a non-cooperative merging maneuver which only considers E and a single non-cooperating vehicle (either NF or NL is defined, depending on whether E should merge before another vehicle or behind). This maneuver is non-cooperative in the sense that E must plan without other vehicles adapting their behavior, only relying on the conservative treatment as described in Sec. 2.3.1.

4.2.2. Simulation Results

A Matlab-based simulation environment has been implemented in order to analyze the effectiveness of the proposed framework. The test setup was chosen as follows: vehicles are generated every t seconds, where t is randomly chosen anew every time a vehicle has been generated, with $t \in [3, 5]$ s (left lane) and $t \in [3, 4]$ s (right lane). The initial longitudinal position is set to 0 and the lateral position to the respective lane center with zero lateral velocity. Longitudinal velocities are chosen in compliance with velocity-dependent safety constraints $\mathcal{X}_\mathrm{safe}$ to a preceding vehicle, but are completely random apart from that (right lane) or as close to $100\,\mathrm{km\,h^{-1}}$ as possible (left lane).

The simulation was run for 40 s, during which five vehicles were generated on the left lane and three on the right lane. Fig. 4.8 shows the final constellation of the vehicles, where the numbering reflects the order of their generation (2, 4, and 7 started on the merge lane). The emerging behavior is as follows: after 19 s, Vehicle 2 executes a cooperative merging maneuver with $H_\mathrm{plan} = 13$, where the vehicles $\mathcal{C} = \{2, 5, 6\}$ are cooperating, while the vehicles $\mathcal{N} = \{3, 8\}$ are considered as non-cooperative. The resulting longitudinal velocities of the cooperating vehicles 2, 5, and 6 are shown in Fig. 4.9. The cooperative nature of the maneuver is illustrated by the fact that the leading vehicle 5 accelerates slightly in order to allow vehicle 2 to merge behind it. The following vehicle 6, on the other hand, is so far behind that it does not need to adapt its velocity to open a gap. Vehicles 4 and 7 are unable to find a gap into which they can merge cooperatively, such

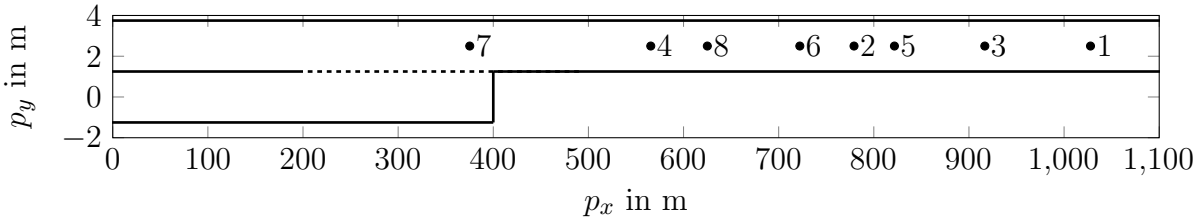


Figure 4.8.: Constellation after 40 s of simulation (bullets mark vehicles; numbers indicate order of their generation).

that they wait until all vehicles have passed and merge behind 9.5 s respectively 14.5 s after having appeared on the map, with $H_{\text{plan}} = 18$ and $H_{\text{plan}} = 14$, respectively. These maneuvers take so long because they already include the process of slowing down and waiting on the merge lane. The lateral velocities of the merging vehicles 2, 4, and 7 are given in Fig. 4.11, while Fig. 4.10 compares the actual longitudinal distances for a selected pair of vehicles to the constraints resulting from Proposition 2.2 and those resulting from the approximation $\mathcal{X}_{\text{safe}}$ as in Proposition 2.3. The plot shows that the constraints are never violated and demonstrates that the approximation does not lead to overly cautious driving.

The fact that vehicles 4 and 7 cannot find a gap to merge between the other vehicles can be attributed to three major reasons: on the one hand, the constellation of the involved vehicles might simply be such that safe merging is impossible. On the other hand, however, it is possible that the entry sets of the maneuvers contained in the maneuver library are too small, either because the planning horizons have not been chosen long enough, or because approximations of the actual controllable sets are too coarse. Third, it is possible that the number of different maneuvers in the library is too small. While in the given example, an alternative maneuver (merge behind all other vehicles) could be found, in general, it could happen that no maneuver is admissible at all, which would force the affected vehicles to emergency brake. Thus, the design of the maneuver library is of paramount importance, aiming to prevent such situations.

Having analyzed the efficacy of the planning method in the two examples considered in this chapter, the focus of the thesis now shifts towards the task of tracking a planned trajectory, which is considered in the following part.

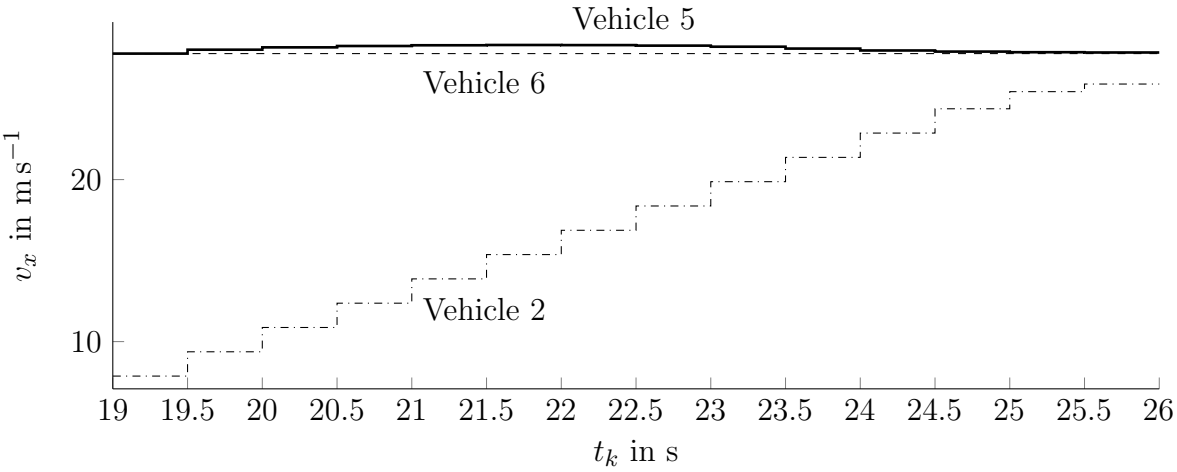


Figure 4.9.: Longitudinal velocities of cooperating vehicles during merging of vehicle 2.

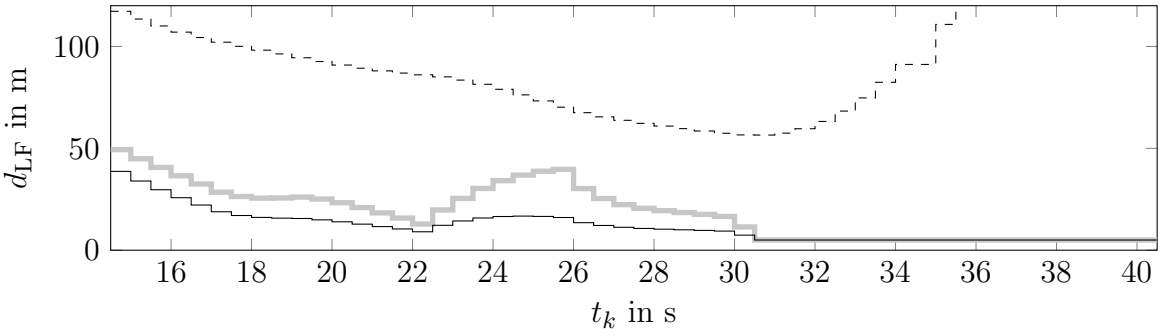


Figure 4.10.: Longitudinal distance (dashed) and required safety distance (gray: conservative; solid: exact) for vehicle pairing: L = 4, F = 7.

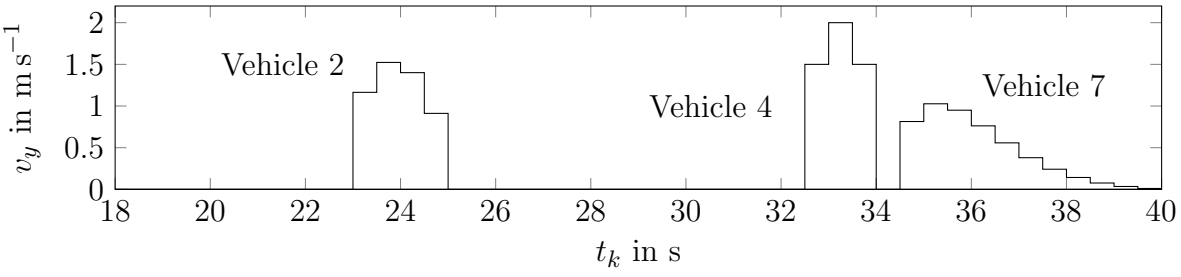


Figure 4.11.: Lateral speed of vehicles 2, 4, and 7.

Part II.

Low-level Tracking Control

5. A Novel Approach to Trajectory Tracking with Guarantees

Feedback controllers are typically deployed in order to stabilize a system, to impose a desired behavior, and to counteract the effect of disturbances. The trajectory planning layer of Part I introduces feedback by frequent re-planning and therefore allows to react to changes in the environment. However, this does not ensure that a vehicle actually follows a planned trajectory, owing to a mismatch between the typically simpler dynamic models used for planning and the actual vehicle dynamics. Also, the inputs of the planning model correspond to different physical quantities than the input signals of the actual vehicle, such that the input signals from the trajectory planner simply cannot be directly applied to a vehicle. Therefore, in the proposed framework, the planning algorithm is complemented by a low-level trajectory tracking controller, which is based on a more detailed vehicle model and enables a vehicle to closely follow a planned trajectory.

In addition to the model mismatch, deviations from a planned behavior can result from external disturbances such as wind gusts, varying road surface conditions, or slope of the road. While the following derivations could in principle be extended to account for these effects, detailed disturbance models, parameters, and corresponding assumptions would be required, lying beyond the scope of this thesis. Finally, the discrete-time implementation of a continuous-time controller always leads to deviations at the inter-sample times, being a further source of disturbance. As it is common practice, this problem is adressed by a sufficiently high control frequency, which poses no challenge from a computational point of view for the tracking controller to be derived subsequently.

The mismatch between the planned and the resulting behavior of a vehicle will be referred to as *tracking error* in the following. Consistency between the planning and tracking layers requires a characterization of the set of reference trajectories for which guarantees about a bound on the magnitude of the tracking error can be given. Such a guarantee allows to make provisions for tracking errors during the design phase by including sufficient safety margins in the maneuver formulations of the trajectory planning problem. This enables the trajectory planner to plan trajectories which can be tracked with an error small enough to not put safety at risk.

The general functionality of the control algorithm to be used in the proposed framework is described in this chapter. Related to the boundedness of the tracking error is the question of the existence of admissible control inputs, which is considered in Chapter 6. The synthesis of the controller parameters and analysis of the resulting closed-loop dynamics is detailed in Chapter 7, while the efficacy of the approach is illustrated by simulation examples in Chapter 8, followed by a discussion of the overall approach.

5.1. Related Work on Trajectory Tracking

As one of the fundamental challenges in control theory, the task of tracking time-varying reference values has gained considerable attention, cf. [80] for an introduction. Considering the control of autonomous vehicles, different notions of the reference tracking problem exist. While no strict categorization exists, the following terminology is widely used: the so-called *lateral control* or *path following* task is concerned with the control of a vehicle along a path. As mentioned in Part I, a path is understood as an ordered set of coordinates without temporal information, such that the longitudinal position along the path is not prescribed, cf. [41, 4, 158]. It typically finds application in control of slowly-moving vehicles such as autonomous underwater vehicles [87, 3] for which collision avoidance with moving obstacles is not as relevant as in on-road traffic.

In the following, the more complex problem of *trajectory tracking* is considered, which not only prescribes reference positions, but also corresponding points in time. This imposes stricter requirements, which are, however, necessary in the presence of moving obstacles, where collision avoidance requires information about an obstacle's position at a certain time.

Several aspects contribute to the complexity of the tracking task in the considered context: first, even after simplifying assumptions, the vehicle dynamics is nonlinear [121, 114]. Furthermore, constraints on the states and inputs of the controlled vehicles must be considered. This typically motivates the use of model-predictive control (MPC), which allows to explicitly account for constraints. This advantage comes at a price: as vehicles have nonlinear dynamics, computation of inputs by optimization of predicted values in general leads to nonlinear optimization problems. These are comparably expensive to solve and lack guarantees of convergence to a globally optimal solution due to their non-convex nature [25]. Nonetheless, several approaches to the tracking problem based on nonlinear MPC exist in the vehicle context [50, 62, 53]. An alternative [51, 78, 64, 45, 94, 92, 154] to the direct use of nonlinear vehicle models within the online optimization is the linearization of these models around the initial state or along a given reference trajectory. This leads to a linear, time-varying model and inevitably introduces approximation errors, but reduces computational complexity at run-time. Nonetheless, MPC in general is computationally demanding when compared to most non-predictive controllers. Also, it is relatively difficult to establish properties such as stability or recursive feasibility for model-predictive controllers. A third way to address the nonlinearity of a prediction model is parameter-varying control [19, 111, 8, 33, 5], in which nonlinearities are perceived as parameters which vary linearly over a convex domain; whatever result is to be obtained must then be enforced for all possible parameter values. While this reduces the computational complexity, it also increases the conservatism with which a result can be stated.

A different approach to the tracking task relies on input-output linearization of an often-used vehicle model, the so-called *bicycle model*. Originating in the work of [57] and further analyzed and also applied in practical experiments by [148], the tracking controller in this approach generates inputs by solution of a nonlinear system of equations, which – just as in the case of MPC – must be solved online. Just as in the case of MPC, this requires more

computational effort than an algebraic feedback controller and has effectively prevented a stability analysis so far, as no closed-form expression of the controller is available. Also, despite first attempts [116], no input or state constraints are considered in this kind of approaches.

More recently, the focus in tracking control for autonomous vehicles has shifted towards safety and the ability of tracking controllers to guarantee properties such as stability, boundedness, or constraint compliance. A classical way of testing a design for such properties relies on experiments and simulation. For example, a simulation-based assessment of the performance in emergency situations of inversion-based controllers such as the one proposed in [57] is given in [68]. There, the set of possible solutions under time-varying disturbances and different initial states is explored, relying on Monte-Carlo simulation and rapidly-exploring random trees. While these methods aim to assess both the average and the worst-case performance, the results are only of stochastic nature. In general, even though a valuable tool for falsification of a design, such methods are not able to guarantee safety, as typically, only a finite set of test cases can be considered. This is problematic in the face of an infinite number of possible reference trajectories in the tracking task, which are unknown prior to run-time. Both makes it impossible to check all relevant cases during system design.

The need for statements pertaining to an infinite number of trajectories has motivated the use of set-based methods from the early days of control theory on, when A. M. Lyapunov proposed to characterize invariant sets by sub-level sets of a certain class of functions [99]. In addition, numerical tools for the explicit computation of (robust) (control) invariant sets, reachable, or controllable sets have been devised more recently:

(Robust) control invariant and *(robust) controllable sets* often find application in the context of MPC [79, 123, 24], where they can be employed to ensure stability or recursive feasibility even in the presence of bounded disturbances. For an example in the tracking context cf. [36], where a model-predictive controller is employed to control a linear system along a reference trajectory which is the output of a reference generator driven by unknown, but bounded inputs. The controller must guarantee compliance with constraints on inputs, states, and the magnitude of the tracking error, which is enabled by the computation of a robust control invariant set for an augmented system comprising both the plant and the reference generator. Also, persistent feasibility can be guaranteed in this way.

Reachable sets and their numerical computation have gained considerable attention in the context of system verification [34, 108, 134, 7, 56]. These sets allow, for example, to check whether a controlled system can reach a forbidden region in its state space, in which case the design would be deemed unsafe. Examples in the vehicle tracking context are given in [69, 6, 129, 100], where the reachable sets of a vehicle while tracking different reference trajectories are computed.

Despite the conceptual promises of set-based methods for system verification, the corresponding numerical computations are generally too demanding to be carried out in real-time during operation of a system. This problem is often met by pre-computing and storing sets offline, making it possible to quickly carry out computations during online operation by resorting to the stored sets. For example, the approach in [129] relies on reachable sets

for the closed-loop system of a vehicle and a tracking controller; these sets are computed offline for different reference values, while in online operation, a planned trajectory is then partitioned and matched to the reachable sets which have been computed for similar reference values before. In doing so, the reachable set of the vehicle state relative to the reference can be assembled online. However, the approach is only able to verify the ability to track a reference trajectory *after* it has been planned, without specifying constructive criteria which could be used by a planning algorithm to produce admissible trajectories.

The approach in [137] combines trajectory planning and tracking for nonlinear systems based on so-called *LQR trees*. These consist of pre-computed reference trajectories, to which a linear, time-varying quadratic regulator is assigned. The reference trajectories end in the region of attraction of a feedback controller that stabilizes a target state. Estimates of the domain of attraction of the time-varying controller of a trajectory characterize states which can be guaranteed to reach the target state. Branches of a tree can be combined to transfer a state over longer distances, provided that the end of one branch is within the region of attraction of the controller associated with the succeeding branch. Computation of approximations of the regions of attraction is facilitated based on sums-of-squares programming, which limits the approach to rather low dimensions. Also, it is difficult to account for moving obstacles and state constraints, as the control inputs depend completely on the pre-computed tracking controllers of the reference trajectories, which cannot be modified online.

The work of [100], while mainly focusing on the planning rather than the tracking task, extends on [137]: now, both parametric and external uncertainty are accounted for and it is possible to account for moving obstacles by shifting reference trajectories in state space. This, however, requires the online solution of a convex, yet comparably expensive quadratically-constrained quadratic program. As a conceptual difference to [137], approximations of reachable sets – similar to [69, 129] – are computed instead of the domain of attraction of controllers, guaranteeing collision-free tracking if the reachable sets do not intersect with obstacles.

The framework in [131] also relies on a separation into an online/offline part in order to ensure robust motion planning for robots with nonlinear dynamics under bounded disturbances and constraints. In the offline part, which is most relevant for the tracking task, a controller is devised which can guarantee robust tracking despite the impact of disturbances. Unlike the approaches [69, 100], it does not rely on computation of reachable sets for finitely many reference trajectories separately; rather, the use of contraction theory allows to establish an invariant tube of fixed size around any feasible trajectory. While this introduces conservatism, it decreases the dependency on the expressiveness of the chosen pre-computed reference trajectories as in [69, 100], such that this approach is claimed to be well-suited for unstructured environments.

In [94], the *viability kernel*, a concept pertaining to robust control invariant sets, is computed in a gridding-based approach in order to enable collision-free driving of autonomous race cars. While a gridding-based approach has the potential to reduce conservatism as its results are not confined to a certain set representation (such as polytopes, ellipsoids, or zonotopes), it is computationally expensive and therefore limited to low-dimensional state

spaces.

The modular approach in [66, 65] can ensure safe tracking for many different planning algorithms, which is enabled by modeling the interaction between the planner and the tracking algorithm as a pursuit-evasion differential game. The value function of the game is defined such that its sub-level sets allow a mapping between an initial tracking error and the maximum error that can occur during tracking under worst-case assumptions for disturbances and planner actions. This information enables the construction of a hybrid controller which makes the sub-level sets of the value function invariant: as long as the tracking error is within predefined bounds, a controller designed by standard methods is active. Once the error approaches the boundary of the admissible region, a safety controller is enabled, which computes control inputs based on gradient information of the value function. The methodology relies on the numerical computation of the value function based on level-set methods [108], which is – despite endeavors to mitigate this drawback by problem decomposition – notoriously expensive.

As outlined in Chapter 1, it should be well noted that all of these methods – just like the method to be presented subsequently – can only provide guarantees with respect to a system model, but not to the actual system itself, as long as the relation between these has not been determined. The field of *robust control* seeks to establish conditions under which system properties still hold under perturbations of the system's parameters (cf. [2] for a general introduction and also a discussion in the vehicle control context); however, assumptions on the perturbations are still required. Despite this conceptual limitation, it is important to characterize the conditions under which a control algorithm is able to work without causing problems.

5.2. Preliminaries

Prior to detailing the tracking algorithm used in this part, several preliminaries regarding coordinate systems, reference trajectories, vehicle and tire models, and the tracking error dynamics are given in this section.

Coordinate Systems

The following derivation relies on the use of three different planar coordinate systems, which are depicted in Fig. 5.1: a cartesian, earth-fixed reference frame with unit basis vectors $\mathbf{e}_x$ and $\mathbf{e}_y$, a curvilinear, so-called Frenet frame with basis $\mathbf{e}_\mathrm{t}$ and $\mathbf{e}_\mathrm{n}$, and a moving, body-fixed coordinate system with basis $\mathbf{e}_X$ and $\mathbf{e}_Y$. In the cartesian frame, the position of a point $r = \begin{bmatrix} p_x & p_y \end{bmatrix} \begin{bmatrix} \mathbf{e}_x & \mathbf{e}_y \end{bmatrix}^\mathsf{T}$ is given relative to an earth-fixed origin by its projections p_x and p_y on the two orthogonal basis vectors. While a cartesian coordinate system is beneficial for planning purposes, the tracking problem can be better approached in a curvilinear coordinate system, the so-called Frenet frame. Instead of using a fixed origin as reference, a Frenet frame locates positions relative to their projection on a reference path, in which a point on the path is given by the path coordinate s and a normal offset to it by the coordinate n in direction of $\mathbf{e}_\mathrm{n}$. The reference path is parameterized in terms

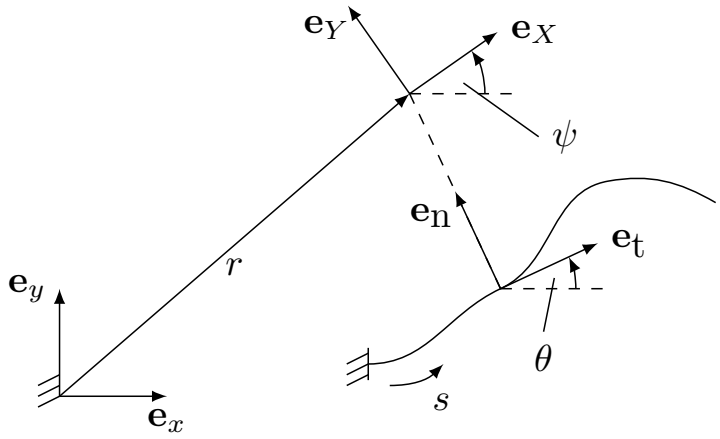


Figure 5.1.: Relation between the cartesian, the body-fixed, and the Frenet coordinate system.

of its orientation $\theta(s)$ with respect to an earth-fixed reference line, which depends on the path coordinate s. The body-fixed coordinate system to be considered is also a cartesian frame, but without an earth-fixed origin. Rather, its origin is attached to the center of gravity of a vehicle and follows its translational and rotational motion. The basis vector $\mathbf{e}_X$ is aligned with the vehicle longitudinal direction, the vector $\mathbf{e}_Y$ with the lateral one. This body-fixed coordinate system conveniently allows to express the vehicle kinetics.

Reference Trajectory

Assume in the following that the planning procedure from Part I provides a reference trajectory $\bar{x}_\mathrm{cart}(\cdot)$ consisting of discrete-time samples from a trajectory in line with Definition 2.1, i.e., of finite duration, given as a sequence of positions $\bar{p}_x$ and $\bar{p}_y$ along with corresponding temporal derivatives (of order three, i.e. $n = 2$ is assumed in the following) in the cartesian reference frame, where the vector $\bar{x}_\mathrm{cart}$ is defined as:

$$\bar{x}_\mathrm{cart} = \begin{bmatrix} \bar{p}_x & \dot{\bar{p}}_x & \ddot{\bar{p}}_x & \dddot{\bar{p}}_x & \bar{p}_y & \dot{\bar{p}}_y & \ddot{\bar{p}}_y & \dddot{\bar{p}}_y \end{bmatrix}^\mathsf{T}. \tag{5.1}$$

Assume that the trajectories of all elements of $\bar{x}_\mathrm{cart}$ except for the third-order derivatives are sampled from continuous solutions to a differential equation, while $\dddot{\bar{p}}_x$ and $\dddot{\bar{p}}_y$ were taken from a piecewise constant, but not necessarily continuous signal. Given a reference trajectory $\bar{x}_\mathrm{cart}(\cdot)$ in the cartesian frame with these continuity properties, it is possible to compute a corresponding reference trajectory $\bar{x}(\cdot)$ in the Frenet frame, where the vector $\bar{x}$ is defined as:

$$\bar{x} = \begin{bmatrix} s & \dot{s} & \ddot{s} & \dddot{s} & \theta & \dot{\theta} & \ddot{\theta} \end{bmatrix}^\mathsf{T}, \tag{5.2}$$

where $\dot{s}$, $\ddot{s}$, $\dddot{s}$, $\dot{\theta}$ and $\ddot{\theta}$ denote the temporal derivatives of s and θ. The transformation can be carried out according to Appendix B.

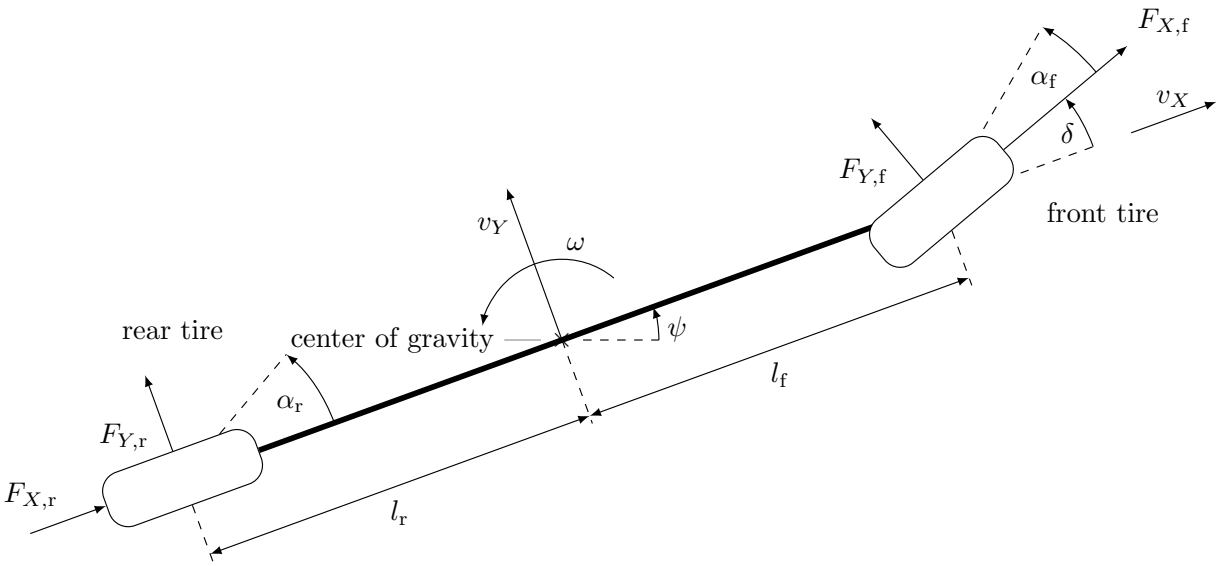


Figure 5.2.: Standard model for vehicle control: the so-called bicycle model.

Vehicle Model

The basis for the design of the tracking controller is the so-called bicycle model as illustrated in Figure 5.2; the quantities shown there will be introduced throughout this section. This model is widely used for vehicle control purposes. Its state vector comprises the position $\begin{bmatrix} p_x & p_y \end{bmatrix}^\mathsf{T}$ of the center of gravity and the vehicle orientation ψ in the cartesian reference frame, the longitudinal and lateral vehicle speed v_X and v_Y in body-fixed coordinates, and the yaw rate ω. The state vector reads:

$$\chi := \begin{bmatrix} p_x & p_y & \psi & v^\mathsf{T} \end{bmatrix}^\mathsf{T}, \quad v := \begin{bmatrix} v_X & v_Y & \omega \end{bmatrix}^\mathsf{T}, \tag{5.3}$$

while the input vector contains as actuator inputs the longitudinal tire slip s_X and the front wheel steering angle δ:

$$\mu := \begin{bmatrix} s_X \\ \delta \end{bmatrix}.$$

Roll, pitch, and heave dynamics are neglected just as elastic deformations of the vehicle body. Therefore, a separate consideration of left and right wheels is not necessary, such that these are lumped together, resulting in the bicycle-like shape.

Next, the orthogonal rotation matrix $R : \mathbb{R} \to \mathbb{R}^{2\times 2}$ is introduced, transforming vectors from the body-fixed vehicle frame into the global cartesian coordinate frame:

$$R(\bullet) = \begin{bmatrix} \cos(\bullet) & -\sin(\bullet) \\ \sin(\bullet) & \cos(\bullet) \end{bmatrix}.$$

Along with the vector of external accelerations:

$$a := \begin{bmatrix} a_X \\ a_Y \\ a_\psi \end{bmatrix},$$

the equations of motion of the vehicle's center of gravity are given by:

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\psi} \\ \dot{v}_X \\ \dot{v}_Y \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} R(\psi) \begin{bmatrix} v_X \\ v_Y \end{bmatrix} \\ \omega \\ \begin{bmatrix} a_X \\ a_Y \\ a_\psi \end{bmatrix} + \begin{bmatrix} v_Y \omega \\ -v_X \omega \\ 0 \end{bmatrix} \end{bmatrix}. \tag{5.4}$$

The accelerations a_X, a_Y, a_ψ result from external forces such as gravity, wind, or tire forces. All these components are difficult to model and therefore a major source of uncertainty in (5.4). This is explicitly accounted for by defining:

$$\Delta a = \begin{bmatrix} \Delta a_X \\ \Delta a_Y \\ \Delta a_\psi \end{bmatrix} := a - \bar{a}, \quad \bar{a} := \begin{bmatrix} \bar{a}_X \\ \bar{a}_Y \\ \bar{a}_\psi \end{bmatrix} \tag{5.5}$$

as mismatch between the actual external accelerations a and the nominal accelerations $\bar{a}$ based on a given model. It is assumed that the magnitude of the mismatch is bounded by a scalar $\Delta a_{\max}$, which allows to define a set:

$$\mathcal{A}_\Delta := \{\Delta a |\ ||\Delta a|| \leq \Delta a_{\max}\}. \tag{5.6}$$

In the following, the discussion of external accelerations will focus on tire forces, which are the main cause for external accelerations that can be influenced. Additional forces and corresponding uncertainties could be considered in the same framework, but are beyond the scope of the exposition. It is assumed that $\bar{a}$ models the dependency on the following variables:

$$\bar{a} = \bar{a}(\delta, s_X, v_X, v_Y, \omega), \tag{5.7}$$

while other influences are not modeled explicitly, but attributed to the uncertainty Δa.

Tire model

External accelerations due to longitudinal and lateral tire force vectors (in the body-fixed frame) $\begin{bmatrix} F_{X,\mathrm{f}} & F_{Y,\mathrm{f}} \end{bmatrix}^\mathsf{T}$ and $\begin{bmatrix} F_{X,\mathrm{r}} & F_{Y,\mathrm{r}} \end{bmatrix}^\mathsf{T}$ at front and rear axle, respectively, result from:

$$\begin{aligned} \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & \Theta \end{bmatrix} a_{\text{tire}} &= \begin{bmatrix} \begin{bmatrix} F_{X,\mathrm{r}} \\ F_{Y,\mathrm{r}} \end{bmatrix} + \begin{bmatrix} \cos\delta & -\sin\delta \\ \sin\delta & \cos\delta \end{bmatrix} \begin{bmatrix} F_{X,\mathrm{f}} \\ F_{Y,\mathrm{f}} \end{bmatrix} \\ \left(F_{Y,\mathrm{f}} \cos\delta + F_{X,\mathrm{f}} \sin\delta\right) l_\mathrm{f} - F_{Y,\mathrm{r}} l_\mathrm{r} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -l_\mathrm{r} \end{bmatrix} \begin{bmatrix} F_{X,\mathrm{r}} \\ F_{Y,\mathrm{r}} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & l_\mathrm{f} \end{bmatrix} R(\delta) \begin{bmatrix} F_{X,\mathrm{f}} \\ F_{Y,\mathrm{f}} \end{bmatrix}, \end{aligned} \tag{5.8}$$

with vehicle mass m and yaw inertia Θ as well as the distances l_r and l_f between center of gravity and rear respectively front axle, cf. Figure 5.2. The tire forces are highly affected by the front and rear tire slip vectors s_f and s_r, respectively (note that different definitions exist in the literature):

$$s_\mathrm{f} = \begin{bmatrix} s_{X,\mathrm{f}} \\ s_{Y,\mathrm{f}} \end{bmatrix} := \frac{\left(\begin{bmatrix} v_{\mathrm{wheel,f}} \\ 0 \end{bmatrix} - R(\delta)^\mathsf{T} \begin{bmatrix} v_X \\ v_Y + l_\mathrm{f}\omega \end{bmatrix} \right)}{\left\| \begin{bmatrix} v_X \\ v_Y + l_\mathrm{f}\omega \end{bmatrix} \right\|},$$

$$s_\mathrm{r} = \begin{bmatrix} s_{X,\mathrm{r}} \\ s_{Y,\mathrm{r}} \end{bmatrix} := \frac{\left(\begin{bmatrix} v_{\mathrm{wheel,r}} \\ 0 \end{bmatrix} - \begin{bmatrix} v_X \\ v_Y - l_\mathrm{r}\omega \end{bmatrix} \right)}{\left\| \begin{bmatrix} v_X \\ v_Y - l_\mathrm{r}\omega \end{bmatrix} \right\|},$$

which quantify the deviation of the vehicle wheels' rim speed, $v_{\mathrm{wheel,f}}$ and $v_{\mathrm{wheel,r}}$, and the corresponding motions of the vehicle body. Front and rear longitudinal slip are assumed to be related through the control input s_X:

$$s_{X,\mathrm{f}} = \gamma s_X, \quad s_{X,\mathrm{r}} = (1-\gamma) s_X,$$

where the vehicle parameter $\gamma \in [0,1]$ determines the distribution of traction force between front and rear axle in an all-wheel drive train. Between the lateral tire slips, no such relation exists. For simplicity, they are substituted by the tire side slip angle:

$$\alpha_i := \arcsin s_{Y,i}, \ i \in \{\mathrm{f}, \mathrm{r}\}. \tag{5.9}$$

Apart from tire slip, many other quantities impact the tire forces, such as camber angle, normal load, or the friction of the road surface. Accordingly, a variety of tire models of different complexity exists [114, 121]. In the following, a model is considered which linearly relates tire forces and tire slip through longitudinal and lateral tire stiffness c_X, c_Y:

$$\begin{aligned} F_{X,\mathrm{f}} &= c_X s_{X,\mathrm{f}} = c_X \gamma s_X, & F_{X,\mathrm{r}} &= c_X s_{X,\mathrm{r}} = c_X (1-\gamma) s_X, \\ F_{Y,\mathrm{f}} &= c_Y \alpha_\mathrm{f}, & F_{Y,\mathrm{r}} &= c_Y \alpha_\mathrm{r}. \end{aligned} \tag{5.10}$$

It is assumed that the tire stiffness parameters c_X and c_Y are subject to additive uncertainties Δc_Y and Δc_X around nominal values $\bar{c}_X$ and $\bar{c}_Y$:

$$c_X = \bar{c}_X + \Delta c_X, \quad c_Y = \bar{c}_Y + \Delta c_Y. \tag{5.11}$$

These uncertainty terms are assumed to be bounded:

$$|\Delta c_X| \leq \Delta c_{X,\max}, \quad |\Delta c_Y| \leq \Delta c_{Y,\max}$$

and contribute to the acceleration disturbance Δa. Postponing a detailed discussion of the effects of the tire uncertainty to Section 5.3, the combination of (5.10) and (5.8) gives:

$$a_\mathrm{tire} = \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & -\frac{l_\mathrm{r}}{\Theta} \end{bmatrix} \begin{bmatrix} (1-\gamma) c_X s_X \\ c_Y \arctan\left(\frac{l_\mathrm{r}\omega - v_Y}{v_X}\right) \end{bmatrix} + \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & \frac{l_\mathrm{f}}{\Theta} \end{bmatrix} R(\delta) \begin{bmatrix} \gamma c_X s_X \\ c_Y \left(\delta - \arctan\left(\frac{l_\mathrm{f}\omega + v_Y}{v_X}\right)\right) \end{bmatrix}. \tag{5.12}$$

Table 5.1.: Vehicle parameters (from [57])

Parameter	m	Θ	γ	l_f	l_r	$\bar{c}_X$	$\bar{c}_Y$
Value	1529 kg	1344 kg m^2	0.6	1.481 m	1.08 m	100 000 $\frac{\mathrm{N}}{\mathrm{rad}}$	100 000 $\frac{\mathrm{N}}{\mathrm{rad}}$

Error Dynamics

The mismatch between a reference trajectory and the vehicle states is described by the error state vector:

$$e := \begin{bmatrix} e_\mathrm{t} & e_\mathrm{n} & \dot{e}_\mathrm{t} & \dot{e}_\mathrm{n} & e_\psi & e_\omega \end{bmatrix}^\mathsf{T} = \begin{bmatrix} e_\mathrm{pos}^\mathsf{T} & \dot{e}_\mathrm{pos}^\mathsf{T} & e_\mathrm{yaw}^\mathsf{T} \end{bmatrix}^\mathsf{T},$$

where

$$e_\mathrm{pos} := \begin{bmatrix} e_\mathrm{t} \\ e_\mathrm{n} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \left(\begin{bmatrix} p_x \\ p_y \end{bmatrix} - \begin{bmatrix} \bar{p}_x \\ \bar{p}_y \end{bmatrix} \right) \tag{5.13}$$

gives the offset in tangential and lateral direction relative to the reference trajectory, with temporal derivative:

$$\dot{e}_\mathrm{pos} := \frac{\mathrm{d}}{\mathrm{d}t} e_\mathrm{pos} = \begin{bmatrix} \dot{e}_\mathrm{t} \\ \dot{e}_\mathrm{n} \end{bmatrix} = \dot{\theta} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} e_\mathrm{pos} - \begin{bmatrix} \dot{s} \\ 0 \end{bmatrix} + R(\psi - \theta) \begin{bmatrix} v_X \\ v_Y \end{bmatrix}, \tag{5.14}$$

and the orientation error and its derivative:

$$e_\mathrm{yaw} := \begin{bmatrix} e_\psi \\ e_\omega \end{bmatrix}, \; e_\psi := \psi - \theta, \; e_\omega := \frac{\mathrm{d}}{\mathrm{d}t} e_\psi. \tag{5.15}$$

The relation to the external accelerations a is given by:

$$\frac{\mathrm{d}}{\mathrm{d}t} \dot{e}_\mathrm{pos} = \begin{bmatrix} \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} e_\mathrm{pos} \\ \dot{e}_\mathrm{pos} \end{bmatrix} - \begin{bmatrix} \ddot{s} \\ \dot{\theta}\dot{s} \end{bmatrix} + R(e_\psi) \begin{bmatrix} a_X \\ a_Y \end{bmatrix}, \tag{5.16}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} e_\omega = a_\psi - \ddot{\theta}. \tag{5.17}$$

Note that substantial simplifications of the dynamics can be obtained if the longitudinal velocity is assumed to be constant [121, 2]. For planning purposes, however, this assumption is too restrictive.

5.3. Partial Compensation of Nonlinearities by Feedback

Combining (5.14) with (5.16) and replacing the actual acceleration a by $\bar{a}$ and Δa according to (5.5) gives:

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} e_\mathrm{pos} \\ \dot{e}_\mathrm{pos} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} e_\mathrm{pos} \\ \dot{e}_\mathrm{pos} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \ddot{s} \\ \dot{\theta}\dot{s} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ R(e_\psi) \end{bmatrix} \begin{bmatrix} \bar{a}_X \\ \bar{a}_Y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ R(e_\psi) \end{bmatrix} \begin{bmatrix} \Delta a_X \\ \Delta a_Y \end{bmatrix} \tag{5.18}$$

and similarly, based on (5.15) and (5.17),

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} e_\psi \\ \dot{e}_\omega \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} e_\psi \\ \dot{e}_\omega \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}(\bar{a}_\psi + \Delta a_\psi - \ddot{\theta}). \tag{5.19}$$

According to (5.7), the nominal acceleration $\bar{a}$ can be controlled for given vehicle velocities v_X, v_Y, and ω by choice of δ and s_X, while Δa acts as an acceleration disturbance and is beyond control. Tracking requires to influence the tracking error dynamics by feedback, such that it is desirable to determine s_X and δ and, thereby, $\bar{a}$, in dependency of e. A common way [148, 57] to achieve this is to introduce

$$\tilde{\mu} := \begin{bmatrix} \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix}\begin{bmatrix} e_{\mathrm{pos}} \\ \dot{e}_{\mathrm{pos}} \end{bmatrix} - \begin{bmatrix} \ddot{s} \\ \dot{\theta}\dot{s} \end{bmatrix} + R(e_\psi)\begin{bmatrix} \bar{a}_X \\ \bar{a}_Y \end{bmatrix} \tag{5.20}$$

$$\Leftrightarrow \begin{bmatrix} \bar{a}_X \\ \bar{a}_Y \end{bmatrix} = R(e_\psi)^{\mathsf{T}}\left(\tilde{\mu} + \begin{bmatrix} \ddot{s} \\ \dot{\theta}\dot{s} \end{bmatrix} - \begin{bmatrix} \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix}\begin{bmatrix} e_{\mathrm{pos}} \\ \dot{e}_{\mathrm{pos}} \end{bmatrix}\right) \tag{5.21}$$

and to substitute (5.21) into (5.18), leading to:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} e_{\mathrm{pos}} \\ \dot{e}_{\mathrm{pos}} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}\begin{bmatrix} e_{\mathrm{pos}} \\ \dot{e}_{\mathrm{pos}} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix}\tilde{\mu} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ R(e_\psi) \end{bmatrix}\begin{bmatrix} \Delta a_X \\ \Delta a_Y \end{bmatrix}. \tag{5.22}$$

Considering $\tilde{\mu}$ as an artificial input quantity and letting

$$\tilde{\mu} = -K\begin{bmatrix} e_{\mathrm{pos}} \\ \dot{e}_{\mathrm{pos}} \end{bmatrix}, \tag{5.23}$$

with a feedback matrix $K \in \mathbb{R}^{2\times 4}$ to be designed subsequently, the resulting closed-loop position and velocity error dynamics Σ_{pos} becomes:

$$\Sigma_{\mathrm{pos}}: \qquad \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} e_{\mathrm{pos}} \\ \dot{e}_{\mathrm{pos}} \end{bmatrix} = f_{\mathrm{pos}}(e, \Delta a) := \begin{bmatrix} 0_{2\times 2} & I_{2\times 2} \\ -K & \end{bmatrix}\begin{bmatrix} e_{\mathrm{pos}} \\ \dot{e}_{\mathrm{pos}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ R(e_\psi) \end{bmatrix}\begin{bmatrix} \Delta a_X \\ \Delta a_Y \end{bmatrix}. \tag{5.24}$$

The translation of a prescribed value for $\bar{a}$ as given in (5.21) into control inputs δ and s_X as well as the yaw error dynamics resulting from it are detailed in the following.

Input Computation

Introducing (5.11) into (5.12) and replacing nonlinear terms by first-order Taylor series expansion gives:

$$a_{\mathrm{tire}} = \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & -\frac{l_{\mathrm{r}}}{\Theta} \end{bmatrix}\begin{bmatrix} (1-\gamma)\bar{c}_X s_X \\ \bar{c}_Y \frac{l_{\mathrm{r}}\omega - v_Y}{v_X} \end{bmatrix} + \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & \frac{l_{\mathrm{f}}}{\Theta} \end{bmatrix}\begin{bmatrix} 1 & -\delta \\ \delta & 1 \end{bmatrix}\begin{bmatrix} \gamma\bar{c}_X s_X \\ \bar{c}_Y\left(\delta - \frac{l_{\mathrm{f}}\omega + v_Y}{v_X}\right) \end{bmatrix} + \Delta a_{\mathrm{R}} + \Delta a_c, \tag{5.25}$$

where

$$\Delta a_c := \begin{bmatrix} \Delta a_{X,c} & \Delta a_{Y,c} & \Delta a_{\psi,c} \end{bmatrix}^\mathsf{T}$$

accounts for the tire uncertainty according to (5.11) and

$$\Delta a_\mathrm{R} = \begin{bmatrix} \Delta a_{X,\mathrm{R}} & \Delta a_{Y,\mathrm{R}} & \Delta a_{\psi,\mathrm{R}} \end{bmatrix}^\mathsf{T}$$

contains the series expansion remainder. A detailed discussion of these two uncertainty terms is given in Appendix C along with a derivation of upper bounds such that $||\Delta a|| \leq \Delta a_\mathrm{max}$ as in (5.6). The subsequent analysis is based on:

$$\bar{a} := \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & -\frac{l_\mathrm{r}}{\Theta} \end{bmatrix} \begin{bmatrix} (1-\gamma)\bar{c}_X s_X \\ \bar{c}_Y \frac{l_\mathrm{r}\omega - v_Y}{v_X} \end{bmatrix} + \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & \frac{l_\mathrm{f}}{\Theta} \end{bmatrix} \begin{bmatrix} 1 & -\delta \\ \delta & 1 \end{bmatrix} \begin{bmatrix} \gamma \bar{c}_X s_X \\ \bar{c}_Y \left(\delta - \frac{l_\mathrm{f}\omega + v_Y}{v_X}\right) \end{bmatrix} \tag{5.26}$$

and:

$$\Delta a := \Delta a_\mathrm{R} + \Delta a_c. \tag{5.27}$$

From the first two lines of (5.26), it follows that the inputs s_X and δ can be obtained from:

$$0 = \delta^3 - \delta^2 \frac{l_\mathrm{f}\omega + v_Y}{v_X} + \delta \left(\frac{m}{\bar{c}_Y} \bar{a}_X + \frac{1}{\gamma} \right) + \frac{\omega(l_\mathrm{r} - l_\mathrm{f}) - 2v_Y}{\gamma v_X} - \frac{m}{\gamma \bar{c}_Y} \bar{a}_Y, \tag{5.28}$$

$$s_X = \frac{\bar{c}_Y}{\bar{c}_X} \left(\delta^2 - \frac{l_\mathrm{f}\omega + v_Y}{v_X} \delta + \frac{m}{\bar{c}_Y} \bar{a}_X \right). \tag{5.29}$$

In general, the solution of (5.28) is not unique. However, constraining both the parameters of the polynomial (5.28) as well as the variable δ as derived in Chapter 6 allows to bypass this problem (cf. also Appendix F). Also, the solution(s) can be obtained using an explicit formula, such that no iterative numerical computations are required. The tracking control algorithm as given in Algorithm 5.1 summarizes the results from this section.

Resulting Yaw Error Dynamics

The control law in Algorithm 5.1 allows to derive a closed-form expression for the yaw error dynamics (5.19), which has not been possible in earlier work [148, 57]. Starting point is the third line of (5.26), from which it follows that

$$\Theta \bar{a}_\psi = \left(l_\mathrm{f} \begin{bmatrix} \delta & 1 \end{bmatrix} \begin{bmatrix} \gamma s_X & 0 \\ 0 & \delta - \frac{l_\mathrm{f}\omega + v_Y}{v_X} \end{bmatrix} + \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & -l_\mathrm{r} \frac{l_\mathrm{r}\omega - v_Y}{v_X} \end{bmatrix} \right) \begin{bmatrix} \bar{c}_X \\ \bar{c}_Y \end{bmatrix}$$

and, according to (5.29),

$$\delta \cdot s_X = \frac{\bar{c}_Y}{\bar{c}_X} \left(\delta^3 - \delta^2 \frac{l_\mathrm{f}\omega + v_Y}{v_X} + \delta \frac{m}{\bar{c}_Y} \bar{a}_X \right) = \frac{\bar{c}_Y}{\bar{c}_X} \left(\frac{m}{\bar{c}_Y \gamma} \bar{a}_Y + \frac{l_\mathrm{f}\omega + v_Y}{\gamma v_X} - \frac{l_\mathrm{r}\omega - v_Y}{\gamma v_X} - \frac{\delta}{\gamma} \right).$$

Algorithm 5.1.: Proposed Tracking Control Algorithm

procedure TRACKINGCONTROLLER(χ,$\bar{x}_{\text{cart}}(\cdot)$)
  $\bar{x}(\cdot) \leftarrow \text{Cart2Curved}(\bar{x}_{\text{cart}}(\cdot))$ ▷ Transformation according to Appendix B
  $e_{\text{pos}} \leftarrow \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \left(\begin{bmatrix} p_x \\ p_y \end{bmatrix} - \begin{bmatrix} \bar{p}_x \\ \bar{p}_y \end{bmatrix} \right)$
  $\dot{e}_{\text{pos}} \leftarrow \dot{\theta} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} e_{\text{pos}} - \begin{bmatrix} \dot{s} \\ 0 \end{bmatrix} + R(\psi - \theta) \begin{bmatrix} v_X \\ v_Y \end{bmatrix}$
  $e_\psi \leftarrow \psi - \theta$
  $\begin{bmatrix} \bar{a}_X \\ \bar{a}_Y \end{bmatrix} \leftarrow R(e_\psi)^\mathsf{T} \left(-K \begin{bmatrix} e_{\text{pos}} \\ \dot{e}_{\text{pos}} \end{bmatrix} + \begin{bmatrix} \ddot{s} \\ \dot{\theta}\dot{s} \end{bmatrix} - \begin{bmatrix} \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} e_{\text{pos}} \\ \dot{e}_{\text{pos}} \end{bmatrix} \right)$
  $\delta \leftarrow \text{Solve } 0 = \delta^3 - \delta^2 \frac{l_\text{f}\omega + v_Y}{v_X} + \delta \left(\frac{m}{\bar{c}_Y} \bar{a}_X + \frac{1}{\gamma} \right) + \frac{\omega(l_\text{r} - l_\text{f}) - 2v_Y}{\gamma v_X} - \frac{m}{\gamma \bar{c}_Y} \bar{a}_Y$
  $s_X \leftarrow \frac{\bar{c}_Y}{\bar{c}_X} \left(\delta^2 - \frac{l_\text{f}\omega + v_Y}{v_X} \delta + \frac{m}{\bar{c}_Y} \bar{a}_X \right)$
  return $\mu = \begin{bmatrix} s_X \\ \delta \end{bmatrix}$
end procedure

Combining these and introducing the result into (5.19) gives:

$$\begin{aligned}
\dot{e}_\omega &= \frac{1}{\Theta} \left(\begin{bmatrix} l_\text{f} \gamma \frac{\bar{c}_Y}{\bar{c}_X} \left(\frac{m}{\bar{c}_Y \gamma} \bar{a}_Y + \frac{l_\text{f}\omega + v_Y}{\gamma v_X} - \frac{l_\text{r}\omega - v_Y}{\gamma v_X} - \frac{\delta}{\gamma} \right) & l_\text{f} \left(\delta - \frac{l_\text{f}\omega + v_Y}{v_X} \right) - l_\text{r} \frac{l_\text{r}\omega - v_Y}{v_X} \end{bmatrix} \right) \begin{bmatrix} \bar{c}_X \\ \bar{c}_Y \end{bmatrix} - \ddot{\theta} \ldots \\
&\quad + \Delta a_\psi \\
&= \frac{\bar{c}_X}{\Theta} \left(\frac{m}{\bar{c}_X} l_\text{f} \bar{a}_Y - \frac{\bar{c}_Y}{\bar{c}_X} \delta l_\text{f} + \frac{\bar{c}_Y}{\bar{c}_X} l_\text{f} \left(\frac{l_\text{f}\omega + v_Y}{v_X} - \frac{l_\text{r}\omega - v_Y}{v_X} \right) \right) \ldots \\
&\quad + \frac{\bar{c}_Y}{\Theta} \left(l_\text{f} \delta - l_\text{f} \frac{l_\text{f}\omega + v_Y}{v_X} - l_\text{r} \frac{l_\text{r}\omega - v_Y}{v_X} \right) - \ddot{\theta} + \Delta a_\psi \\
&= \frac{m}{\Theta} l_\text{f} \bar{a}_Y - \frac{\bar{c}_Y (l_\text{f} + l_\text{r})}{\Theta} \frac{l_\text{r}\omega - v_Y}{v_X} - \ddot{\theta} + \Delta a_\psi,
\end{aligned} \tag{5.30}$$

with Δa_ψ from (5.27). Because the effect of δ cancels out in (5.30), it is not necessary to substitute the explicit solution of (5.28). Rewriting (5.14) as:

$$\begin{bmatrix} v_X \\ v_Y \end{bmatrix} = R(e_\psi)^\mathsf{T} \left(\begin{bmatrix} \dot{s} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_{\text{pos}} \\ \dot{e}_{\text{pos}} \end{bmatrix} \right) \tag{5.31}$$

and recalling the definition of $\bar{a}_Y$ according to (5.21) as well as that $\omega = \dot{\theta} + e_\omega$ by virtue of (5.15), (5.30) can be rewritten such that the yaw error dynamics is given by $f_{\text{yaw}} : \mathbb{R}^6 \times \mathbb{R}^{2(n+1)} \times \mathbb{R}^3 \to \mathbb{R}^2$:

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} e_\psi \\ e_\omega \end{bmatrix} = f_{\text{yaw}}(e, \bar{x}, \Delta a),$$

with

$$f_{\text{yaw}}(e, \bar{x}, \Delta a) := \tag{5.32}$$

$$\begin{bmatrix} e_\omega \\ \frac{m}{\Theta} l_\text{f} \begin{bmatrix} 0 & 1 \end{bmatrix} R(e_\psi)^\mathsf{T} \left(-K \begin{bmatrix} e_\text{pos} \\ \dot{e}_\text{pos} \end{bmatrix} + \begin{bmatrix} \ddot{s} \\ \dot{\theta}\dot{s} \end{bmatrix} - \begin{bmatrix} \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} e_\text{pos} \\ \dot{e}_\text{pos} \end{bmatrix} \right) \dots \\ - \frac{\bar{c}_Y l_\text{r} (l_\text{f}+l_\text{r})}{\Theta \left(\begin{bmatrix} 1 & 0 \end{bmatrix} R(e_\psi)^\mathsf{T} \left(\begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_\text{pos} \\ \dot{e}_\text{pos} \end{bmatrix} \right) + \dot{s} \cos e_\psi \right)} \dot{\theta} \dots \\ - \frac{\bar{c}_Y l_\text{r} (l_\text{f}+l_\text{r})}{\Theta \left(\begin{bmatrix} 1 & 0 \end{bmatrix} R(e_\psi)^\mathsf{T} \left(\begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_\text{pos} \\ \dot{e}_\text{pos} \end{bmatrix} \right) + \dot{s} \cos e_\psi \right)} e_\omega \dots \\ + \frac{\bar{c}_Y (l_\text{f}+l_\text{r})}{\Theta} \frac{\begin{bmatrix} 0 & 1 \end{bmatrix} R(e_\psi)^\mathsf{T} \left(\begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_\text{pos} \\ \dot{e}_\text{pos} \end{bmatrix} \right) - \dot{s} \sin e_\psi}{\begin{bmatrix} 1 & 0 \end{bmatrix} R(e_\psi)^\mathsf{T} \left(\begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_\text{pos} \\ \dot{e}_\text{pos} \end{bmatrix} \right) + \dot{s} \cos e_\psi} - \ddot{\theta} + \Delta a_\psi \end{bmatrix} .$$

The closed-loop yaw error dynamics (5.32) is nonlinear and subject to external disturbances. In addition, it varies over time in dependency of the reference trajectory $\bar{x}(\cdot)$.

Having detailed both the control law and the error dynamics resulting from its application to the vehicle model, the following chapter details the constraints which must be accounted for during operation.

6. Characterizing Admissible Vehicle States and Reference Trajectories

Both the vehicle states and the control inputs are subject to constraints, which must be explicitly accounted for during tracking in order to ensure safe operation of a vehicle. The following chapter characterizes those vehicle states which are admissible and, given the control law as in Algorithm 5.1, result in admissible control inputs. Then, temporarily making assumptions on the error dynamics (which will be validated in the next chapter), the resulting implications on admissible parameters of the reference trajectory are analyzed.

6.1. Constraints on the Vehicle Dynamics

Constraints on the vehicle dynamics result from different sources: *physical laws* impose limits on the possible accelerations, maximum velocities, and actuator inputs; these limits are immutable for a given vehicle design and enforce themselves. The passengers' *driving comfort* typically requires more conservative bounds than those imposed by physical limitations, and if no special care is taken in the design of the control algorithm, violations of these constraints can occur. This also holds for velocity constraints resulting from *legal restrictions*. In addition, it may be desirable to impose constraints due to *modeling considerations*, e.g. in order to explicitly account for the region of validity of a model used in the controller design process. In the following, constraints on different quantities resulting from such considerations are collected.

The simplest constraints on the vehicle velocities are imposed as magnitude constraints:

$$v_{X,\min} \leq v_X \leq v_{X,\max},\ |v_Y| \leq v_{Y,\max},\ |\omega| \leq \omega_{\max}, \tag{6.1}$$

where it is assumed that $v_{X,\min} > 0$, i.e., a vehicle may not come to standstill nor back up during regular operation on a highway. Additional velocity constraints result from bounds on the rear side-slip angle α_r, which follows from (5.9) as:

$$\alpha_\mathrm{r} = \arctan\left(\frac{l_\mathrm{r}\omega - v_Y}{v_X}\right).$$

Motivated by the fact that a linear tire model is widely considered a good approximation of the actual vehicle dynamics for *small* tire slip angles, the constraint:

$$|\alpha_\mathrm{r}| \leq \tilde{\alpha}_{\max} \tag{6.2}$$

is imposed with a positive scalar $\tilde{\alpha}_{\max}$. In addition, the so-called body side-slip angle α_{bss}:

$$\alpha_{\text{bss}} := \arctan \frac{v_Y}{v_X}$$

is introduced, which can be related to the handling limits of a vehicle [121]. This motivates the requirement that $|\alpha_{\text{bss}}| \leq \alpha_{\text{bss,max}}$. Then, all constraints on the vehicle's velocities in the tuple v as defined in (5.3) can be combined in the polyhedral, convex set:

$$\begin{aligned}\mathcal{V} := \Big\{ v | & v_{X,\min} \leq v_X \leq v_{X,\max}, |v_Y| \leq v_{Y,\max}, |\omega| \leq \omega_{\max}, \\ & |l_{\text{r}}\omega - v_Y| \leq v_X \tan(\tilde{\alpha}_{\max}),\ |v_Y| \leq v_X \tan(\alpha_{\text{bss,max}}) \Big\}. \end{aligned} \tag{6.3}$$

According to (5.4), interval constraints:

$$|\dot{v}_X| \leq \dot{v}_{X,\max}, \quad |\dot{v}_Y| \leq \dot{v}_{Y,\max}$$

on the resulting accelerations $\dot{v}_X$ and $\dot{v}_Y$ impose constraints on the velocities v and external accelerations a. According to (5.5), the value of a is subject to uncertainty; while the acceleration constraints apply for the sum of $\bar{a}$ and Δa, only $\bar{a}$ can be controlled to actually meet the requirements on a. Therefore, it is desirable to derive constraints on $\bar{a}$ which, if met, also guarantee compliance with those on a despite possible uncertainty Δa. To that end, consider that

$$\begin{aligned} |a_X + v_Y\omega| &= |\bar{a}_X + \Delta a_X + v_Y\omega| \leq |\bar{a}_X + v_Y\omega| + |\Delta a_X|, \\ |a_Y - v_X\omega| &= |\bar{a}_Y + \Delta a_Y - v_X\omega| \leq |\bar{a}_Y - v_X\omega| + |\Delta a_Y|. \end{aligned}$$

Noting that $|\Delta a_i| \leq ||\Delta a||$, $i \in \{X, Y\}$, and defining

$$\begin{aligned} \bar{a}_{X,\max} &:= \dot{v}_{X,\max} - ||\Delta a||, \\ \bar{a}_{Y,\max} &:= \dot{v}_{Y,\max} - ||\Delta a||, \end{aligned}$$

it clearly holds that

$$|\bar{a}_X + v_Y\omega| \leq \bar{a}_{X,\max} \Rightarrow |a_X + v_Y\omega| \leq \dot{v}_{X,\max}, \tag{6.4}$$

$$|\bar{a}_Y - v_X\omega| \leq \bar{a}_{Y,\max} \Rightarrow |a_Y - v_X\omega| \leq \dot{v}_{Y,\max}. \tag{6.5}$$

Unlike the pure velocity constraints $\mathcal{V}$, the acceleration constraints on the left side of the implications (6.4) and (6.5) are nonlinear. Also, they pertain to both accelerations and velocities and thus introduce a coupling between these quantities. In Appendix D, a polytopic inner approximation $\mathcal{A}$ of the admissible set defined by combination of (6.4) and (6.5) is derived, such that

$$\begin{bmatrix} v \\ \bar{a} \end{bmatrix} \in \mathcal{A} \Rightarrow |a_X + v_Y\omega| \leq \dot{v}_{X,\max},\ |a_Y - v_X\omega| \leq \dot{v}_{Y,\max}.$$

An important step towards safe trajectory tracking is to characterize the circumstances under which an admissible control input exists. Just like the velocities and accelerations, the control inputs are subject to magnitude constraints:

$$|s_X| \leq s_{X,\max},\ |\delta| \leq \delta_{\max}. \tag{6.6}$$

Under the control law from Algorithm 5.1, a magnitude constraint on the longitudinal slip s_X according to (5.29) requires that:

$$\left| \delta^2 - \frac{l_\mathrm{f}\omega + v_Y}{v_X}\delta + \frac{m}{\bar{c}_Y}\bar{a}_X \right| \leq \frac{\bar{c}_X}{\bar{c}_Y} s_{X,\max},$$

which is a coupling constraint on δ, v_X, v_Y, ω, and $\bar{a}_X$. Based on (5.9), the front side slip angle α_f depends on the input δ according to:

$$\alpha_\mathrm{f} = \delta - \arctan\left(\frac{l_\mathrm{f}\omega + v_Y}{v_X}\right), \tag{6.7}$$

such that a magnitude constraint $|\alpha_\mathrm{f}| \leq \tilde{\alpha}_{\max}$ introduces another coupling constraint. For practical reasons, subsequent derivations require polynomial expressions, such that trigonometric terms must be approximated. Linearizing (6.7) and obtaining an upper bound on the linearization error as detailed in Appendix E eventually leads to:

$$\frac{l_\mathrm{f}\omega + v_Y}{v_X} - \alpha_{\max} \leq \delta \leq \frac{l_\mathrm{f}\omega + v_Y}{v_X} + \alpha_{\max}, \tag{6.8}$$

with modified upper bound $\alpha_{\max} \leq \tilde{\alpha}_{\max}$ such that (6.8) $\Rightarrow |\alpha_\mathrm{f}| \leq \tilde{\alpha}_{\max}$, cf. Appendix E. A magnitude constraint on δ is equivalent to the requirement that a solution of the cubic equation (5.28) must lie in a certain interval on the real axis. Collecting all constraints pertaining to the control inputs gives:

$$\begin{aligned}\mathcal{U} := \Bigg\{ \begin{bmatrix} v \\ \bar{a} \\ \mu \end{bmatrix} \ \Bigg|\ & \left|\delta - \frac{l_\mathrm{f}\omega + v_Y}{v_X}\right| \leq \alpha_{\max},\ \left|\delta^2 - \frac{l_\mathrm{f}\omega + v_Y}{v_X}\delta + \frac{m}{\bar{c}_Y}\bar{a}_X\right| \leq \frac{\bar{c}_X}{\bar{c}_Y} s_{X,\max}, \\ & |\delta| \leq \delta_{\max},\ \delta^3 - \delta^2\frac{l_\mathrm{f}\omega + v_Y}{v_X} + \delta\left(\frac{m}{\bar{c}_Y}\bar{a}_X + \frac{1}{\gamma}\right) + \frac{\omega(l_\mathrm{r} - l_\mathrm{f}) - 2v_Y}{\gamma v_X} - \frac{m}{\gamma\bar{c}_Y}\bar{a}_Y = 0 \Bigg\}.\end{aligned} \tag{6.9}$$

In order to account for all constraints on velocities, accelerations, and control inputs si-

multaneously, the constraints $\mathcal{V}$, $\mathcal{A}$, and $\mathcal{U}$ are combined in a set $\mathcal{C}_{v\bar{a}\mu}$:

$$\begin{aligned}
\mathcal{C}_{v\bar{a}\mu} &:= \left(\left\{ \begin{bmatrix} v \\ \bar{a} \end{bmatrix} | v \in \mathcal{V}, \begin{bmatrix} v \\ \bar{a} \end{bmatrix} \in \mathcal{A} \right\} \times \mathbb{R}^2 \right) \cap \mathcal{U} \\
&= \left\{ \begin{bmatrix} v \\ \bar{a} \\ \mu \end{bmatrix} | v_{X,\min} \leq v_X \leq v_{X,\max}, |v_Y| \leq v_{Y,\max}, |\omega| \leq \omega_{\max}, \right. \\
&|l_\mathrm{r}\omega - v_Y| \leq v_X \tan(\alpha_{\max}),\ |v_Y| \leq v_X \tan(\alpha_{\mathrm{bss},\max}), \\
&|\bar{a}_X + v_Y\omega| \leq \bar{a}_{X,\max}, |\bar{a}_Y - v_X\omega| \leq \bar{a}_{Y,\max}, \\
&\left| \delta - \frac{l_\mathrm{f}\omega + v_Y}{v_X} \right| \leq \alpha_{\max},\ \left| \delta^2 - \frac{l_\mathrm{f}\omega + v_Y}{v_X}\delta + \frac{m}{\bar{c}_Y}\bar{a}_X \right| \leq \frac{\bar{c}_X}{\bar{c}_Y} s_{X,\max}, \\
&\left. |\delta| \leq \delta_{\max},\ \delta^3 - \delta^2 \frac{l_\mathrm{f}\omega + v_Y}{v_X} + \delta \left(\frac{m}{\bar{c}_Y}\bar{a}_X + \frac{1}{\gamma} \right) + \frac{\omega(l_\mathrm{r} - l_\mathrm{f}) - 2v_Y}{\gamma v_X} - \frac{m}{\gamma \bar{c}_Y}\bar{a}_Y = 0 \right\}.
\end{aligned} \tag{6.10}$$

This set characterizes velocities, accelerations, and inputs which are admissible based on the control law from Algorithm 5.1.

6.2. Admissible Reference Trajectories

The objective of this section is to derive information about the admissibility of velocity and acceleration values which, if used by the trajectory planning layer, result in reference trajectories which can be tracked such that all constraints from the previous section hold. The set $\mathcal{C}_{v\bar{a}\mu}$ in (6.10) characterizes admissible combinations of vehicle velocities and accelerations, yet under consideration of control inputs μ. Information regarding these are not available on the planning layer when plans are made; nonetheless, $\mathcal{C}_{v\bar{a}\mu}$ provides valuable information, as it allows to derive a characterization of the velocities and accelerations for which the *existence* of admissible control inputs can be guaranteed under a given control law. To that end, a (for computational efficiency polytopic) inner approximation $\mathcal{C}_{v\bar{a}}$ of the projection of the set (6.10) on the space of velocities and accelerations:

$$\mathcal{C}_{v\bar{a}} := \left\{ \begin{bmatrix} v \\ \bar{a} \end{bmatrix} | A_{va} \begin{bmatrix} v \\ \bar{a} \end{bmatrix} \leq b_{va} \right\} \subseteq \mathtt{proj}_{v\bar{a}} \mathcal{C}_{v\bar{a}\mu} = \left\{ \begin{bmatrix} v \\ \bar{a} \end{bmatrix} | \exists \mu : \begin{bmatrix} v \\ \bar{a} \\ \mu \end{bmatrix} \in \mathcal{C}_{v\bar{a}\mu} \right\}, \tag{6.11}$$

is determined. The computation of this set is detailed in Appendix F.

In order to analyze the implications on the reference trajectory resulting from the set $\mathcal{C}_{v\bar{a}}$, the inequality in the definition of (6.11) is written as:

$$A_{va[:,1:2]} \begin{bmatrix} v_X \\ v_Y \end{bmatrix} + A_{va[:,4:5]} \begin{bmatrix} \bar{a}_X \\ \bar{a}_Y \end{bmatrix} + A_{va[:,3]}\omega \leq b_{va}.$$

Replacing the vehicle states and external accelerations by error and reference quantities according to (5.21) and (5.31) results in:

$$\begin{aligned}
&A_{va[:,1:2]}R(e_\psi)^\mathsf{T}\begin{bmatrix}\dot{s}\\0\end{bmatrix} + A_{va[:,4:5]}R(e_\psi)^\mathsf{T}\begin{bmatrix}\ddot{s}\\\dot{\theta}\dot{s}\end{bmatrix} + A_{va[:,3]}\dot{\theta}\ldots\\
&+ A_{va[:,3]}e_\omega + A_{va[:,1:2]}R(e_\psi)^\mathsf{T}\begin{bmatrix}0 & -\dot{\theta} & 1 & 0\\ \dot{\theta} & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}e_{\text{pos}}\\ \dot{e}_{\text{pos}}\end{bmatrix}\ldots\\
&- A_{va[:,4:5]}R(e_\psi)^\mathsf{T}\begin{bmatrix}\dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta}\\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0\end{bmatrix}\begin{bmatrix}e_{\text{pos}}\\ \dot{e}_{\text{pos}}\end{bmatrix} + A_{va[:,4:5]}R(e_\psi)^\mathsf{T}\tilde{\mu} \leq b_{va}.
\end{aligned}$$

This inequality determines the following set of admissible errors and reference values:

$$\begin{aligned}
\tilde{\mathcal{C}}_{\bar{x}e} := \Bigg\{ &\begin{bmatrix}\dot{s}\\\dot{\theta}\\\ddot{s}\\\dot{\theta}\dot{s}\end{bmatrix}, \begin{bmatrix}e_{\text{pos}}\\\dot{e}_{\text{pos}}\\e_\psi\\e_\omega\end{bmatrix} \mid \begin{bmatrix}A_{va[:,1:2]} & A_{va[:,4:5]}\end{bmatrix}\begin{bmatrix}R(e_\psi)^\mathsf{T} & 0\\ 0 & R(e_\psi)^\mathsf{T}\end{bmatrix}\begin{bmatrix}\dot{s}\\0\\\ddot{s}\\\dot{\theta}\dot{s}\end{bmatrix}\ldots \\
&+ A_{va[:,3]}\dot{\theta} + A_{va[:,3]}e_\omega + A_{va[:,4:5]}R(e_\psi)^\mathsf{T}\tilde{\mu}\ldots\\
&+ \begin{bmatrix}A_{va[:,1:2]} & -A_{va[:,4:5]}\end{bmatrix}\begin{bmatrix}R(e_\psi)^\mathsf{T} & 0\\ 0 & R(e_\psi)^\mathsf{T}\end{bmatrix}\begin{bmatrix}0 & -\dot{\theta} & 1 & 0\\ \dot{\theta} & 0 & 0 & 1\\ \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta}\\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0\end{bmatrix}\begin{bmatrix}e_{\text{pos}}\\\dot{e}_{\text{pos}}\end{bmatrix} \leq b_{va}\Bigg\}.
\end{aligned} \tag{6.12}$$

The appearance of error terms prevents the use of (6.12) on the planning layer because future tracking errors are unknown during the planning stage. However, as will be shown in Chapter 7, it is valid to assume that the error state is confined to a bounded set:

Assumption 6.1 (Bounded Error)**.** *Assume that there exist positive scalars ξ, $e_{\psi,\max}$, and $e_{\omega,\max}$ such that during tracking of a reference trajectory, it holds for the error state vector e:*

$$e \in \mathcal{E} := \left\{ \begin{bmatrix}e_{pos}^\mathsf{T} & \dot{e}_{pos}^\mathsf{T} & e_{yaw}^\mathsf{T}\end{bmatrix}^\mathsf{T} \mid \left|\left|\begin{bmatrix}e_{pos}\\\dot{e}_{pos}\end{bmatrix}\right|\right|^2 \leq \xi,\ |e_\psi| \leq e_{\psi,\max},\ |e_\omega| \leq e_{\omega,\max} \right\}. \tag{6.13}$$

If constraints on the reference trajectory parameters are formulated in a way such that they hold *for all* errors $e \in \mathcal{E}$, no information regarding the error at a certain time in online operation is required. As a second requirement for the set of admissible reference parameters, computational efficiency requires a simple set representation such as a polytopic set, whereas the set (6.12) contains nonlinear terms. Thus, the objective is to determine:

$$\tilde{\mathcal{C}}_{\bar{x}} := \left\{ \begin{bmatrix}\dot{s}\\\dot{\theta}\\\ddot{s}\\\dot{\theta}\dot{s}\end{bmatrix} \mid \tilde{A}_{\bar{x}}\begin{bmatrix}\dot{s}\\\dot{\theta}\\\ddot{s}\\\dot{\theta}\dot{s}\end{bmatrix} \leq \tilde{b}_{\bar{x}} \right\} \subseteq \left\{ \begin{bmatrix}\dot{s}\\\dot{\theta}\\\ddot{s}\\\dot{\theta}\dot{s}\end{bmatrix} \mid \begin{bmatrix}\dot{s}\\\dot{\theta}\\\ddot{s}\\\dot{\theta}\dot{s}\\e\end{bmatrix} \in \tilde{\mathcal{C}}_{\bar{x}e} \forall e \in \mathcal{E} \right\}.$$

A solution to this problem is given by the following theorem:

Theorem 6.1. *Given a set $\mathcal{E}$ of possible errors as in Assumption 6.1, assume without loss of generality that $\left|\left|\begin{bmatrix} A_{va[i,1:2]} & A_{va[i,4:5]} \end{bmatrix}\right|\right| = 1$ for all n_{va} rows of A_{va}, $i = 1, 2, \ldots, n_{va}$, and define a function $\tilde{b} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^{n_{va}}$, where*

$$\tilde{b}_{[i]}(\dot{\theta}_{\max}, \ddot{\theta}_{\max}) := |A_{va[i,3]}| e_{\omega,\max} + \max_{\substack{|\dot{\theta}| \leq \dot{\theta}_{\max}, \\ |\ddot{\theta}| \leq \ddot{\theta}_{\max}}} \left|\left| \begin{bmatrix} 0_{2\times 4} \\ -K \end{bmatrix} + \begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \\ \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix} \right|\right|.$$

Also, define

$$\tilde{b}_{\bar{x}} := b_{va} - \tilde{b}(\dot{\theta}_{\max}, \ddot{\theta}_{\max}),$$

a map $R_p : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^{4\times 4}$ using the Kronecker product $\otimes$:

$$R_p(p_1, p_2) := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} p_1 & p_2 \\ -p_2 & p_1 \end{bmatrix},$$

and

$$\tilde{A}_{\bar{x}} := \begin{bmatrix} \begin{bmatrix} A_{va[:,1:2]} & A_{va[:,4:5]} \end{bmatrix} R_p(p_{1,\min}, p_{2,\min}) \\ \begin{bmatrix} A_{va[:,1:2]} & A_{va[:,4:5]} \end{bmatrix} R_p(p_{1,\min}, p_{2,\max}) \\ \begin{bmatrix} A_{va[:,1:2]} & A_{va[:,4:5]} \end{bmatrix} R_p(p_{1,\max}, p_{2,\max}) \\ \begin{bmatrix} A_{va[:,1:2]} & A_{va[:,4:5]} \end{bmatrix} R_p(p_{1,\max}, p_{2,\min}) \end{bmatrix} + \begin{bmatrix} A_{va[:,3]} \\ A_{va[:,3]} \\ A_{va[:,3]} \\ A_{va[:,3]} \end{bmatrix} \begin{bmatrix} 0 & \dot{\theta} & 0 & 0 \end{bmatrix}, \tag{6.14}$$

where $p_{1,\min} = \cos(e_{\psi,\max})$, $p_{1,\max} = 1$, $p_{2,\min} = -\sin(e_{\psi,\max}) = -p_{2,\max}$. Then, if there exist $\dot{\theta}_{\max}$ and $\ddot{\theta}_{\max}$ such that the set

$$\tilde{\mathcal{C}}_{\bar{x}} = \left\{ \begin{bmatrix} \dot{s} & \dot{\theta} & \ddot{s} & \dot{\theta}\dot{s} \end{bmatrix}^{\mathsf{T}} | \tilde{A}_{\bar{x}} \begin{bmatrix} \dot{s} & \dot{\theta} & \ddot{s} & \dot{\theta}\dot{s} \end{bmatrix}^{\mathsf{T}} \leq \tilde{b}_{\bar{x}} \right\}$$

is non-empty and $\texttt{proj}_{\dot{\theta}} \tilde{\mathcal{C}}_{\bar{x}} \subseteq [-\dot{\theta}_{\max}, \dot{\theta}_{\max}]$, it holds that $\tilde{\mathcal{C}}_{\bar{x}} \times \mathcal{E} \subset \tilde{\mathcal{C}}_{\bar{x}e}$.

Proof. For each row of the left-hand side of the inequalities in (6.12), it holds that:

$$\begin{aligned}
&\begin{bmatrix} A_{va[i,1:2]} & A_{va[i,4:5]} \end{bmatrix} \begin{bmatrix} R(e_\psi)^\mathsf{T} & 0 \\ 0 & R(e_\psi)^\mathsf{T} \end{bmatrix} \begin{bmatrix} \dot{s} \\ 0 \\ \ddot{s} \\ \dot{\theta}\dot{s} \end{bmatrix} + A_{va[i,3]}\dot{\theta} + A_{va[i,3]}e_\omega + A_{va[i,4:5]}R(e_\psi)^\mathsf{T}\tilde{\mu} \dots \\
&+ \begin{bmatrix} A_{va[i,1:2]} & -A_{va[i,4:5]} \end{bmatrix} \begin{bmatrix} R(e_\psi)^\mathsf{T} & 0 \\ 0 & R(e_\psi)^\mathsf{T} \end{bmatrix} \begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \\ \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} e_{\text{pos}} \\ \dot{e}_{\text{pos}} \end{bmatrix} \\
&\leq \begin{bmatrix} A_{va[i,1:2]} & A_{va[i,4:5]} \end{bmatrix} \begin{bmatrix} R(e_\psi)^\mathsf{T} & 0 \\ 0 & R(e_\psi)^\mathsf{T} \end{bmatrix} \begin{bmatrix} \dot{s} \\ 0 \\ \ddot{s} \\ \dot{\theta}\dot{s} \end{bmatrix} + A_{va[i,3]}\dot{\theta} + \max_{|e_\omega| \leq e_{\omega,\max}} \left| A_{va[i,3]}e_\omega \right| \dots \\
&+ \max_{\substack{|\dot{\theta}| \leq \dot{\theta}_{\max},\, |\ddot{\theta}| \leq \ddot{\theta}_{\max}, \\ \left\| \begin{bmatrix} e_{\text{pos}} \\ \dot{e}_{\text{pos}} \end{bmatrix} \right\| \leq \xi}} \left\| \left(\begin{bmatrix} 0_{2\times 4} \\ -K \end{bmatrix} + \begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \\ \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix} \right) \begin{bmatrix} e_{\text{pos}} \\ \dot{e}_{\text{pos}} \end{bmatrix} \right\| \\
&\leq \begin{bmatrix} A_{va[i,1:2]} & A_{va[i,4:5]} \end{bmatrix} \begin{bmatrix} R(e_\psi)^\mathsf{T} & 0 \\ 0 & R(e_\psi)^\mathsf{T} \end{bmatrix} \begin{bmatrix} \dot{s} \\ 0 \\ \ddot{s} \\ \dot{\theta}\dot{s} \end{bmatrix} + A_{va[i,3]}\dot{\theta} \dots \qquad (6.15) \\
&+ \left| A_{va[i,3]} \right| e_{\omega,\max} + \max_{|\dot{\theta}| \leq \dot{\theta}_{\max},\, |\ddot{\theta}| \leq \ddot{\theta}_{\max}} \left\| \left(\begin{bmatrix} 0_{2\times 4} \\ -K \end{bmatrix} + \begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \\ \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix} \right) \right\| \cdot \frac{1}{\xi^2}.
\end{aligned}$$

Furthermore, for all $e_\psi \in \texttt{proj}_{e_\psi}\mathcal{E}$,

$$\begin{aligned}
&\begin{bmatrix} R(e_\psi)^\mathsf{T} & 0 \\ 0 & R(e_\psi)^\mathsf{T} \end{bmatrix} \in \left\{ R_p(p_1,p_2) \middle| \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \in [p_{1,\min}, p_{1,\max}] \times [p_{2,\min}, p_{2,\max}] \right\} \\
&= \texttt{convh}\left(\left\{ \begin{bmatrix} p_{1,\min} \\ p_{2,\min} \end{bmatrix}, \begin{bmatrix} p_{1,\min} \\ p_{2,\max} \end{bmatrix}, \begin{bmatrix} p_{1,\max} \\ p_{2,\min} \end{bmatrix}, \begin{bmatrix} p_{1,\max} \\ p_{2,\max} \end{bmatrix} \right\} \right) =: \mathcal{R}.
\end{aligned}$$

Thus,

$$\begin{bmatrix} A_{va[:,1:2]} & A_{va[:,4:5]} \end{bmatrix} R_p(p_1,p_2) \begin{bmatrix} \dot{s} & 0 & \ddot{s} & \dot{\theta}\dot{s} \end{bmatrix}^\mathsf{T} + A_{va[:,3]}\dot{\theta} \leq \tilde{b}_{\bar{x}} \;\forall \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \in \mathtt{v}(\mathcal{R})$$

implies that (6.12) holds for all $e \in \mathcal{E}$; replacing the *all* quantifier by enumeration of the inequalities resulting for each vertex eventually gives the definition of $\tilde{A}_{\bar{x}}$ as in (6.14).

□

Combination of $\tilde{\mathcal{C}}_{\bar{x}}$ with interval constraints on θ, $\ddot{\theta}$, and $\dddot{s}$ allows to characterize admissible values of a reference trajectory:

$$\mathcal{C}_{\bar{x}} := \left\{ \bar{x} \middle| \begin{bmatrix} \dot{s} \\ \dot{\theta} \\ \ddot{s} \\ \dot{\theta}\dot{s} \end{bmatrix} \in \tilde{\mathcal{C}}_{\bar{x}}, |\ddot{\theta}| \leq \ddot{\theta}_{\max},\ |\dddot{s}| \leq \dddot{s}_{\max},\ |\theta| \leq \theta_{\max} \right\} \subset \mathbb{R}^7. \tag{6.16}$$

The different set operations required to derive this set are summarized in Figure 6.1. Note that the set $\mathcal{C}_{\bar{x}}$ is unbounded as no constraints on s are imposed, which is, however, unproblematic. The set (6.16) encodes constraints on a reference trajectory in the Frenet frame, while the planning procedure in Part I is based on the cartesian frame. This mismatch can be overcome by either planning in the Frenet frame or transforming the constraints to the cartesian frame according to Appendix B. Planning in the Frenet frame is difficult because collision checking requires a nonlinear transformation to cartesian coordinates at run-time. A transformation of (6.16) to the cartesian frame, on the other hand, is also nonlinear, but at least not required at run-time, making it the more promising option. However, even though it was possible to obtain admissible trajectories in this thesis (cf. the example in Chapter 8), an automated solution of this problem is beyond the scope of this work and must be considered an open problem. Note that in any case, it is possible to employ $\mathcal{C}_{\bar{x}}$ to quickly assess the admissibility of a planned trajectory. Given an admissible reference trajectory, it now is possible to analyze properties of the error dynamics of the vehicle under the control law devised in the previous chapter, which is carried out in the following.

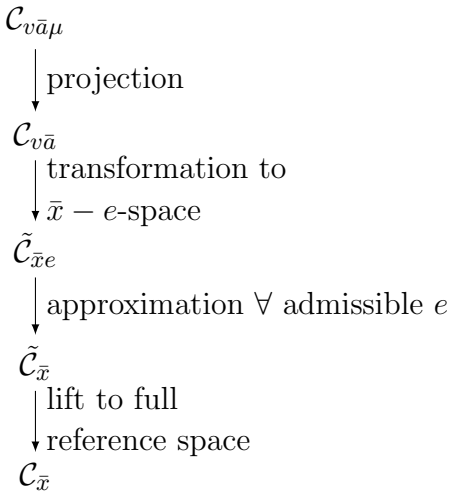


Figure 6.1.: Relation of the different sets used in Section 6.2.

7. Boundedness of the Tracking Error

As motivated in Chapter 5, safe trajectory tracking requires guarantees regarding the boundedness of the tracking error in the sense of Assumption 6.1. The following chapter justifies this assumption, building on the notion of *quadratic boundedness* as defined in [28]:

Definition 7.1 (Quadratic Boundedness)**.** *Given a bounded and closed set $\mathcal{W} \subset \mathbb{R}^{n_w}$ and a function $V(x) = x^\mathsf{T} P x$, $x \in \mathbb{R}^{n_x}$, $P \in \mathbb{R}^{n_x \times n_x}$, a system:*

$$\dot{x} = Ax + B_w w \tag{7.1}$$

with $w \in \mathcal{W}$ is quadratically bounded with Lyapunov matrix P *if P is a positive definite symmetric matrix and if*

$$x^\mathsf{T} P x > 1 \Rightarrow \frac{\mathsf{d}}{\mathsf{d}t} V(x) < 0$$

for all $w \in \mathcal{W}$.

According to [28], using arguments from Lyapunov analysis, the following holds for a quadratically bounded system:

Proposition 7.1. *The set*

$$\left\{ x | x^\mathsf{T} P x \leq 1 \right\}$$

is positive invariant *for the dynamics* (7.1)*, that is, every trajectory starting in the set remains in it forever.*

Define the tracking error dynamics Σ_e as combination of (5.24) and (5.32):

$$\Sigma_e : \qquad \frac{\mathsf{d}}{\mathsf{d}t} \begin{bmatrix} e_{\text{pos}} \\ \dot{e}_{\text{pos}} \\ e_{\text{yaw}} \end{bmatrix} = f_e(t, e, \bar{x}, \Delta a) := \begin{bmatrix} f_{\text{pos}}(t, e, \bar{x}, \Delta a) \\ f_{\text{yaw}}(t, e, \bar{x}, \Delta a) \end{bmatrix}. \tag{7.2}$$

Fig. 7.1 shows the system structure: the position error dynamics Σ_{pos} influences the yaw error dynamics Σ_{yaw}, together with the planning layer. The dashed line indicates a minor interaction, reflecting the impact of Σ_{yaw} on Σ_{pos} through the orientation error e_ψ and the disturbance Δa. This interaction vanishes when no uncertainty is considered such that Σ_{yaw} becomes the zero dynamics of the system, as is the case in [57, 148].

Stability would imply boundedness of the error quantities, but is difficult to proof for time-varying, nonlinear systems. In cases where the reference trajectory is a solution of

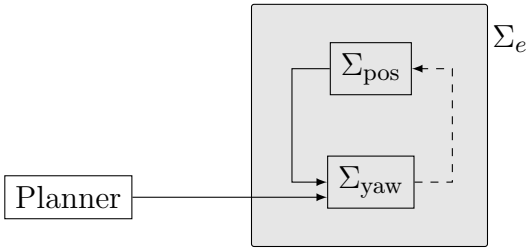


Figure 7.1.: Decomposition of the tracking error dynamics into interconnected subsystems.

the system equations, the origin $e = 0$ of the error dynamics can be an equilibrium. In the considered application, however, this is not the case and a constant equilibrium state might not even exist in general. Rather, a so-called steady-state response will result when tracking a reference trajectory for an undisturbed system:

Definition 7.2 (Steady state response [73])**.** *For the controlled nominal dynamics* $\dot{e} = f_e(t, e, \bar{x}(\cdot), 0)$ *of a system with* $f_e(t, 0, 0, 0) = 0$*, let* $e(t, e_0, \bar{x}(\cdot), 0)$ *denote the trajectory resulting for a given reference trajectory* $\bar{x}(\cdot)$ *and an initial state* e_0*. Suppose there exists an initial state* e^* *and reference* $\bar{x}^*(\cdot)$ *such that*

$$\lim_{t\to\infty} ||e(t, e_0, \bar{x}^*(\cdot), 0) - e(t, e^*, \bar{x}^*(\cdot), 0)|| = 0$$

for every e_0 *in some neighborhood of* e^**. Then,*

$$e_{ssr}(t) = e(t, e^*, \bar{x}^*(\cdot), 0)$$

is called steady state response *to* $\bar{x}^*(\cdot)$*.*

Given a steady state response for a specific reference trajectory, the stability of the considered dynamics relative to this trajectory (rather than relative to a constant equilibrium) can be analyzed. Existence and computation of a steady state response are discussed by [73] in the context of *periodic* reference trajectories as generated by an autonomous reference generator. In that case, the steady state trajectory results as solution of a nonlinear system of partial differential equations. An extension to non-periodic reference trajectories is not given in [73] and would be of questionable practical value in the considered case because it would have to be solved for every admissible reference trajectory separately, which is impossible for an infinite number. Therefore, no attempt to analyze the stability of the error dynamics is made in this chapter.

Instead, the focus is put on *boundedness* of the error quantities, which does not hinge on the notion of equilibrium. The following approach is pursued: the dynamics (7.2) is decomposed into the two subsystems Σ_{pos} and Σ_{yaw}. Assuming that the vehicle velocities and accelerations do not leave the admissible domain $\mathcal{C}_{v\bar{a}}$ as given by (6.11), the worst-case impact of Σ_{yaw} on Σ_{pos}, whose magnitude depends only on the uncertainty Δa, but not on e_ψ, can be bounded independently of the state of Σ_{yaw}, cf. Appendix C. This resolves the feedback interconnection of the subsystems by disconnecting the dashed line in Figure 7.1

and allows analysis as *cascade* interconnection, where the state of Σ_{pos} has influence on Σ_{yaw}, but not vice versa. Then, boundedness of the state trajectories of each subsystem can be analyzed separately. Unlike in the case of a feedback interconnection, where a small-gain like argument would be required, boundedness of the subystems' states implies boundedness of the overall system in a cascade structure [73].

In the case of Σ_{pos}, state boundedness and the magnitude of the impact on Σ_{yaw} can be influenced by design of the feedback gain K in (5.23). In the case of Σ_{yaw}, no such degree of freedom exists. Rather, an analysis must consider both the influence of Σ_{pos} and of the reference trajectory.

Regarding the reference trajectory, the admissibility of single *values* is considered, not of *temporal sequences* thereof. While this removes all temporal information, it allows to cover the entire reference set at once, without considering separate trajectories. Also, it is not necessary to specifically account for the finite duration of reference trajectories, which would otherwise require special care when analyzing asymptotic properties such as stability or boundedness.

7.1. Feedback Gain Synthesis

Preparing the computation of the feedback gain K used in (5.23), (5.22) is rewritten as:

$$\frac{\mathsf{d}}{\mathsf{d}t}\begin{bmatrix} e_{\text{pos}} \\ \dot{e}_{\text{pos}} \end{bmatrix} = A\begin{bmatrix} e_{\text{pos}} \\ \dot{e}_{\text{pos}} \end{bmatrix} + B_u\tilde{\mu} + B_w w, \tag{7.3}$$

with

$$A := \begin{bmatrix} 0_{2\times 2} & I_{2\times 2} \\ 0_{2\times 2} & 0_{2\times 2} \end{bmatrix},\ B_u := \begin{bmatrix} 0_{2\times 2} \\ I_{2\times 2} \end{bmatrix},\ B_w := \Delta a_{\max}\cdot B_u,\ w := \frac{1}{\Delta a_{\max}}\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ R(e_\psi) \end{bmatrix}\begin{bmatrix} \Delta a_X \\ \Delta a_Y \end{bmatrix}.$$

Appendix C details the computation of $\Delta a_{\max}$ with $0 < \left|\left|\begin{bmatrix} \Delta a_X \\ \Delta a_Y \end{bmatrix}\right|\right| \leq \Delta a_{\max}$, such that

$$\mathcal{W} := \left\{ w | w^{\mathsf{T}} w \leq 1 \right\}.$$

The objective in the design of the feedback controller K is to ensure quadratic boundedness of e_{pos}, $\dot{e}_{\text{pos}}$, and consequently $\tilde{\mu}$ despite the impact of w, where the magnitude of the states should be as small as possible in order to reduce the impact of Σ_{pos} on Σ_{yaw}. The following problem formulation is considered:

Problem 7.1 (Synthesis)**.** *Given a positive scalar ϵ_1 and assuming that $w^{\mathsf{T}} w \leq 1$, determine matrices $X \in \mathbb{R}^{4\times 4}$, $X = X^{\mathsf{T}}$, $Y \in \mathbb{R}^{2\times 4}$, and scalars ξ_1 and ξ_2 by solving:*

$$\min_{X,Y,\xi_1,\xi_2} \xi_1 + \xi_2$$

subject to:

$$AX + XA^\mathsf{T} + B_u Y + Y^\mathsf{T} B_u^\mathsf{T} + \frac{B_w B_w^\mathsf{T}}{\epsilon_1} + \epsilon_1 X \leq 0, \tag{7.4}$$

$$\begin{bmatrix} X & Y^\mathsf{T} \\ Y & \xi_2 I \end{bmatrix} \geq 0,\ X > 0,\ \xi_1 I - X \geq 0,\ \xi_2 > 0,\ \xi_1 > 0. \tag{7.5}$$

This problem is a semidefinite program and can be solved efficiently using numerical solvers such as [133] or [110]. Similar settings have been considered by [1, 26, 124, 140]. Combining several results from [26] leads to the following theorem:

Theorem 7.1 (Bounded state of the position error dynamics Σ_{pos})**.** *Given X, Y, ξ_1, and ξ_2 as solution to Problem 7.1 and letting $K = -YX^{-1}$, the system* (7.3) *is quadratically bounded with Lyapunov matrix $P_{pos} := X^{-1}$, such that*

$$\mathcal{E}_{pos} := \left\{ \begin{bmatrix} e_{pos} \\ \dot{e}_{pos} \end{bmatrix} \mid \begin{bmatrix} e_{pos}^\mathsf{T} & \dot{e}_{pos}^\mathsf{T} \end{bmatrix} P_{pos} \begin{bmatrix} e_{pos} \\ \dot{e}_{pos} \end{bmatrix} \leq 1 \right\}$$

is an invariant set for the position error dynamics (5.24)*. Furthermore, it holds that* $\begin{bmatrix} e_{pos}^\mathsf{T} & \dot{e}_{pos}^\mathsf{T} \end{bmatrix}^\mathsf{T} \begin{bmatrix} e_{pos} \\ \dot{e}_{pos} \end{bmatrix} \leq \xi_1$ *and* $\tilde{\mu}^\mathsf{T}\tilde{\mu} \leq \xi_2$ *for all times* $t \geq 0$*.*

Proof. With $A_{\text{cl}} := A - B_u K$ and temporarily letting $x := \begin{bmatrix} e_{\text{pos}}^\mathsf{T} & \dot{e}_{\text{pos}}^\mathsf{T} \end{bmatrix}^\mathsf{T}$, according to the S-procedure [26], the implication:

$$\begin{aligned} \left(\begin{bmatrix} x \\ w \end{bmatrix}^\mathsf{T} \begin{bmatrix} P_{\text{pos}} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} - 1 \geq 0 \bigwedge 1 - \begin{bmatrix} x \\ w \end{bmatrix}^\mathsf{T} \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} \geq 0 \right) \\ \Rightarrow - \begin{bmatrix} x \\ w \end{bmatrix}^\mathsf{T} \begin{bmatrix} A^\mathsf{T} P_{\text{pos}} + P_{\text{pos}} A & P_{\text{pos}} B_w \\ (P_{\text{pos}} B_w)^\mathsf{T} & 0 \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} \geq 0 \end{aligned} \tag{7.6}$$

holds if and only if there exist $\epsilon_1 > 0$, $\epsilon_2 > 0$ such that *for all* x, w:

$$\begin{aligned} - \begin{bmatrix} x \\ w \end{bmatrix}^\mathsf{T} \begin{bmatrix} A^\mathsf{T} P_{\text{pos}} + P_{\text{pos}} A & P_{\text{pos}} B_w \\ (P_{\text{pos}} B_w)^\mathsf{T} & 0 \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} - \epsilon_2 \left(1 - \begin{bmatrix} x \\ w \end{bmatrix}^\mathsf{T} \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} \right) \dots \\ - \epsilon_1 \left(\begin{bmatrix} x \\ w \end{bmatrix}^\mathsf{T} \begin{bmatrix} P_{\text{pos}} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} - 1 \right) \geq 0 \end{aligned}$$

or equivalently,

$$\begin{bmatrix} x \\ w \\ 1 \end{bmatrix}^\mathsf{T} \begin{bmatrix} A^\mathsf{T} P_{\text{pos}} + P_{\text{pos}} A + \alpha P_{\text{pos}} & P_{\text{pos}} B_w & 0 \\ (P_{\text{pos}} B_w)^\mathsf{T} & -\epsilon_2 I & 0 \\ 0 & 0 & \epsilon_2 - \epsilon_1 \end{bmatrix} \begin{bmatrix} x \\ w \\ 1 \end{bmatrix} \leq 0. \tag{7.7}$$

Without loss of generality, it is possible to let $\epsilon_1 = \epsilon_2$ [26]. With $X = P_{\text{pos}}^{-1}$ and $Y := -KX$, (7.7) is equivalent to:

$$\begin{bmatrix} X & 0 \\ 0 & I \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} A^{\mathsf{T}}P_{\text{pos}} + P_{\text{pos}}A + \epsilon_1 P_{\text{pos}} & P_{\text{pos}}B_w \\ (P_{\text{pos}}B_w)^{\mathsf{T}} & -\epsilon_1 I \end{bmatrix} \begin{bmatrix} X & 0 \\ 0 & I \end{bmatrix}$$
$$= \begin{bmatrix} AX + XA^{\mathsf{T}} + \epsilon_1 X + B_u Y + Y^{\mathsf{T}}B_u^{\mathsf{T}} & B_w \\ B_w^{\mathsf{T}} & -\epsilon_1 I \end{bmatrix} \leq 0,$$

or, by the Schur complement, to (7.4). With $V(x) := x^{\mathsf{T}}P_{\text{pos}}x$ and

$$\dot{V}(x) = x^{\mathsf{T}}(A^{\mathsf{T}}P_{\text{pos}} + P_{\text{pos}}A)x + x^{\mathsf{T}}P_{\text{pos}}B_w w + w^{\mathsf{T}}B_w^{\mathsf{T}}P_{\text{pos}}x$$
$$= \begin{bmatrix} x \\ w \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} A^{\mathsf{T}}P_{\text{pos}} + P_{\text{pos}}A & P_{\text{pos}}B_w \\ (P_{\text{pos}}B_w)^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix},$$

(7.6) (and equivalently (7.4)) implies that $\dot{V}(x) \leq 0$ for all $w^{\mathsf{T}}w \leq 1$ and $x^{\mathsf{T}}P_{\text{pos}}x \geq 1$, thus fulfilling the conditions of Definition 7.1. Invariance of $\mathcal{E}_{\text{pos}}$ follows from Proposition 7.1.

Denote the smallest and largest eigenvalue of a matrix by $\lambda_{\min}$ and $\lambda_{\max}$, respectively. If the initial state is inside the set $\mathcal{E}_{\text{pos}}$, $1 \geq x^{\mathsf{T}}X^{-1}x \geq \lambda_{\min}\left(X^{-1}\right)x^{\mathsf{T}}x \geq 0$. Along with

$$\xi_1 \cdot I - X \geq 0 \Leftrightarrow \lambda_{\max}(X) = \frac{1}{\lambda_{\min}(X^{-1})} \leq \xi_1,$$

it follows that $x^{\mathsf{T}}x \leq \xi_1$. Furthermore, temporarily defining $z := X^{-\frac{1}{2}}x$, it holds that:

$$\begin{bmatrix} X & Y^{\mathsf{T}} \\ Y & \xi_2 I \end{bmatrix} > 0 \Leftrightarrow \left(X - \frac{1}{\xi_2}Y^{\mathsf{T}}Y > 0 \bigwedge \xi_2 > 0\right) \Leftrightarrow \left(\xi_2 I - X^{-\frac{1}{2}^{\mathsf{T}}}Y^{\mathsf{T}}YX^{-\frac{1}{2}} > 0 \bigwedge \xi_2 > 0\right)$$
$$\Leftrightarrow$$
$$\xi_2 > \lambda_{\max}\left(X^{-\frac{1}{2}^{\mathsf{T}}}Y^{\mathsf{T}}YX^{-\frac{1}{2}}\right) = \left|\left|YX^{-\frac{1}{2}}\right|\right|^2 = \max_{||z||=1}\left|\left|YX^{-\frac{1}{2}}z\right|\right|^2 = \max_{x\in\mathcal{E}_{\text{pos}}}\left|\left|YX^{-\frac{1}{2}}X^{-\frac{1}{2}}x\right|\right|^2$$
$$= \max_{x\in\mathcal{E}_{\text{pos}}}\left|\left|-YX^{-1}x\right|\right|^2 = \max_{x\in\mathcal{E}_{\text{pos}}}||-Kx||^2 \geq \max_{t\geq 0}||-Kx(t)||^2 = \max_{t\geq 0}||\tilde{\mu}(t)||^2.$$

□

7.2. Analysis of the Yaw Error Dynamics

As discussed in Section 5.3, the direct analysis of the yaw error dynamics (5.32) is impeded by its nonlinear, time-varying nature. In the following section, a polytopic linear differential inclusion [26, p. 53] is derived from (5.32), relying on the fact that the domain $\mathcal{D}_{\text{yaw}}$ of f_{yaw}:

$$\mathcal{D}_{\text{yaw}} := \mathcal{E} \times \mathcal{C}_{\bar{x}} \times \mathcal{A}_{\Delta} \tag{7.8}$$

is bounded. The differential inclusion will be simpler to analyze, while obtained analysis results carry over to the nonlinear dynamics (5.32).

Theorem 7.2 (Polytopic Linear Differential Inclusion of the Yaw Error Dynamics)**.** *Let there be given polytopes $\mathcal{C}_{\bar{x}}$ and $\mathcal{E}$ as in Sec. 6.2, a set $\mathcal{Y}$ and a polytope $\mathcal{P} \subset \mathbb{R}^4$ such that $\mathcal{Y} \subseteq \mathcal{P}$, and a function $f_{param} : \mathcal{E} \times \mathcal{C}_{\bar{x}} \to \mathcal{Y}$:*

$$f_{param}(e, \bar{x}) := \begin{bmatrix} \ddot{s} \\ \frac{1}{v_X} \\ \frac{\dot{s}}{v_X} \\ \frac{m}{\Theta} l_f \dot{\theta} \dot{s} - \frac{\bar{c}_Y l_r (l_f + l_r)}{\Theta} \frac{\dot{\theta}}{v_X} \end{bmatrix}, \tag{7.9}$$

with $v_X = v_X(e, \bar{x})$ as a function defined by (5.31). Next, define $p := \begin{bmatrix} p_3 & p_5 & p_6 & p_7 \end{bmatrix}^\mathsf{T}$,

$$A_{yaw}(p) := \begin{bmatrix} 0 & 1 \\ -\left(p_6 \frac{\bar{c}_Y (l_f + l_r)}{\Theta} + \frac{m}{\Theta} l_f p_3 \right) & -p_5 \frac{\bar{c}_Y l_r (l_f + l_r)}{\Theta} \end{bmatrix}, \; B_{w_2} := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and, with constants c_1 and c_2 according to Theorem A.2, $w_{1,\max}(p) := c_1 + c_2 \cdot p_5$ and

$$B_{w_1}(p) := \begin{bmatrix} 0 \\ w_{1,\max}(p) \end{bmatrix}. \tag{7.10}$$

Further, introduce the function $\tilde{w}_1 : \mathcal{D}_{yaw} \to \mathbb{R}$:

$$\begin{aligned} \tilde{w}_1(e, \bar{x}, \Delta a) &:= \frac{m}{\Theta} l_f \left(\dot{\theta} \dot{s} (\cos e_\psi - 1) - \ddot{s} (\sin e_\psi - e_\psi) \right) - \frac{\dot{s}}{v_X(e, \bar{x})} \frac{\bar{c}_Y (l_f + l_r)}{\Theta} (\sin e_\psi - e_\psi) \\ &+ \Delta a_\psi - \ddot{\theta} + \begin{bmatrix} 0 & 1 \end{bmatrix} R(e_\psi)^\mathsf{T} \frac{m}{\Theta} l_f K \begin{bmatrix} e_{pos} \\ \dot{e}_{pos} \end{bmatrix} + \dots \qquad (7.11) \\ &+ \begin{bmatrix} 0 & \frac{1}{v_X(e,\bar{x})} \frac{\bar{c}_Y (l_f + l_r)}{\Theta} & 0 & -\frac{m}{\Theta} l_f \end{bmatrix} \begin{bmatrix} R(e_\psi)^\mathsf{T} & 0 \\ 0 & R(e_\psi)^\mathsf{T} \end{bmatrix} \begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \\ \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} e_{pos} \\ \dot{e}_{pos} \end{bmatrix} \end{aligned}$$

and a scalar $w_1 \in \mathbb{R}$. Then, it holds that

$$f_{yaw}(e, \bar{x}, \Delta a) \in \mathcal{F}_{yaw} \left(\begin{bmatrix} e_\psi \\ e_\omega \end{bmatrix} \right) := \left\{ A_{yaw}(p) \begin{bmatrix} e_\psi \\ e_\omega \end{bmatrix} + B_{w_1}(p) w_1 + B_{w_2} p \,\middle|\, w_1^\mathsf{T} w_1 \leq 1, \; p \in \mathcal{P} \right\} \tag{7.12}$$

for all $\begin{bmatrix} e^\mathsf{T} & \bar{x}^\mathsf{T} & \Delta a^\mathsf{T} \end{bmatrix}^\mathsf{T} \in \mathcal{D}_{yaw}$.

Proof. Rewriting (5.21) as:

$$\begin{aligned} \frac{m}{\Theta} l_\mathrm{f} \bar{a}_Y &= \frac{m}{\Theta} l_\mathrm{f} \left(\dot{\theta} \dot{s} \cos e_\psi - \ddot{s} \sin e_\psi \right) + \frac{m}{\Theta} l_\mathrm{f} \begin{bmatrix} 0 & 1 \end{bmatrix} R(e_\psi)^\mathsf{T} \left(\tilde{\mu} - \begin{bmatrix} \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} e_\mathrm{pos} \\ \dot{e}_\mathrm{pos} \end{bmatrix} \right) \\ &= \frac{m}{\Theta} l_\mathrm{f} \left[-\ddot{s} e_\psi - (\sin e_\psi - e_\psi) \ddot{s} + \dot{\theta} \dot{s} + (\cos e_\psi - 1) \dot{\theta} \dot{s} \right] \dots \\ &+ \frac{m}{\Theta} l_\mathrm{f} \begin{bmatrix} 0 & 1 \end{bmatrix} R(e_\psi)^\mathsf{T} \left(\tilde{\mu} - \begin{bmatrix} \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} e_\mathrm{pos} \\ \dot{e}_\mathrm{pos} \end{bmatrix} \right), \end{aligned}$$

the second line of (5.31) as:

$$\begin{aligned}\frac{\bar{c}_Y(l_\mathrm{f}+l_\mathrm{r})}{\Theta}v_Y &= \frac{\bar{c}_Y(l_\mathrm{f}+l_\mathrm{r})}{\Theta}\left(\begin{bmatrix}0 & 1\end{bmatrix}R(e_\psi)^\mathsf{T}\begin{bmatrix}0 & -\dot\theta & 1 & 0\\ \dot\theta & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}e_\mathrm{pos}\\ \dot e_\mathrm{pos}\end{bmatrix} - \dot s\sin e_\psi\right)\\ &= \frac{\bar{c}_Y(l_\mathrm{f}+l_\mathrm{r})}{\Theta}\left(\begin{bmatrix}0 & 1\end{bmatrix}R(e_\psi)^\mathsf{T}\begin{bmatrix}0 & -\dot\theta & 1 & 0\\ \dot\theta & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}e_\mathrm{pos}\\ \dot e_\mathrm{pos}\end{bmatrix} - \dot s e_\psi - \dot s(\sin e_\psi - e_\psi)\right),\end{aligned}$$

and – using the definition of $\tilde{w}_1$ as above – introducing these into (5.30) gives:

$$\begin{aligned}\dot e_\omega = &\left(-\frac{\dot s}{v_X}\frac{\bar{c}_Y(l_\mathrm{f}+l_\mathrm{r})}{\Theta} - \frac{m}{\Theta}l_\mathrm{f}\ddot s\right)e_\psi - \frac{1}{v_X}\frac{\bar{c}_Y l_\mathrm{r}(l_\mathrm{f}+l_\mathrm{r})}{\Theta}e_\omega\ldots\\ &+\tilde{w}_1(e,\bar x,v_X) + \frac{m}{\Theta}l_\mathrm{f}\dot\theta\dot s - \frac{\bar{c}_Y l_\mathrm{r}(l_\mathrm{f}+l_\mathrm{r})}{\Theta}\frac{\dot\theta}{v_X}.\end{aligned}\tag{7.13}$$

As is shown in Theorem A.2 in the appendix, it holds that:

$$\tilde{w}_1(e,\bar x,\Delta a) \le w_{1,\max}\left(f_\mathrm{param}(e,\bar x)\right)\ \forall\left[e^\mathsf{T}\bar x^\mathsf{T}\Delta a^\mathsf{T}\right]^\mathsf{T}\in\mathcal{D}_\mathrm{yaw},$$

such that (because $w_{1,\max}>0$):

$$\frac{\tilde{w}_1(e,\bar x,\Delta a)^\mathsf{T}}{w_{1,\max}(p)}\frac{\tilde{w}_1(e,\bar x,\Delta a)}{w_{1,\max}(p)} \le 1.$$

Letting $w_1 := \frac{\tilde{w}_1}{w_{1,\max}}$ and making substitutions in (7.13) then gives:

$$f_\mathrm{yaw}(e,\bar x,\Delta a)\in\left\{A_\mathrm{yaw}(p)\begin{bmatrix}e_\psi\\ e_\omega\end{bmatrix} + B_{w_1}(p)w_1 + B_{w_2}p \,|\, w_1^\mathsf{T}w_1\le 1,\ p = f_\mathrm{param}(e,\bar x)\right\}.$$

Because by requirement,

$$f_\mathrm{param}(e,\bar x)\in\mathcal{Y}\subseteq\mathcal{P}\ \forall\begin{bmatrix}e^\mathsf{T} & \bar x^\mathsf{T} & \Delta a^\mathsf{T}\end{bmatrix}^\mathsf{T}\in\mathcal{D}_\mathrm{yaw},$$

the result follows. □

An important aspect is the computation of the polytope $\mathcal{P}$ as required in Theorem 7.2. This rather technical derivation is provided in Appendix G. On that basis, the analysis problem can be formulated as a semidefinite program:

Problem 7.2 (Analysis)**.** *Having chosen design parameters ϵ_3, $\epsilon_4>0$, a polytope $\mathcal{E}$ as in Assumption 6.1, a polytope $\mathcal{P}\subset\mathbb{R}^4$, and assuming that $w_1^\mathsf{T}w_1\le 1$, determine a matrix $X\in\mathbb{R}^{2\times 2}$ by solving:*

$$\min_X \quad -\log\det X$$

subject to:

$$X > 0, \tag{7.14}$$

$$\begin{bmatrix} XA_{yaw}(p)^{\mathsf{T}} + A_{yaw}(p)X + \epsilon_3 X & B_{w1}(p) & B_{w_2}p \\ B_{w1}(p)^{\mathsf{T}} & -\epsilon_3 I & 0 \\ (B_{w_2}p)^{\mathsf{T}} & 0 & \epsilon_3 - \epsilon_4 \end{bmatrix} \leq 0 \ \forall p \in \mathtt{v}\left(\mathcal{P}\right), \tag{7.15}$$

$$\begin{bmatrix} \frac{1}{e_{\psi,\max}} & 0 \end{bmatrix} X \begin{bmatrix} \frac{1}{e_{\psi,\max}} \\ 0 \end{bmatrix} \leq 1, \quad \begin{bmatrix} 0 & \frac{1}{e_{\omega,\max}} \end{bmatrix} X \begin{bmatrix} 0 \\ \frac{1}{e_{\omega,\max}} \end{bmatrix} \leq 1, \tag{7.16}$$

with $A_{yaw}(p)$, $B_{w_1}(p)$, and B_{w_2} according to Theorem 7.2.

Extending ideas from [26], the following theorem holds.

Theorem 7.3 (Bounded state of the yaw error dynamics Σ_{yaw})**.** *Given X as solution to Problem 7.2 and defining $P_{yaw} := X^{-1}$, the set*

$$\mathcal{E}_{yaw} := \left\{ \begin{bmatrix} e_\psi \\ e_\omega \end{bmatrix} \mid \begin{bmatrix} e_\psi & e_\omega \end{bmatrix} P_{yaw} \begin{bmatrix} e_\psi \\ e_\omega \end{bmatrix} \leq 1 \right\}$$

is a subset of $\mathtt{proj}_{e_\psi, e_\omega}\mathcal{E}$ and an invariant set for the yaw error dynamics (5.32)*.*

Proof. Along the lines of the proof to (7.1), a function

$$V(e_\psi, e_{\text{yaw}}) = \begin{bmatrix} e_\psi & e_\omega \end{bmatrix} P_{\text{yaw}} \begin{bmatrix} e_\psi \\ e_\omega \end{bmatrix}$$

is defined with

$$\begin{aligned} \dot{V} &= 2 \begin{bmatrix} e_\psi & e_\omega \end{bmatrix} P_{\text{yaw}} f_{\text{yaw}}(e, \bar{x}, \Delta a) \\ &= \begin{bmatrix} e_\psi \\ e_\omega \\ w_1 \\ 1 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} A_{\text{yaw}}(p)^{\mathsf{T}} P_{\text{yaw}} + P_{\text{yaw}} A_{\text{yaw}}(p) & P_{\text{yaw}} B_{w1} & P_{\text{yaw}} B_{w_2} w_2 \\ (P_{\text{yaw}} B_{w1})^{\mathsf{T}} & 0 & 0 \\ (P_{\text{yaw}} B_{w_2} w_2)^{\mathsf{T}} & 0 & 0 \end{bmatrix} \begin{bmatrix} e_\psi \\ e_\omega \\ w_1 \\ 1 \end{bmatrix} \end{aligned}$$

According to the S-procedure, the implication:

$$\left(w_1^{\mathsf{T}} w_1 \leq 1 \bigwedge \begin{bmatrix} e_\psi & e_\omega \end{bmatrix} P_{\text{yaw}} \begin{bmatrix} e_\psi \\ e_\omega \end{bmatrix} \geq 1 \Rightarrow \dot{V} \leq 0 \right)$$

holds if and only if there exist positive scalars ϵ_3 and ϵ_4 such that:

$$\begin{bmatrix} A_{\text{yaw}}(p)^{\mathsf{T}} P_{\text{yaw}} + P_{\text{yaw}} A_{\text{yaw}}(p) + \epsilon_4 P_{\text{yaw}} & P_{\text{yaw}} B_{w_1} & P_{\text{yaw}} B_{w_2} w_2 \\ (P_{\text{yaw}} B_{w_1})^{\mathsf{T}} & -\epsilon_3 I & 0 \\ (P_{\text{yaw}} B_{w_2} w_2)^{\mathsf{T}} & 0 & \epsilon_3 - \epsilon_4 \end{bmatrix} \leq 0 \ \forall p \in \mathcal{P}.$$

Left- and right-multiplying this by $\texttt{diag}\left(P_{\text{yaw}}^{-1}, I, I\right)$, equivalence to (7.15) is established. Defining

$$A_e := \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, \; b_e := \begin{bmatrix} e_{\psi,\max} \\ e_{\psi,\max} \\ e_{\omega,\max} \\ e_{\omega,\max} \end{bmatrix},$$

as noted in [26, Sec. 5.2.2],

$$\mathcal{E}_{\text{yaw}} \subset \texttt{proj}_{e_\psi, e_\omega}\mathcal{E} = \left\{ \begin{bmatrix} e_\psi \\ e_\omega \end{bmatrix} \mid A_e \begin{bmatrix} e_\psi \\ e_\omega \end{bmatrix} \leq b_e \right\},$$

holds if and only if

$$\frac{A_{e[i,:]}}{b_{e[i]}} P_{\text{yaw}}^{-1} \frac{A_{e[i,:]}^\mathsf{T}}{b_{e[i]}} \leq 1$$

for all $i = 1, 2, 3, 4$ as encoded by (7.16). This holds because

$$\mathcal{E}_{\text{yaw}} \subset \texttt{proj}_{e_\psi, e_\omega}\mathcal{E}$$

is equivalent to the requirement that (with $z := P_{\text{yaw}}^{\frac{1}{2}} x$):

$$\begin{aligned} 1 \geq \max \left\{ \frac{A_{e[i,:]}}{b_{e[i]}} x | x \in \mathcal{E}_{\text{yaw}} \right\} &= \max_{z^\mathsf{T} z \leq 1} \frac{A_{e[i,:]}}{b_{e[i]}} P_{\text{yaw}}^{-\frac{1}{2}} z \\ &= \frac{A_{e[i,:]}}{b_{e[i]}} P_{\text{yaw}}^{-\frac{1}{2}} P_{\text{yaw}}^{-\frac{1}{2}} \frac{A_{e[i,:]}}{b_{e[i]}} \cdot \frac{1}{\sqrt{\frac{A_{e[i,:]}}{b_{e[i]}} P_{\text{yaw}}^{-1} \frac{A_{e[i,:]}^\mathsf{T}}{b_{e[i]}}}}, \end{aligned}$$

where the last step is equivalent to (7.16). Thus, (7.15) ensures quadratic boundedness of the differential inclusion (7.12), for which, according to Theorem 7.1, $\mathcal{E}_{\text{yaw}}$ is an invariant set inside $\texttt{proj}_{e_\psi, e_\omega}\mathcal{E}$. Because of Theorem 7.2, it is also an invariant set for the original yaw error dynamics (5.32).

□

In order to illustrate these findings and the interplay of the components derived in this chapter and the preceding ones, the following chapter provides an example along with a discussion of the approach.

8. Examples and Discussion

In the following, the results obtained by theoretical analysis so far in this part are complemented by an analysis based on numerical simulation, which also serves as basis for a discussion of the devised methodology.

8.1. Validation in Simulation

For the purpose of numerical simulation, Simulink [138] models of the proposed controller and the employed vehicle model are used. While the controller design and analysis have been carried out in the continous time domain, numerical simulation relies on fixed- or variable-step discretization of the time domain. In the following, in order to allow for rapid simulation of thousands of test cases, simulations are run with fixed time steps of $T_\mathrm{s} = 10\,\mathrm{ms}$. Because the dynamics under control is a mechanical system, it does not evolve much during this time period, such that this can be assumed to provide a good approximation of the actual system behavior.

The objective is to apply the proposed tracking controller to a reference trajectory obtained for a specific maneuver based on Part I. The tracking performance in terms of constraint compliance is evaluated for a large number of initial error states, randomly chosen from a set $\mathcal{E}_0$, defined as:

$$\mathcal{E}_0 := \left\{ \begin{bmatrix} e_\mathrm{pos}^\mathsf{T} & \dot{e}_\mathrm{pos}^\mathsf{T} & e_\mathrm{yaw}^\mathsf{T} \end{bmatrix}^\mathsf{T} \mid \begin{bmatrix} e_\mathrm{pos}^\mathsf{T} & \dot{e}_\mathrm{pos}^\mathsf{T} \end{bmatrix} P_\mathrm{pos} \begin{bmatrix} e_\mathrm{pos} \\ \dot{e}_\mathrm{pos} \end{bmatrix} \leq 1,\ e_\mathrm{yaw}^\mathsf{T} P_\mathrm{yaw} e_\mathrm{yaw} \leq 1 \right\}$$

based on Theorems 7.1 and 7.3. Questions of measurement errors and robustness against external disturbances are beyond the scope of this thesis; accordingly, this simulation setup relies on perfect information of all required state quantities and considers only those uncertainties as specified in Chapter 5.

For planning purposes, the cooperative overtaking maneuver from Section 4.1.1 is considered, with the slight modification that the longitudinal and the lateral jerk are used as inputs of the ego vehicle instead of the accelerations. Again, cf. Table A.2 and Table 5.1 for the parameter values used throughout the example. The boundedness of the tracking error is analyzed for an uncertainty bound $\Delta a_\mathrm{max} = 0.1\,\mathrm{m\,s^{-2}}$ as defined in (5.6).

The projection of the set $\mathcal{E}_0$ on the yaw error coordinates e_ψ and e_ω is shown in Figure 8.1, along with the trajectories resulting from the random initial states. This phase portrait suggests the invariance of the set; in addition, the quick convergence of the trajectories towards a closer vicinity of the origin suggests the existence of a smaller ellipsoid, i.e., of smaller error bounds.

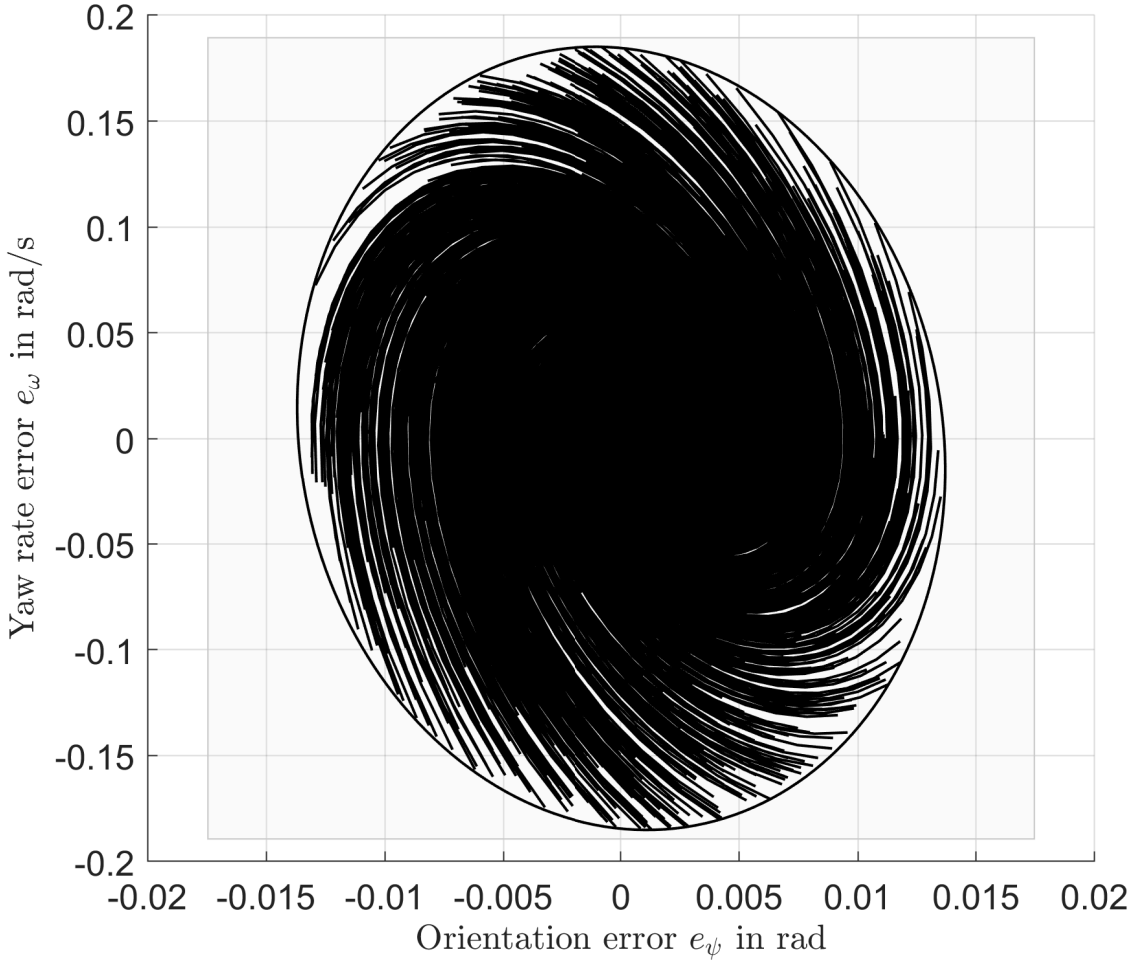
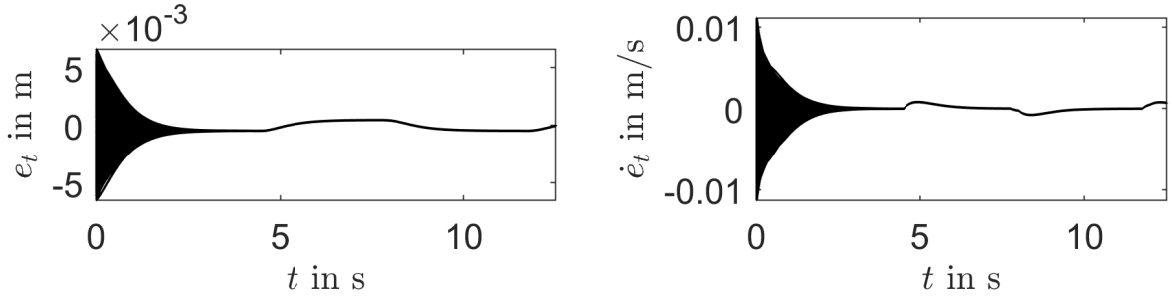


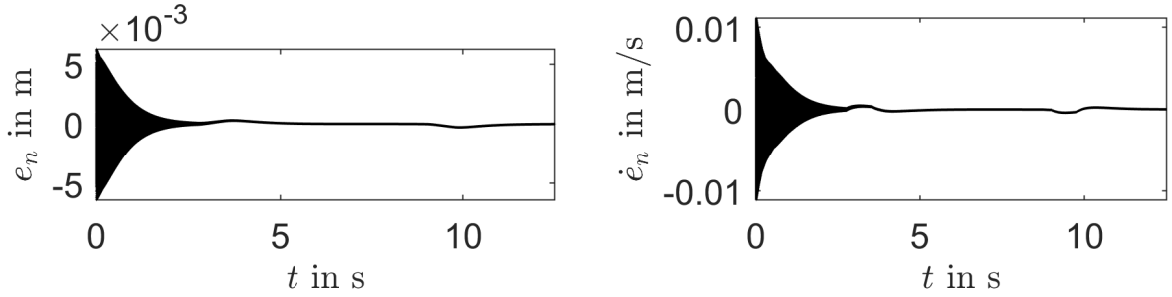
Figure 8.1.: Invariant set $\mathcal{E}_{\text{yaw}}$ (solid black ellipsoid) inside domain of admissible yaw errors (shaded grey) with trajectories starting in randomly chosen initial states close to the set boundary.



(a) Tangential position error e_t.

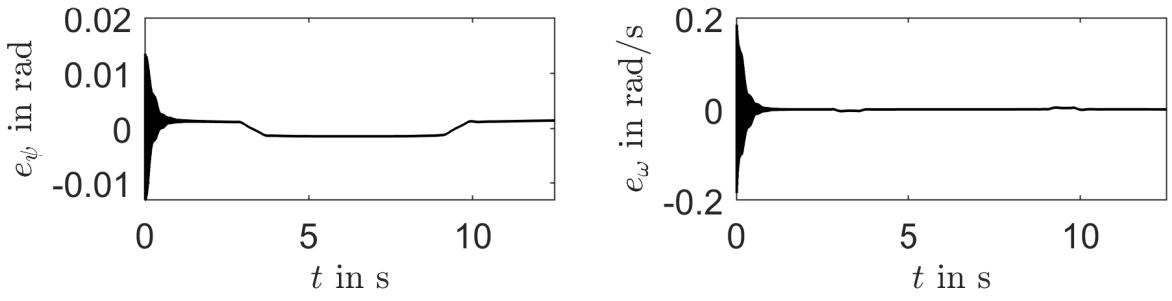
(b) Tangential velocity error $\dot{e}_\text{t}$.

Figure 8.2.: Tangential tracking errors.



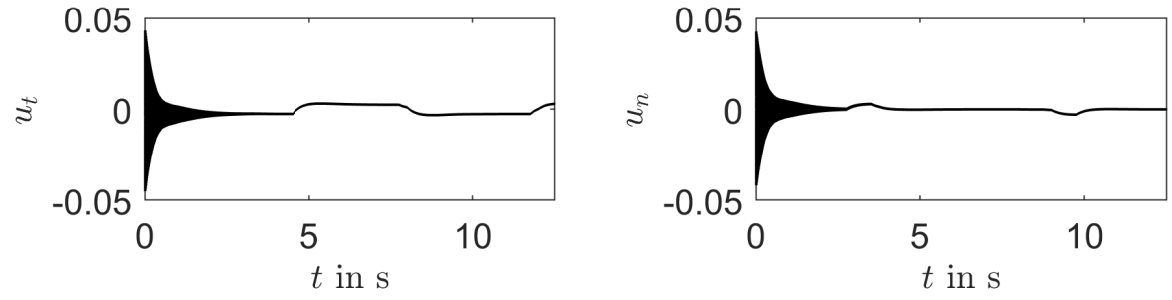
(a) Normal position error e_{n}. (b) Normal velocity error $\dot{e}_{\mathrm{n}}$.

Figure 8.3.: Normal tracking errors.



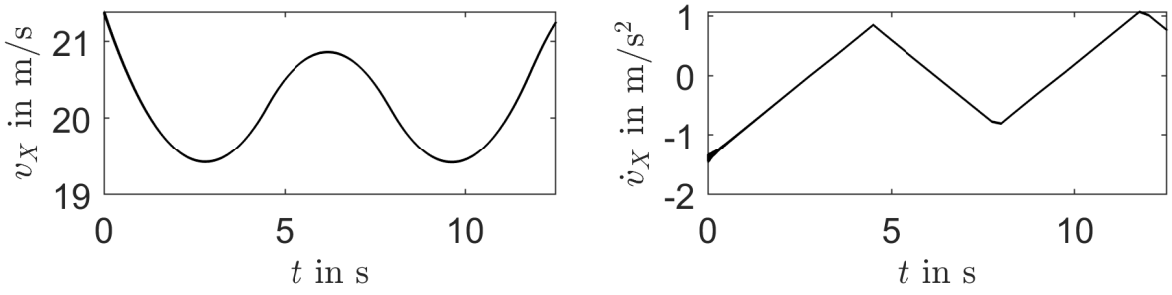
(a) Orientation error e_{ψ}. (b) Yaw rate error e_{ω}.

Figure 8.4.: Yaw errors.



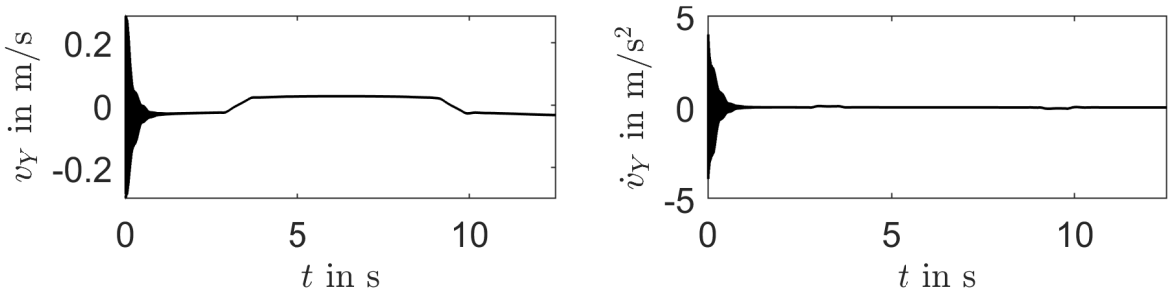
(a) Input u_{t}. (b) Input u_{n}.

Figure 8.5.: Components of the control input vector $\tilde{\mu} = \begin{bmatrix} u_{\mathrm{t}} & u_{\mathrm{n}} \end{bmatrix}^{\mathsf{T}}$.



(a) Longitudinal velocity v_X. (b) Longitudinal acceleration $\dot{v}_X$.

Figure 8.6.: Longitudinal velocity and acceleration.



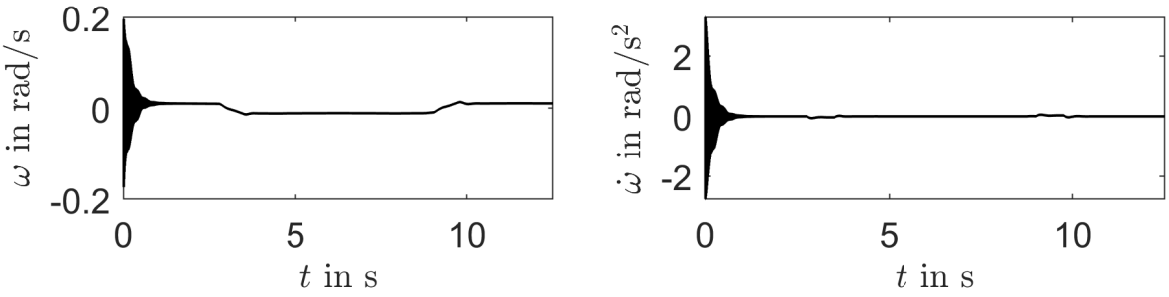
(a) Lateral velocity v_Y. (b) Lateral acceleration $\dot{v}_Y$.

Figure 8.7.: Lateral velocity and acceleration.

Further insight is provided by Figure 8.2, Figure 8.3, and Figure 8.4, showing the trajectories of all error quantities separately. Starting from different initial states, all trajectories converge not to the origin, but to a close vicinity of what is assumed to be a steady-state trajectory according to Definition 7.2. The artificial control inputs $\tilde{\mu}$, being related to the tangential and normal error quantities through the feedback matrix K, exhibit a corresponding behavior as shown in Figure 8.5, whereas the resulting actuator inputs s_X and δ are given in Figure 8.9. The trajectories of the actual vehicle states corresponding to the error and reference trajectories are shown in Figure 8.6, Figure 8.7, and Figure 8.8, emphasizing that no constraints as imposed by the set $\mathcal{C}_{v\bar{a}}$ are violated. The resulting external accelerations a are shown in Fig. 8.10.

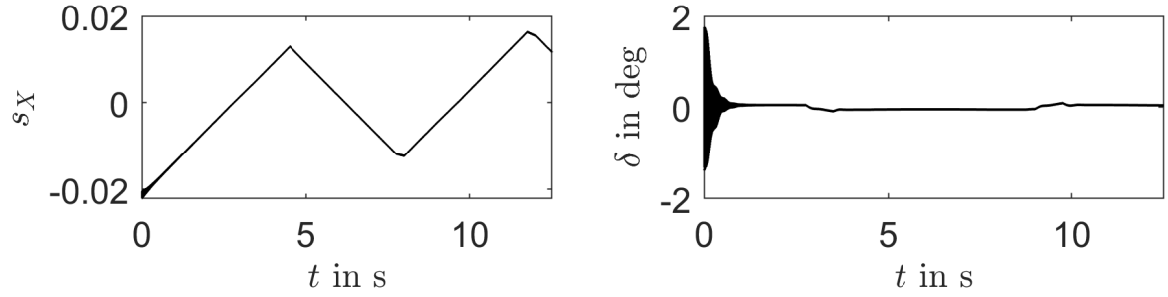
8.2. Discussion

While the theoretical results of Chapter 7 state conditions for the operational capability of the devised tracking controller, the simulation results from the previous section demonstrate the existence of parameter values that fulfill these conditions. The described approach has several significant benefits: Most importantly, it represents a feedback controller for which it is precisely known under which conditions constraint compliance is guaranteed. This not only enables safe tracking, but, based on a guaranteed upper bound



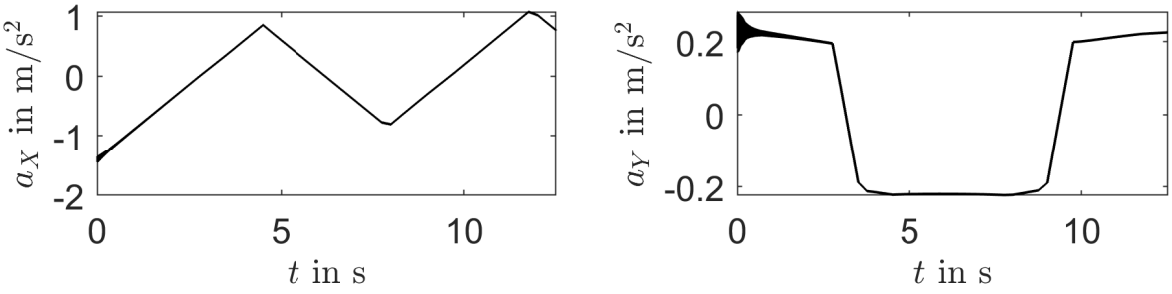
(a) Yaw rate ω. (b) Yaw acceleration $\dot{\omega}$.

Figure 8.8.: Vehicle orientation and yaw rate.



(a) Longitudinal tire slip s_x. (b) Front wheel angle δ.

Figure 8.9.: Actuator inputs.



(a) Longitudinal external acceleration a_X. (b) Lateral external acceleration a_Y.

Figure 8.10.: External accelerations.

on the tracking error, also problem-free interaction with the planning layer in Part I, which can provide sufficient safety margins when planning, such that no collisions occur.

Guarantees concerning constraint compliance typically require the employment of model-predictive controllers, which, however, are computationally demanding. The proposed controller, in contrast, is computationally very efficient, even in comparison to related approaches such as [148, 57]. These rely on almost the same procedure as described in Algorithm 5.1, however with one crucial difference in line 7: instead of finding the roots of a cubic polynomial, the system of nonlinear equations (5.7) must be solved for δ and s_X. Because no analytical solution is known, the use of numerical methods is required. Even for such a small number of equations, this is computationally more demanding than a closed-form expression control law and complicates control at low sampling times (such as the 10 ms used in the example). The proposed approach, in contrast, allows for short sampling times in digital implementations of the controller, such that analysis results from the continuous-time domain can hold and even high-frequency disturbances are counteracted. This advantage not only pertains to the actual implementation, but also to numerical simulation, which plays an important role during the design stage. There, the evaluation of simulation runs for many different initial states is drastically accelerated.

Another difference to approaches along the lines of [148, 57] lies in the fact that these typically assume that $\Delta a = 0$ in the position error dynamics (5.24). This ignores matters of robustness, but has the advantage to remove the influence of the yaw error states e_ψ and e_ω on the position error dynamics, which can then be analyzed independently of (5.32). The yaw error dynamics in turn becomes part of the zero dynamics of the error system, whose behavior must be carefully analyzed nonetheless in order to judge about the behavior of the vehicle during trajectory tracking. The non-explicit nature of the feedback law in approaches along the lines of [148, 57] severely complicates this analysis because the replacement of δ and s_X is prevented such that no closed-form expression for the closed-loop yaw error dynamics can be given. First attempts to analyze the stability of the zero dynamics [148] are limited to the case of a vehicle driving on a straight line with constant velocity. The approach proposed in this thesis, in contrast, is able to give guarantees for far more general driving scenarios.

Despite all advantages of the proposed method, the simulation results also indicate room for improvement. A major challenge arises from the fact that its outcome can be conservative, meaning that either no result can be found even if it exists, or only under relatively restrictive conditions (such as the comparably small value of $\Delta a_{\max}$ chosen in this simulation setting). This is caused by the fact that the procedure relies on over-approximation of parameter sets and only sufficient, but not necessary conditions for boundedness – an inherent feature of Lyapunov/LPV-based approaches in general, as already discussed in [124]. There, the LMI-conditions in Problem (7.1) and Problem (7.2) are emphasized to be a major source of conservatism, depending on the choice of the ϵ-parameters, whose values are not obvious to choose during the design phase. While this is inherent to the stated LMI-conditions, the over-approximation of parameter sets as detailed in Appendix F offers room for improvement.

Part III.

Conclusions and Future Work

9. Conclusions

This thesis has addressed the topic of consistency in hierarchical planning architectures for use in cooperative autonomous driving. An example for such an architecture has been proposed, consisting of three layers. Besides hierarchical consistency, the design of these layers has aimed at computational efficiency as outlined in Part I. To that end, several requirements had to be considered:

In order to enable fast selection of a maneuver, many different options must be checked quickly for their feasibility. For a chosen maneuver, a plan must then be obtained with as little delay as possible, because its admissibility can only be guaranteed for the current situation, while unforeseen changes could make the schedule obsolete. During planning, safety margins must be provided in order to allow for minor tracking errors; bounds on these errors must be derived during the design of the tracking controller, which in turn requires the reference trajectories' parameters to lie in a pre-defined set.

Several methods have been employed for fulfillment of these requirements, with the aim to demonstrate the potential of the overall concept. The encoding of the planning problem as optimization problem within the maneuver concept enables computationally efficient planning and constraint compliance, while computation of polytopic controllable sets allows for feasibility assessment and, based on approximations, for a further increase in computational efficiency. The tracking task was approached by extending an existing controller design based on input-output linearization, additionally considering state and input constraints and proving boundedness of the resulting tracking error.

These choices are certainly not the only possible realizations of the concept and far from perfect. Yet, when evaluating, one must differentiate between the hierarchical framework and the different elements for its realization: the framework itself clearly has the disadvantage to be inflexible in so far as that it completely relies on the maneuvers specified prior to deployment. The more maneuvers and corresponding controllable sets are designed, the higher both the design effort and the memory requirements. On the other hand, as can be concluded from both the theoretical results and the simulation studies, the hierarchical approach is actually able to fulfill the requirements with respect to consistency and computational efficiency, such that it seems a very promising subject for future investigations.

Concerning the realization of the framework, several disadvantages must be noted: first, the exact computation of controllable sets is both computationally demanding and, especially when dealing with polytopic set representations, numerically challenging, which makes the design process difficult. While approximations mitigate these issues, they are unlikely to be ever overcome completely, as the problem itself is subject to the curse of dimensionality. However, planning without controllable sets seems even less attractive, as it implies that for a given situation, online planning must be carried out without any

knowledge of the potential outcome. This makes situations possible in which neither a feasible plan can be found for a chosen parameter set of the planning problem, nor a new attempt to planning can be made, as the first search took too long. This is even more critical when assessing the feasibility of many different options – two problems which are resolved by the use of controllable sets, such that these should be a main ingredient of any planning method despite the associated challenges.

The tracking controller requires not less design effort than the planning part; especially the computation of the admissible sets using symbolic quantifier elimination and the analysis of the yaw error dynamics is intricate and requires careful choice of parameters. The chosen approach to the analysis task is complicated by the fact that it relies on only sufficient conditions, such that obtained results are very likely to be conservative. Also, obtaining such a result is difficult, as it relies on the appropriate choice of design parameters which have no clear physical meaning and are beyond a designer's intuition. This, however, is a problem in all approaches somehow related to the design of a Lyapunov function and even a conservative result must be regarded a success, considering the difficulty of the problem to compute an invariant set for a highly nonlinear, time-varying, and disturbed system. A thinkable alternative approach based on numerical computation of reachable sets would likewise produce conservative results, while the handling of this problem class would pose a conceptual challenge. Apart from the proof of boundedness of the yaw error dynamics, the proposed tracking controller represents a consequent improvement of the existing approaches in various directions: it is computationally more efficient than its predecessors as it does not anymore require solution of a system of nonlinear equations and it is able to account for constraints. Also, a first attempt to consider uncertainty has been made. Despite these advances, room for improvement remains, which is outlined in the following chapter.

10. Future Work

With the focus of this thesis being on the development of a methodology and numerical simulations for validation purposes, an obvious extension of the presented work is its implementation and testing on an experimental vehicle. Besides that, regarding the methodology, various directions of future research exist:

First, being the major bottleneck in the realization of the maneuver concept, it is desirable to develop more efficient methods for the computation of controllable sets. However, as this task is subject to the *curse of dimensionality*, there is not much hope for a conceptually simple solution and even an increase in computational capabilities would not lead to a major improvement. Rather, approximation-based methods should be developed further – for example by obtaining safety-guarantees for approximate dynamic programming algorithms. Also, decomposition approaches could help to reduce the dimensionality of the relevant state spaces. Such approaches could be combined with real-world data, for example to specify different temporal evolutions of a maneuver based on human-driven vehicles. Unlike in reinforcement learning, these data would not be directly used to train a policy, but to identify promising parameters for the devised framework. This approach could also help to address the problem of *coverage*, i.e., the problem of providing the maneuver library with a sufficient amount of maneuvers with sufficiently large entry sets. As such questions are difficult to answer solely based on the intuition of the system designer, real-world data could prove beneficial.

Another aspect, which was only touched upon in this thesis, is the design of strategies for the scheduling algorithm, e.g. optimizing traffic flow or a weighted sum of each involved vehicle's individual cost function. The effectiveness of different strategies could be evaluated in a large-scale traffic simulator. This should also account for heterogeneous vehicle dynamics, for example computing maneuvers for classes of vehicles in order to limit the complexity of the resulting maneuver libraries.

As far as the tracking controller from Part II is concerned, several extensions are conceivable: so far, perfect knowledge of the states of the error dynamics has been assumed. In practical deployment of the controller, this assumption will not hold; rather, the required information must be obtained from sensor readings by use of observers. These introduce measurement and estimation errors, which must be accounted for in the computation of the bound on the tracking error. Practical considerations necessitate to make the controller robust not only against uncertainty in the tire characteristics, but also against external disturbances such as wind gusts or varying road surface conditions or road incline. Larger uncertainty typically increases the conservatism of results, as these must be obtained for all possible realizations of a disturbance. The employed methods must counteract this mechanism in order to still produce meaningful results, which could, for example, be facilitated

by use of parameter-dependent Lyapunov functions.

Finally, the thesis has made use of polytopic inner approximations of non-convex sets several times, which were obtained manually. The development of a structured, automated approach to this problem would prove beneficial. Some of the considered non-convex sets resulted from application of algorithms for symbolic quantifier elimination, which – even though only applicable to small-scale problems – deserve more attention in the control community.

Appendix

A. Invariant Set Computation for Longitudinal Collision Avoidance

This section provides the proof for Proposition 2.2:

Proof. The objective is to compute for each initial longitudinal velocity the minimum time for a vehicle to come to standstill. If this is possible without collisions, then it is possible for all other initial velocities and corresponding input trajectories, as these lead to longer braking times. Considering the longitudinal position and velocity dynamics of each vehicle $i \in \{\mathrm{L}, \mathrm{F}\}$ in continuous time:

$$\begin{aligned} d^{(\mathrm{LF})} &= p_x^{(\mathrm{L})} - p_x^{(\mathrm{F})} \\ \dot{p}_x^{(\mathrm{L})} &= v_x^{(\mathrm{L})} & \dot{v}_x^{(\mathrm{L})} &= u_x^{(\mathrm{L})} \\ \dot{p}_x^{(\mathrm{F})} &= v_x^{(\mathrm{F})} & \dot{v}_x^{(\mathrm{F})} &= u_x^{(\mathrm{F})}, \end{aligned}$$

the minimum times result for maximum braking, i.e. $u_x^{(\mathrm{L})} = u_x^{(\mathrm{F})} = -u_{x,\max}$, such that Vehicle L can come to stand-still in:

$$t_{\mathrm{stp}}^{(\mathrm{L})} = \frac{v_x^{(\mathrm{L})}(t_0)}{|u_{x,\min}|}$$

and Vehicle F in:

$$t_{\mathrm{stp}}^{(\mathrm{F})} = \frac{v_x^{(\mathrm{F})}(t_0)}{|u_{x,\min}|}.$$

This gives the position trajectories:

$$p_x^{(\mathrm{L})}(t) = \begin{cases} p_x^{(\mathrm{L})}(t_0) + v_x^{(\mathrm{L})}(t_0) \cdot (t - t_0) + \frac{1}{2} u_{x,\min}(t - t_0)^2 & t < t_{\mathrm{stp}}^{(\mathrm{L})} \\ p_x^{(\mathrm{L})}(t_{\mathrm{stp}}^{(\mathrm{L})}) & t \geq t_{\mathrm{stp}}^{(\mathrm{L})} \end{cases}$$

and

$$p_x^{(\mathrm{F})}(t) = \begin{cases} p_x^{(\mathrm{F})}(t_0) + v_x^{(\mathrm{F})}(t_0) \cdot (t - t_0) + \frac{1}{2} u_{x,\min}(t - t_0)^2 & t < t_{\mathrm{stp}}^{(\mathrm{F})} \\ p_x^{(\mathrm{F})}(t_{\mathrm{stp}}^{(\mathrm{F})}) & t \geq t_{\mathrm{stp}}^{(\mathrm{F})} \end{cases}$$

Letting $t_0 = 0$ and assuming a collision-free initial state, i.e., $p_x^{(\mathrm{L})}(0) - p_x^{(\mathrm{F})}(0) \geq l_{x,\mathrm{safe}}$, a velocity-dependent safety distance is only required if $v_x^{(\mathrm{F})} \geq v_x^{(\mathrm{L})}$ because then $t_{\mathrm{stp}}^{(\mathrm{F})} \leq t_{\mathrm{stp}}^{(\mathrm{L})}$, such that $d^{(\mathrm{LF})} \geq l_{x,\mathrm{safe}}\ \forall t \geq 0$. Under these assumptions,

$$d^{(\mathrm{LF})}(t) = \begin{cases} p_x^{(\mathrm{L})}(0) - p_x^{(\mathrm{F})}(0) + \left(v_x^{(\mathrm{L})}(0) - v_x^{(\mathrm{F})}(0)\right) \cdot t & 0 \leq t < t_{\mathrm{stp}}^{(\mathrm{L})} \\ p_x^{(\mathrm{L})}\left(t_{\mathrm{stp}}^{(\mathrm{L})}\right) - p_x^{(\mathrm{F})}\left(t_{\mathrm{stp}}^{(\mathrm{L})}\right) - v_x^{(\mathrm{F})}\left(t_{\mathrm{stp}}^{(\mathrm{L})}\right)\left(t - t_{\mathrm{stp}}^{(\mathrm{L})}\right) - \frac{1}{2}u_{x,\min}\left(t - t_{\mathrm{stp}}^{(\mathrm{L})}\right)^2 & t_{\mathrm{stp}}^{(\mathrm{L})} \leq t < t_{\mathrm{stp}}^{(\mathrm{F})} \\ p_x^{(\mathrm{L})}\left(t_{\mathrm{stp}}^{(\mathrm{L})}\right) - p_x^{(\mathrm{F})}\left(t_{\mathrm{stp}}^{(\mathrm{F})}\right) & t_{\mathrm{stp}}^{(\mathrm{F})} \leq t \end{cases}$$

with the positions at stand-still:

$$\begin{aligned} p_x^{(\mathrm{L})}\left(t_{\mathrm{stp}}^{(\mathrm{L})}\right) &= p_x^{(\mathrm{L})}(0) + v_x^{(\mathrm{L})}(0) \cdot t_{\mathrm{stp}}^{(\mathrm{L})} + \frac{1}{2}u_{x,\min}\left(t_{\mathrm{stp}}^{(\mathrm{L})}\right)^2 \\ &= p_x^{(\mathrm{L})}(0) + v_x^{(\mathrm{L})}(0)\frac{v_x^{(\mathrm{L})}(0)}{|u_{x,\min}|} - \frac{1}{2}|u_{x,\min}|\left(\frac{v_x^{(\mathrm{L})}(0)}{|u_{x,\min}|}\right)^2 \\ &= p_x^{(\mathrm{L})}(0) + \frac{v_x^{(\mathrm{L})}(0)^2}{|u_{x,\min}|} - \frac{v_x^{(\mathrm{L})}(0)^2}{2|u_{x,\min}|} \\ &= p_x^{(\mathrm{L})}(0) + \frac{v_x^{(\mathrm{L})}(0)^2}{2|u_{x,\min}|} \end{aligned}$$

and

$$\begin{aligned} p_x^{(\mathrm{F})}\left(t_{\mathrm{stp}}^{(\mathrm{F})}\right) &= p_x^{(\mathrm{F})}(0) + v_x^{(\mathrm{F})}(0) \cdot t_{\mathrm{stp}}^{(\mathrm{F})} + \frac{1}{2}u_{x,\min}\left(t_{\mathrm{stp}}^{(\mathrm{F})}\right)^2 \\ &= p_x^{(\mathrm{F})}(0) + \frac{v_x^{(\mathrm{F})}(0)^2}{2|u_{x,\min}|}, \end{aligned}$$

such that the relative distance at stand-still reads:

$$d^{(\mathrm{LF})}\left(t_{\mathrm{stp}}^{(\mathrm{F})}\right) = p_x^{(\mathrm{L})}(0) - p_x^{(\mathrm{F})}(0) + \frac{1}{2|u_{x,\min}|}\left(v_x^{(\mathrm{L})}(0)^2 - v_x^{(\mathrm{F})}(0)^2\right).$$

As $d^{(\mathrm{LF})}(t)$ is a strictly monotonically decreasing function, the constraint $d^{(\mathrm{LF})} \geq l_{x,\mathrm{safe}}$ holds at any time if it holds at the minimum, i.e., at stand-still. This determines the set in Proposition 2.2.

□

B. Transformation of Reference Trajectory Representations

This section summarizes the relation between a reference trajectory in cartesian coordinates with state vector (5.1) and a reference trajectory in a Frenet frame as represented by the

state vector (5.2). From Fig. 5.1, it is clear that:

$$\begin{bmatrix} \dot{s} \\ \dot{n} \end{bmatrix} = R(\psi - \theta) \begin{bmatrix} v_X \\ v_Y \end{bmatrix} - \begin{bmatrix} n\dot{\theta} \\ 0 \end{bmatrix}. \tag{A.1}$$

Cartesian Frame $\rightarrow$ Frenet Frame

A reference trajectory $\bar{x}(\cdot)$ in the cartesian frame is interpreted as motion of a point with $n(\cdot) = 0$ and $\psi(\cdot) = 0$, such that $v_X = \dot{\bar{p}}_x$ and $v_Y = \dot{\bar{p}}_y$. Then, (A.1) becomes

$$\begin{bmatrix} \dot{s} \\ 0 \end{bmatrix} = R(-\theta) \begin{bmatrix} \dot{\bar{p}}_x \\ \dot{\bar{p}}_y \end{bmatrix}. \tag{A.2}$$

The velocity vector is, per definition, tangential to the reference path in the Frenet frame, such that (A.2) is compatible with:

$$\dot{s} = \left\| \begin{bmatrix} \dot{\bar{p}}_x \\ \dot{\bar{p}}_y \end{bmatrix} \right\| = \sqrt{\dot{\bar{p}}_x^2 + \dot{\bar{p}}_y^2}$$

and

$$\theta = \arctan2\left(\dot{\bar{p}}_x, \dot{\bar{p}}_y\right).$$

Assuming sufficient continuity of the trajectories, the tangential acceleration $\ddot{s}$ reads:

$$\ddot{s} = \frac{\mathrm{d}\dot{s}}{\mathrm{d}t} = \frac{\partial \dot{s}}{\partial \dot{\bar{p}}_x}\frac{\partial \dot{\bar{p}}_x}{\mathrm{d}t} + \frac{\partial \dot{s}}{\partial \dot{\bar{p}}_y}\frac{\partial \dot{\bar{p}}_y}{\mathrm{d}t} = \begin{bmatrix} \frac{\partial \dot{s}}{\partial \dot{\bar{p}}_x} & \frac{\partial \dot{s}}{\partial \dot{\bar{p}}_y} \end{bmatrix} \begin{bmatrix} \ddot{\bar{p}}_x \\ \ddot{\bar{p}}_y \end{bmatrix} = \frac{\begin{bmatrix} \dot{\bar{p}}_x & \dot{\bar{p}}_y \end{bmatrix}}{\sqrt{\dot{\bar{p}}_x^2 + \dot{\bar{p}}_y^2}} \begin{bmatrix} \ddot{\bar{p}}_x \\ \ddot{\bar{p}}_y \end{bmatrix} = \frac{\begin{bmatrix} \dot{\bar{p}}_x & \dot{\bar{p}}_y \end{bmatrix}}{\dot{s}} \begin{bmatrix} \ddot{\bar{p}}_x \\ \ddot{\bar{p}}_y \end{bmatrix}. \tag{A.3}$$

Also, it holds that:

$$\begin{bmatrix} \ddot{s} \\ 0 \end{bmatrix} = R(\theta)^{\mathsf{T}} \begin{bmatrix} \dddot{\bar{p}}_x \\ \dddot{\bar{p}}_y \end{bmatrix} - \begin{bmatrix} -\dot{s}\dot{\theta}^2 \\ 2\ddot{s}\dot{\theta} + \dot{s}\ddot{\theta} \end{bmatrix}$$

(this expression can be obtained either by differentiation of (A.3), or – and much simpler – by rearranging (A.4)). Also, using that $\frac{\mathrm{d}}{\mathrm{d}x}\arctan(x) = \frac{1}{1+x^2}$, where $x = \frac{\dot{\bar{p}}_y}{\dot{\bar{p}}_x}$:

$$\begin{aligned} \dot{\theta} = \frac{\mathrm{d}\theta}{\mathrm{d}t} &= \frac{\dot{\bar{p}}_x^2}{\dot{\bar{p}}_x^2 + \dot{\bar{p}}_y}^2 \cdot \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\dot{\bar{p}}_y}{\dot{\bar{p}}_x}\right) \\ &= \frac{1}{\dot{\bar{p}}_x^2 + \dot{\bar{p}}_y^2} \begin{bmatrix} -\dot{\bar{p}}_y & \dot{\bar{p}}_x \end{bmatrix} \begin{bmatrix} \ddot{\bar{p}}_x \\ \ddot{\bar{p}}_y \end{bmatrix} = \frac{1}{\dot{s}^2} \begin{bmatrix} \dot{\bar{p}}_x & \dot{\bar{p}}_y \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{\bar{p}}_x \\ \ddot{\bar{p}}_y \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} &\ddot{\theta} = \\ &\frac{1}{\dot{s}^2}\left(\begin{bmatrix} \ddot{\bar{p}}_x & \ddot{\bar{p}}_y \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{\bar{p}}_x \\ \ddot{\bar{p}}_y \end{bmatrix} + \begin{bmatrix} \dot{\bar{p}}_x & \dot{\bar{p}}_y \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \dddot{\bar{p}}_x \\ \dddot{\bar{p}}_y \end{bmatrix}\right) - \frac{2\ddot{s}}{\dot{s}^3}\left(\begin{bmatrix} \dot{\bar{p}}_x & \dot{\bar{p}}_y \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{\bar{p}}_x \\ \ddot{\bar{p}}_y \end{bmatrix}\right) \\ &= \frac{1}{\dot{s}^2}\left(\begin{bmatrix} \ddot{\bar{p}}_x & \ddot{\bar{p}}_y \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{\bar{p}}_x \\ \ddot{\bar{p}}_y \end{bmatrix} + \begin{bmatrix} \dot{\bar{p}}_x & \dot{\bar{p}}_y \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \dddot{\bar{p}}_x \\ \dddot{\bar{p}}_y \end{bmatrix}\right) - \frac{2\ddot{s}\dot{\theta}}{\dot{s}} \end{aligned}$$

Frenet Frame → Cartesian Frame

If a reference trajectory $\bar{x}_{\text{curv}}$ in the Frenet frame is given, it can be related to a trajectory in the cartesian frame by rearranging (A.2) and subsequent differentiation as follows:

$$\begin{bmatrix} \dot{\bar{p}}_x \\ \dot{p}_y \end{bmatrix} = \dot{s} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} = R(\theta) \begin{bmatrix} \dot{s} \\ 0 \end{bmatrix},$$

$$\begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \end{bmatrix} = \ddot{s} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} + \dot{s}\dot{\theta} \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} = R(\theta) \begin{bmatrix} \ddot{s} \\ \dot{s}\dot{\theta} \end{bmatrix},$$

$$\begin{bmatrix} \dddot{p}_x \\ \dddot{p}_y \end{bmatrix} = \dddot{s} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} + 2\ddot{s}\dot{\theta} \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} + \dot{s} \left(\ddot{\theta} \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} - \dot{\theta}^2 \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \right) = R(\theta) \begin{bmatrix} \dddot{s} - \dot{s}\dot{\theta}^2 \\ 2\ddot{s}\dot{\theta} + \dot{s}\ddot{\theta} \end{bmatrix} \tag{A.4}$$

C. Uncertainty in the Tire Model

In this section, bounds on the uncertainty terms of Section 5.3, Δa_{R}, Δa_c, and, combining these, Δa, are derived. Introducing the tire uncertainty (5.11) into (5.12) leads to:

$$
\begin{aligned}
a_{\text{tire}} = &\begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & -\frac{l_{\mathrm{r}}}{\Theta} \end{bmatrix} \begin{bmatrix} (1-\gamma)(\bar{c}_X + \Delta c_X) s_X \\ (\bar{c}_Y + \Delta c_Y) \arctan\left(\frac{l_{\mathrm{r}}\omega - v_Y}{v_X}\right) \end{bmatrix} + \dots \\
&\begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & \frac{l_{\mathrm{f}}}{\Theta} \end{bmatrix} R(\delta) \begin{bmatrix} \gamma(\bar{c}_X + \Delta c_X) s_X \\ (\bar{c}_Y + \Delta c_Y) \left(\delta - \arctan\left(\frac{l_{\mathrm{f}}\omega + v_Y}{v_X}\right)\right) \end{bmatrix} \\
= &\left(\begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & -\frac{l_{\mathrm{r}}}{\Theta} \end{bmatrix} \begin{bmatrix} (1-\gamma) s_X & 0 \\ 0 & \arctan\left(\frac{l_{\mathrm{r}}\omega - v_Y}{v_X}\right) \end{bmatrix} + \dots \right. \\
&\left. \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & \frac{l_{\mathrm{f}}}{\Theta} \end{bmatrix} R(\delta) \begin{bmatrix} \gamma s_X & 0 \\ 0 & \delta - \arctan\left(\frac{l_{\mathrm{r}}\omega - v_Y}{v_X}\right) \end{bmatrix} \right) \begin{bmatrix} \bar{c}_X + \Delta c_X \\ \bar{c}_Y + \Delta c_Y \end{bmatrix}
\end{aligned}
$$

Further derivations require polynomial expressions. Thus, trigonometric terms are replaced by their series expansion:

$$
\begin{aligned}
\sin(x) &= \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} = P_1(x) + R_1(x) \\
\cos(x) &= \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!} = P_2(x) + R_2(x) \\
\arctan(x) &= \sum_{k=0}^{\infty} (-1)^k \frac{x^{(2k+1)}}{2k+1} = P_3(x) + R_3(x)
\end{aligned}
$$

in all terms which do not depend on Δc_X or Δc_Y, with polynomials P_i, $i \in \{1, 2, 3\}$, and corresponding remainders R_i. This gives:

$$
\begin{aligned}
a_{\text{tire}} = &\left(\begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & -\frac{l_{\mathrm{r}}}{\Theta} \end{bmatrix} \begin{bmatrix} (1-\gamma) s_X & 0 \\ 0 & P_3\left(\frac{l_{\mathrm{r}}\omega - v_Y}{v_X}\right) + R_3\left(\frac{l_{\mathrm{r}}\omega - v_Y}{v_X}\right) \end{bmatrix} + \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & \frac{l_{\mathrm{f}}}{\Theta} \end{bmatrix} \cdot \dots \right. \\
&\left. \begin{bmatrix} P_2(\delta) + R_2(\delta) & -P_1(\delta) - R_1(\delta) \\ P_1(\delta) + R_1(\delta) & P_2(\delta) + R_2(\delta) \end{bmatrix} \begin{bmatrix} \gamma s_X & 0 \\ 0 & \delta - P_3\left(\frac{l_{\mathrm{f}}\omega + v_Y}{v_X}\right) - R_3\left(\frac{l_{\mathrm{f}}\omega + v_Y}{v_X}\right) \end{bmatrix} \right) \begin{bmatrix} \bar{c}_X \\ \bar{c}_Y \end{bmatrix} + \dots \\
&\left(\begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & -\frac{l_{\mathrm{r}}}{\Theta} \end{bmatrix} \begin{bmatrix} (1-\gamma) s_X & 0 \\ 0 & \arctan\frac{l_{\mathrm{r}}\omega - v_Y}{v_X} \end{bmatrix} + \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & \frac{l_{\mathrm{f}}}{\Theta} \end{bmatrix} R(\delta) \begin{bmatrix} \gamma s_X & 0 \\ 0 & \delta - \arctan\frac{l_{\mathrm{f}}\omega + v_Y}{v_X} \end{bmatrix} \right) \begin{bmatrix} \Delta c_X \\ \Delta c_Y \end{bmatrix}.
\end{aligned}
$$

Then, a_{tire} is partitioned according to $a_{\text{tire}} = \bar{a} + \Delta a_c + \Delta a_{\text{R}}$, with:

$$\bar{a} := \left(\begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & -\frac{l_{\text{r}}}{\Theta} \end{bmatrix} \begin{bmatrix} (1-\gamma)s_X & 0 \\ 0 & P_3\left(\frac{l_{\text{r}}\omega - v_Y}{v_X}\right) \end{bmatrix} + \dots \right.$$
$$\left. \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & \frac{l_{\text{f}}}{\Theta} \end{bmatrix} \begin{bmatrix} P_2(\delta) & -P_1(\delta) \\ P_1(\delta) & P_2(\delta) \end{bmatrix} \begin{bmatrix} \gamma s_X & 0 \\ 0 & \delta - P_3\left(\frac{l_{\text{f}}\omega + v_Y}{v_X}\right) \end{bmatrix} \right) \begin{bmatrix} \bar{c}_X \\ \bar{c}_Y \end{bmatrix},$$

$$\Delta a_c := \left(\begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & -\frac{l_{\text{r}}}{\Theta} \end{bmatrix} \begin{bmatrix} (1-\gamma) & 0 \\ 0 & \arctan\frac{l_{\text{r}}\omega - v_Y}{v_X} \end{bmatrix} + \dots \right.$$
$$\left. \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & \frac{l_{\text{f}}}{\Theta} \end{bmatrix} R(\delta) \begin{bmatrix} \gamma & 0 \\ 0 & \delta - \arctan\frac{l_{\text{f}}\omega + v_Y}{v_X} \end{bmatrix} \right) \begin{bmatrix} \Delta c_X s_X \\ \Delta c_Y \end{bmatrix}, \tag{A.5}$$

and

$$\Delta a_{\text{R}} := \left(\begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & -\frac{l_{\text{r}}}{\Theta} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & R_3\left(\frac{l_{\text{r}}\omega - v_Y}{v_X}\right) \end{bmatrix} + \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & \frac{l_{\text{f}}}{\Theta} \end{bmatrix} R(\delta) \begin{bmatrix} 0 & 0 \\ 0 & -R_3\left(\frac{l_{\text{f}}\omega + v_Y}{v_X}\right) \end{bmatrix} \right.$$
$$\left. + \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & \frac{l_{\text{f}}}{\Theta} \end{bmatrix} \begin{bmatrix} R_2(\delta) & -R_1(\delta) \\ R_1(\delta) & R_2(\delta) \end{bmatrix} \begin{bmatrix} \gamma s_X & 0 \\ 0 & \delta - P_3\left(\frac{l_{\text{f}}\omega + v_Y}{v_X}\right) \end{bmatrix} \right) \begin{bmatrix} \bar{c}_X \\ \bar{c}_Y \end{bmatrix}.$$

In a first-order series expansion of the trigonometric terms, $R_1(x) = \sin(x) - x$, $R_2(x) = \cos(x) - 1$, and $R_3(x) = \arctan(x) - x$. Then, the nominal external accelerations (5.26) result. While (A.5) is not affected by this, Δa_{R} becomes:

$$\Delta a_{\text{R}} = \left(\begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & -\frac{l_{\text{r}}}{\Theta} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \arctan\left(\frac{l_{\text{r}}\omega - v_Y}{v_X}\right) - \frac{l_{\text{r}}\omega - v_Y}{v_X} \end{bmatrix} + \dots \right.$$
$$\begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & \frac{l_{\text{f}}}{\Theta} \end{bmatrix} R(\delta) \begin{bmatrix} 0 & 0 \\ 0 & -\arctan\left(\frac{l_{\text{f}}\omega + v_Y}{v_X}\right) + \frac{l_{\text{f}}\omega + v_Y}{v_X} \end{bmatrix}$$
$$\left. + \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \\ 0 & \frac{l_{\text{f}}}{\Theta} \end{bmatrix} \begin{bmatrix} \cos(\delta) - 1 & -\sin(\delta) + \delta \\ \sin(\delta) - \delta & \cos(\delta) - 1 \end{bmatrix} \begin{bmatrix} \gamma s_X & 0 \\ 0 & \delta - \frac{l_{\text{f}}\omega + v_Y}{v_X} \end{bmatrix} \right) \begin{bmatrix} \bar{c}_X \\ \bar{c}_Y \end{bmatrix}. \tag{A.6}$$

The following proposition holds for bounds on Δa_{R}:

Proposition A.1 (Bounds on Δa_{R})**.** *For given vehicle parameters l_r, l_f, m, Θ, $\bar{c}_X$, $\bar{c}_Y$, bounds:*

$$v_{X,\min} \leq v_X \leq v_{X,\max},\ |v_Y| \leq v_{Y,\max},\ |\omega| \leq \omega_{\max},\ |s_X| \leq s_{X,\max},\ |\delta| \leq \delta_{\max},$$

and with

$$M_1 := \max_{\left|\frac{l_r\omega - v_Y}{v_X}\right| \leq \frac{l_r\omega_{\max} - v_{Y,\min}}{v_{X,\min}}} \left|\left| \arctan\left(\frac{l_r\omega - v_Y}{v_X}\right) - \frac{l_r\omega - v_Y}{v_X} \right|\right|,$$

$$M_2 := \max_{\left|\frac{l_f\omega + v_Y}{v_X}\right| \leq \frac{l_f\omega_{\max} + v_{Y,\max}}{v_{X,\min}}} \left|\left| \arctan\left(\frac{l_f\omega + v_Y}{v_X}\right) - \frac{l_f\omega + v_Y}{v_X} \right|\right|,$$

$$M_3 := \max_{|\delta| \leq \delta_{\max}} \left|\left| \begin{bmatrix} \cos(\delta) - 1 & -\sin(\delta) + \delta \\ \sin(\delta) - \delta & \cos(\delta) - 1 \end{bmatrix} \right|\right| \leq \max_{|\delta| \leq \delta_{\max}} \sqrt{(\delta - \sin(\delta))^2 + (\cos(\delta) - 1)^2},$$

and, with $b := -\frac{l_f\omega + v_Y}{v_X}$,

$$M_4 := \max \left(\begin{bmatrix} s_X & \delta & b \end{bmatrix} \begin{bmatrix} (\gamma s_X)^2 & 0 & 0 \\ 0 & \bar{c}_Y^2 & \bar{c}_Y^2 \\ 0 & \bar{c}_Y^2 & \bar{c}_Y^2 \end{bmatrix} \begin{bmatrix} s_X \\ \delta \\ b \end{bmatrix} \right)^{\frac{1}{2}} \quad s.t. \quad \begin{array}{l} |s_X| \leq s_{X,\max}, \\ |b| \leq \frac{l_f\omega_{\max} + v_{y,\max}}{v_{x,\min}}, \\ |\delta| \leq \delta_{\max} \end{array}$$

it holds that:

$$\left|\left| \begin{bmatrix} \Delta a_{X,R} \\ \Delta a_{Y,R} \end{bmatrix} \right|\right| \leq \frac{\bar{c}_Y(M_1 + M_2) + M_3 \cdot M_4}{m}$$

and

$$||\Delta a_{\psi,R}|| \leq \frac{\bar{c}_Y(l_r M_1 + l_f M_2) + l_f M_3 \cdot M_4}{\Theta}.$$

Proof. Noting that $\begin{bmatrix} \Delta a_{X,\mathrm{R}} \\ \Delta a_{Y,\mathrm{R}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \Delta a_{\mathrm{R}}$ and $\Delta a_{\psi,\mathrm{R}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \Delta a_{\mathrm{R}}$, from (A.6),

it follows that:

$$
\begin{aligned}
m \left|\left| \begin{bmatrix} \Delta a_{X,\mathrm{R}} \\ \Delta a_{Y,\mathrm{R}} \end{bmatrix} \right|\right| &= \left|\left| \left(\begin{bmatrix} 0 & 0 \\ 0 & \arctan\left(\frac{l_\mathrm{r}\omega - v_Y}{v_X}\right) - \frac{l_\mathrm{r}\omega - v_Y}{v_X} \end{bmatrix} + R(\delta) \begin{bmatrix} 0 & 0 \\ 0 & -\arctan\left(\frac{l_\mathrm{f}\omega + v_Y}{v_X}\right) + \frac{l_\mathrm{f}\omega + v_Y}{v_X} \end{bmatrix} \right.\right. \\
&\quad \left.\left. + \begin{bmatrix} \cos(\delta) - 1 & -\sin(\delta) + \delta \\ \sin(\delta) - \delta & \cos(\delta) - 1 \end{bmatrix} \begin{bmatrix} \gamma s_X & 0 \\ 0 & \delta - \frac{l_\mathrm{f}\omega + v_Y}{v_X} \end{bmatrix} \right) \begin{bmatrix} \bar{c}_X \\ \bar{c}_Y \end{bmatrix} \right|\right| \\
&\leq \left|\left| \left(\begin{bmatrix} 0 & 0 \\ 0 & \arctan\left(\frac{l_\mathrm{r}\omega - v_Y}{v_X}\right) - \frac{l_\mathrm{r}\omega - v_Y}{v_X} \end{bmatrix} + R(\delta) \begin{bmatrix} 0 & 0 \\ 0 & -\arctan\left(\frac{l_\mathrm{f}\omega + v_Y}{v_X}\right) + \frac{l_\mathrm{f}\omega + v_Y}{v_X} \end{bmatrix} \right) \begin{bmatrix} \bar{c}_X \\ \bar{c}_Y \end{bmatrix} \right|\right| \\
&\quad + \left|\left| \begin{bmatrix} \cos(\delta) - 1 & -\sin(\delta) + \delta \\ \sin(\delta) - \delta & \cos(\delta) - 1 \end{bmatrix} \begin{bmatrix} \gamma s_X & 0 \\ 0 & \delta - \frac{l_\mathrm{f}\omega + v_Y}{v_X} \end{bmatrix} \begin{bmatrix} \bar{c}_X \\ \bar{c}_Y \end{bmatrix} \right|\right| \\
&= \bar{c}_Y \left|\left| \begin{bmatrix} 0 \\ \arctan\left(\frac{l_\mathrm{r}\omega - v_Y}{v_X}\right) - \frac{l_\mathrm{r}\omega - v_Y}{v_X} \end{bmatrix} + R(\delta) \begin{bmatrix} 0 \\ -\arctan\left(\frac{l_\mathrm{f}\omega + v_Y}{v_X}\right) + \frac{l_\mathrm{f}\omega + v_Y}{v_X} \end{bmatrix} \right|\right| \cdots \\
&\quad + \left|\left| \begin{bmatrix} \cos(\delta) - 1 & -\sin(\delta) + \delta \\ \sin(\delta) - \delta & \cos(\delta) - 1 \end{bmatrix} \begin{bmatrix} \gamma s_X & 0 \\ 0 & \delta - \frac{l_\mathrm{f}\omega + v_Y}{v_X} \end{bmatrix} \begin{bmatrix} \bar{c}_X \\ \bar{c}_Y \end{bmatrix} \right|\right| \\
&\leq \bar{c}_Y \left| \arctan\left(\frac{l_\mathrm{r}\omega - v_Y}{v_X}\right) - \frac{l_\mathrm{r}\omega - v_Y}{v_X} \right| + \bar{c}_Y \left| -\arctan\left(\frac{l_\mathrm{f}\omega + v_Y}{v_X}\right) + \frac{l_\mathrm{f}\omega + v_Y}{v_X} \right| \cdots \\
&\quad + \left|\left| \begin{bmatrix} \cos(\delta) - 1 & -\sin(\delta) + \delta \\ \sin(\delta) - \delta & \cos(\delta) - 1 \end{bmatrix} \right|\right| \cdot \left|\left| \begin{bmatrix} \gamma s_X & 0 \\ 0 & \delta - \frac{l_\mathrm{f}\omega + v_Y}{v_X} \end{bmatrix} \begin{bmatrix} \bar{c}_X \\ \bar{c}_Y \end{bmatrix} \right|\right| \\
&\leq \bar{c}_y(M_1 + M_2) + M_3 \cdot M_4,
\end{aligned}
$$

where

$$
\left|\left| \begin{bmatrix} \gamma s_X & 0 \\ 0 & \delta - \frac{l_\mathrm{f}\omega + v_Y}{v_X} \end{bmatrix} \begin{bmatrix} \bar{c}_X \\ \bar{c}_Y \end{bmatrix} \right|\right| =
$$

$$
\left|\left| \begin{bmatrix} \gamma s_X \bar{c}_X \\ \bar{c}_Y \left(\delta - \frac{l_\mathrm{f}\omega + v_Y}{v_X}\right) \end{bmatrix} \right|\right| = \left(\begin{bmatrix} s_X \\ \delta \\ b \end{bmatrix}^\mathsf{T} \begin{bmatrix} (\gamma s_X)^2 & 0 & 0 \\ 0 & \bar{c}_Y^2 & \bar{c}_Y^2 \\ 0 & \bar{c}_Y^2 & \bar{c}_Y^2 \end{bmatrix} \begin{bmatrix} s_X \\ \delta \\ b \end{bmatrix} \right)^{\frac{1}{2}}.
$$

Similarly, from (A.6),

$$
\begin{aligned}
&||\Delta a_{\psi,\mathrm{R}}|| = \\
&\left|\left| \left(-\frac{l_\mathrm{r}}{\Theta} \begin{bmatrix} 0 & \arctan\left(\frac{l_\mathrm{r}\omega - v_Y}{v_X}\right) - \frac{l_\mathrm{r}\omega - v_Y}{v_X} \end{bmatrix} + R(\delta) \begin{bmatrix} 0 & 0 \\ 0 & \frac{l_\mathrm{f}}{\Theta}\left(-\arctan\left(\frac{l_\mathrm{f}\omega + v_Y}{v_X}\right) + \frac{l_\mathrm{f}\omega + v_Y}{v_X}\right) \end{bmatrix} \cdots \right.\right. \\
&\left.\left. + \begin{bmatrix} 0 & \frac{l_\mathrm{f}}{\Theta} \end{bmatrix} \begin{bmatrix} \cos(\delta) - 1 & -\sin(\delta) + \delta \\ \sin(\delta) - \delta & \cos(\delta) - 1 \end{bmatrix} \begin{bmatrix} \gamma s_X & 0 \\ 0 & \delta - \frac{l_\mathrm{f}\omega + v_Y}{v_X} \end{bmatrix} \right) \begin{bmatrix} \bar{c}_X \\ \bar{c}_Y \end{bmatrix} \right|\right| \\
&\leq \frac{\bar{c}_Y(l_\mathrm{r}M_1 + l_\mathrm{f}M_2) + l_\mathrm{f}M_3 \cdot M_4}{\Theta}.
\end{aligned}
$$

Note that the computation of M_4 is a non-convex quadratic program, which, however, can be solved to global optimality by [63] or [152]. □

In a similar way, bounds on the tire parameter uncertainty can be obtained:

Proposition A.2 (Bounds on Δa_c)**.** *Given $\alpha_{\max}$ as in Appendix E, it holds that*

$$\left|\left|\begin{bmatrix}\Delta a_{X,c}\\ \Delta a_{Y,c}\end{bmatrix}\right|\right| \leq \frac{M_5 + M_6}{m} \left|\left|\begin{bmatrix}\Delta c_{X,\max}\\ \Delta c_{Y,\max}\end{bmatrix}\right|\right|$$

and

$$||\Delta a_{\psi,c}|| \leq \frac{l_r \alpha_{\max} + l_f M_6}{\Theta} \left|\left|\begin{bmatrix}\Delta c_{X,\max}\\ \Delta c_{Y,\max}\end{bmatrix}\right|\right|$$

where

$$\begin{aligned} M_5 :&= \max \left|\left|\begin{bmatrix}(1-\gamma)s_X & 0\\ 0 & \arctan\frac{l_r\omega - v_Y}{v_X}\end{bmatrix}\right|\right| \;\; s.t.\; |s_X| \leq s_{X,\max}, \left|\arctan\frac{l_r\omega - v_Y}{v_X}\right| \leq \alpha_{\max} \\ &= \max\left\{(1-\gamma)s_{X,\max}, \alpha_{\max}\right\} \end{aligned}$$

and

$$\begin{aligned} M_6 :&= \max \left|\left|\begin{bmatrix}\gamma s_X & 0\\ 0 & \delta - \arctan\frac{l_f\omega + v_Y}{v_X}\end{bmatrix}\right|\right| \;\; s.t.\; |s_X| \leq s_{X,\max}, \left|\delta - \arctan\frac{l_f\omega + v_Y}{v_X}\right| \leq \alpha_{\max} \\ &= \max\{\gamma s_{X,\max}, \alpha_{\max}\}. \end{aligned}$$

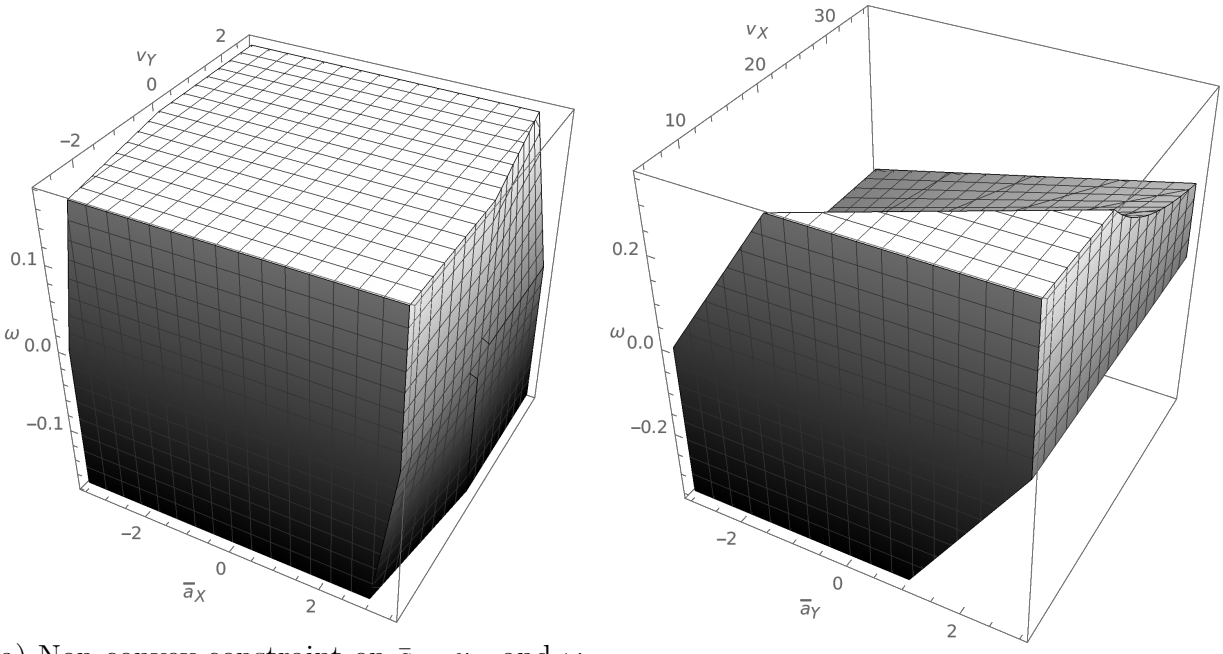
Proof. From (A.5), it follows that:

$$\begin{aligned} &\left|\left|\begin{bmatrix}\Delta a_{X,c}\\ \Delta a_{Y,c}\end{bmatrix}\right|\right| = \\ &\frac{1}{m}\left|\left|\left(\begin{bmatrix}(1-\gamma)s_X & 0\\ 0 & \arctan\frac{l_r\omega - v_Y}{v_X}\end{bmatrix} + R(\delta)\begin{bmatrix}\gamma s_X & 0\\ 0 & \delta - \arctan\frac{l_f\omega + v_Y}{v_X}\end{bmatrix}\right)\begin{bmatrix}\Delta c_X\\ \Delta c_Y\end{bmatrix}\right|\right| \\ &\leq \frac{M_5 + M_6}{m}\left|\left|\begin{bmatrix}\Delta c_{X,\max}\\ \Delta c_{Y,\max}\end{bmatrix}\right|\right| \end{aligned}$$

as well as:

$$\begin{aligned} ||\Delta a_{\psi,c}|| = &\left|\left|\left(\begin{bmatrix}0 & -\frac{l_r}{\Theta}\end{bmatrix}\begin{bmatrix}(1-\gamma)s_X & 0\\ 0 & \arctan\frac{l_r\omega - v_Y}{v_X}\end{bmatrix} + \dots \right.\right. \\ &\left.\left. \begin{bmatrix}0 & \frac{l_f}{\Theta}\end{bmatrix} R(\delta)\begin{bmatrix}\gamma s_X & 0\\ 0 & \delta - \arctan\frac{l_f\omega + v_Y}{v_X}\end{bmatrix}\right)\begin{bmatrix}\Delta c_X\\ \Delta c_Y\end{bmatrix}\right|\right| \\ &\leq \left(\frac{l_r}{\Theta}\left|\left|\arctan\frac{l_r\omega - v_Y}{v_X}\right|\right| + \frac{l_f}{\Theta}\left|\left|\begin{bmatrix}\gamma s_X & 0\\ 0 & \delta - \arctan\frac{l_f\omega + v_Y}{v_X}\end{bmatrix}\right|\right|\right)\left|\left|\begin{bmatrix}\Delta c_{X,\max}\\ \Delta c_{Y,\max}\end{bmatrix}\right|\right| \\ &\leq \frac{l_r\alpha_{\max} + l_f M_6}{\Theta}\left|\left|\begin{bmatrix}\Delta c_{X,\max}\\ \Delta c_{Y,\max}\end{bmatrix}\right|\right|. \end{aligned}$$

□



(a) Non-convex constraint on $\bar{a}_X$, v_y, and ω.

(b) Non-convex constraint on $\bar{a}_Y$, v_x, and ω.

Figure A.1.: Non-convex constraint sets $\tilde{\mathcal{C}}_{\bar{a}_X}$ (left) and $\tilde{\mathcal{C}}_{\bar{a}_Y}$ (right).

D. Polytopic acceleration constraints $\mathcal{A}$

In this section, the nonlinear acceleration constraints (6.4) and (6.5) are linearized. In doing so, it is important to obtain inner approximations of the admissible space in order to comply with the original, nonlinear constraints. At first, define:

$$\tilde{\mathcal{C}}_{\bar{a}_X} := \left\{ \begin{bmatrix} \bar{a}_X \\ v_Y \\ \omega \end{bmatrix} \mid |\bar{a}_X + v_Y\omega| \leq \bar{a}_{X,\max}, \exists v_X : v \in \mathcal{V} \right\}$$

and

$$\tilde{\mathcal{C}}_{\bar{a}_Y} := \left\{ \begin{bmatrix} \bar{a}_Y \\ v_X \\ \omega \end{bmatrix} \mid |\bar{a}_Y - v_X\omega| \leq \bar{a}_{Y,\max}, \exists v_Y : v \in \mathcal{V} \right\}$$

Plots of these sets are shown in Fig. A.1a and A.1b.

Linear bounds on $\bar{a}_X$

From Fig. A.1a, it is apparent that the set $\mathcal{C}_{\bar{a}_X}$ is mildly non-convex, in which the impact of the non-convex part varies with ω. This motivates to determine a set of constraints

which hold *for all* admissible values of ω:

$$\left\{ \begin{bmatrix} \bar{a}_X \\ v_y \end{bmatrix} \mid |\bar{a}_X + v_Y\omega| \leq \bar{a}_{X,\max} \forall \omega \in [-\omega_{\max}, \omega_{\max}] \right\}$$
$$\Leftrightarrow \left\{ \begin{bmatrix} \bar{a}_X \\ v_Y \end{bmatrix} \mid \max_{|\omega|\leq\omega_{\max}} \bar{a}_X + v_Y\omega \leq \bar{a}_{X,\max}, \quad \min_{|\omega|\leq\omega_{\max}} \bar{a}_X + v_Y\omega \geq -\bar{a}_{X,\max} \right\}.$$

While the elimination of the quantifier could be facilitated by using [152], an analytical solution is also possible. For fixed values of $\bar{a}_X$ and v_Y, it holds that:

Proposition A.3.

$$\max_{|\omega|\leq\omega_{\max}} \bar{a}_X + v_Y\omega = \begin{cases} \bar{a}_X - v_Y\omega_{\max} & v_Y < 0 \\ \bar{a}_X + v_Y\omega_{\max} & v_Y \geq 0 \end{cases}.$$

Proof. Definining a Lagrange function

$$L(\omega, \lambda_1, \lambda_2) := -(\bar{a}_X + v_Y\omega) + \lambda_1(\omega - \omega_{\max}) + \lambda_2(-\omega - \omega_{\max}),$$

the KKT-conditions [25] read:

$$\begin{aligned} \nabla_\omega L = -v_Y + \lambda_1 - \lambda_2 = 0 \\ \omega = \omega_{\max} \Rightarrow \lambda_2 = 0 \Rightarrow \lambda_1 = v_Y > 0 \\ \omega = -\omega_{\max} \Rightarrow \lambda_1 = 0 \Rightarrow \lambda_2 = -v_Y > 0 \end{aligned}$$

The considered problem of optimizing a linear function on a polytopic domain is convex, such that the KKT-conditions are necessary and sufficient, giving the result. □

Similarly, it holds that:

Proposition A.4.

$$\min_{\omega\in[\omega_{\min},\omega_{\max}]} \bar{a}_X + v_Y\omega = \begin{cases} \bar{a}_X + v_Y\omega_{\max} & v_Y < 0 \\ \bar{a}_X + v_Y\omega_{\min} & v_Y \geq 0 \end{cases}.$$

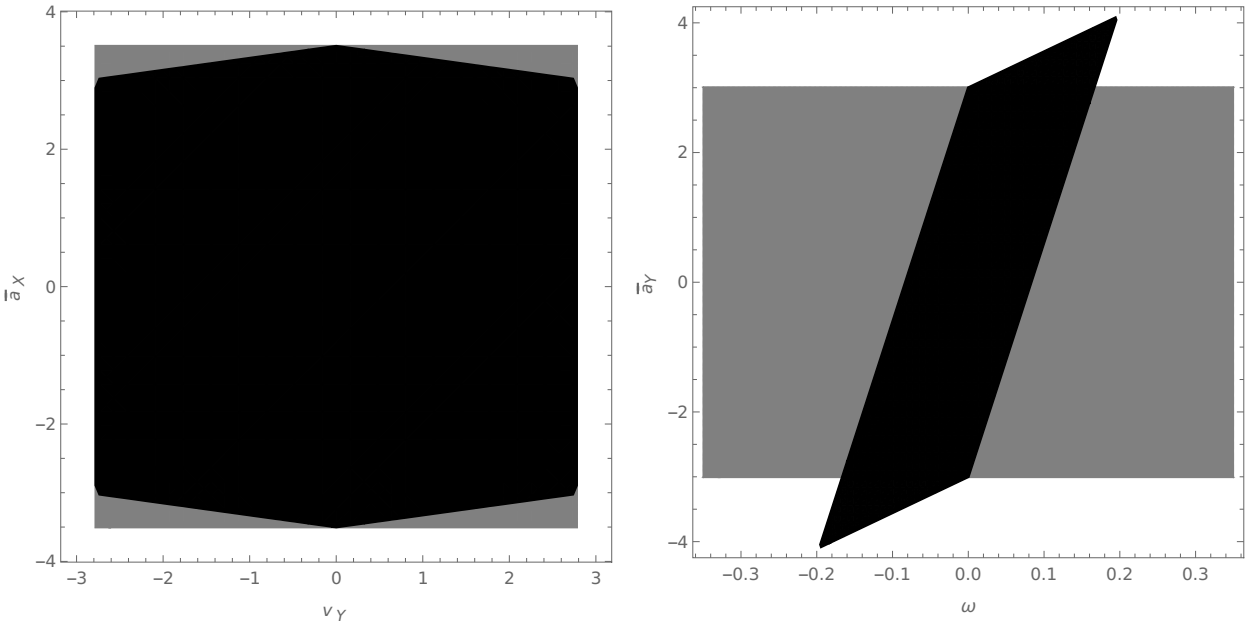
Proof. Again, define:

$$L(\omega, \lambda_1, \lambda_2) = \bar{a}_X + v_Y\omega + \lambda_1(\omega - \omega_{\max}) + \lambda_2(-\omega - \omega_{\max}),$$

with KKT-conditions:

$$\begin{aligned} \nabla_\omega L = v_Y + \lambda_1 - \lambda_2 = 0 \\ \omega = \omega_{\max} \Rightarrow \lambda_2 = 0 \Rightarrow \lambda_1 = -v_Y > 0 \\ \omega = -\omega_{\max} \Rightarrow \lambda_1 = 0 \Rightarrow \lambda_2 = v_Y > 0 \end{aligned}$$

□



(a) Resulting constraints (black) on $\bar{a}_X$ and v_Y when eliminating ω vs. box constraints on $\bar{a}_X$ (gray).

(b) Resulting constraints (black) on $\bar{a}_Y$ and v_Y when eliminating v_X vs. box constraints on $\bar{a}_Y$ (gray).

Figure A.2.: Resulting linearized constraint sets; parameters: $-10\,\mathrm{km\,h^{-1}} \leq v_Y \leq -10\,\mathrm{km\,h^{-1}}$, $-20\,\mathrm{deg/s} \leq \omega \leq 20\,\mathrm{deg/s}$

The following polytopic set results from these propositions:

$$\mathcal{C}_{\bar{a}_X} = \left\{ \begin{bmatrix} \bar{a}_X \\ v_Y \end{bmatrix} \middle| |v_Y| \leq v_{Y,\max}, \bar{a}_X - v_Y\omega_{\max} \leq \bar{a}_{X,\max}, \bar{a}_X + v_Y\omega_{\max} \leq \bar{a}_{X,\max}, \right.$$
$$\left. -\bar{a}_X - v_Y\omega_{\max} \leq -\bar{a}_{X,\min}, -\bar{a}_X + v_Y\omega_{\max} \leq -\bar{a}_{X,\min} \right\}$$

or equivalently, lifting to the full velocity-acceleration-space,

$$\mathcal{C}_{\bar{a}_X} := \left\{ \begin{bmatrix} v_X \\ v_Y \\ \omega \\ \bar{a}_X \\ \bar{a}_Y \end{bmatrix} \middle| \begin{bmatrix} 0 & 1 \\ 0 & -1 \\ 1 & -\omega_{\max} \\ 1 & \omega_{\max} \\ -1 & -\omega_{\max} \\ -1 & \omega_{\max} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_X \\ v_Y \\ \omega \\ \bar{a}_X \\ \bar{a}_Y \end{bmatrix} \leq \begin{bmatrix} v_{Y,\max} \\ v_{Y,\max} \\ \bar{a}_{X,\max} \\ \bar{a}_{X,\max} \\ -\bar{a}_{X,\min} \\ -\bar{a}_{X,\min} \end{bmatrix} \right\}.$$

A plot of the resulting set is shown in Fig. A.2a. The resulting degree of conservatism is small (however, depending on the bounds on ω). Note that instead of deriving constraints for all ω, it is possible to identify constraints which hold for all v_X. Because the derivation parallels the one shown and because the result is much more conservative, it is omitted.

Linear bounds on $\bar{a}_Y$

Even though the set shown in Fig. A.1b appears to be more difficult to linearize than the set $\tilde{\mathcal{C}}_{\bar{a}_X}$, a similar approach is chosen, i.e., a polytopic constraint set is derived which holds *for all* v_X, such that it is valid over the entire operating range of the vehicle:

$$\left\{ \begin{bmatrix} \bar{a}_Y \\ \omega \end{bmatrix} | \forall v_X \in [v_{X,\min}, v_{X,\max}] \ : \ -\bar{a}_{Y,\max} \leq \bar{a}_Y - v_X\omega \leq \bar{a}_{Y,\max} \right\}$$

$$\Leftrightarrow$$

$$\left\{ \begin{bmatrix} \bar{a}_Y \\ \omega \end{bmatrix} | \max_{v_X \in [v_{X,\min}, v_{X,\max}]} \bar{a}_Y - v_X\omega \leq \bar{a}_{Y,\max}, \quad \min_{v_X \in [v_{X,\min}, v_{X,\max}]} \bar{a}_Y - v_X\omega \geq \bar{a}_{Y,\min} \right\}$$

Paralleling the derivations for the longitudinal accelerations, it holds that:

Proposition A.5.

$$\max_{v_X \in [v_{X,\min}, v_{X,\max}]} \bar{a}_Y - v_X\omega = \begin{cases} \bar{a}_Y - v_{X,\max}\omega & \omega < 0 \\ \bar{a}_Y - v_{X,\min}\omega & \omega \geq 0 \end{cases}.$$

Proof. For the maximization problem, the Lagrange function reads:

$$L(v_X, \lambda_1, \lambda_2) = -(\bar{a}_Y - v_X\omega) + \lambda_1(v_X - v_{X,\max}) + \lambda_2(-v_X + v_{X,\min}),$$

while the following KKT-conditions result:

$$\begin{aligned} \nabla_{v_X} L = \omega + \lambda_1 - \lambda_2 = 0, \\ v_X = v_{X,\max} \Rightarrow \lambda_2 = 0 \Rightarrow \lambda_1 = -\omega > 0, \\ v_X = v_{X,\min} \Rightarrow \lambda_1 = 0 \Rightarrow \lambda_2 = \omega > 0. \end{aligned}$$

□

Similarly,

Proposition A.6.

$$\min_{v_X \in [v_{X,\min}, v_{X,\max}]} \bar{a}_Y - v_X\omega = \begin{cases} \bar{a}_Y - v_{X,\min}\omega & \omega < 0 \\ \bar{a}_Y - v_{X,\max}\omega & \omega \geq 0 \end{cases}$$

Proof.

$$L(v_X, \lambda_1, \lambda_2) = \bar{a}_Y - v_X\omega + \lambda_1(v_X - v_{X,\max}) + \lambda_2(-v_X + v_{X,\min})$$

KKT-conditions:

$$\begin{aligned} \nabla_{v_X} L = -\omega + \lambda_1 - \lambda_2 = 0, \\ v_X = v_{X,\max} \Rightarrow \lambda_2 = 0 \Rightarrow \lambda_1 = \omega > 0, \\ v_X = v_{X,\min} \Rightarrow \lambda_1 = 0 \Rightarrow \lambda_2 = -\omega > 0. \end{aligned}$$

□

The following polytopic set results from these propositions:

$$\left\{ \begin{bmatrix} \bar{a}_Y \\ \omega \end{bmatrix} \middle| \, |\omega| \leq \omega_{\max}, \bar{a}_Y - v_{X,\max}\omega \leq \bar{a}_{Y,\max}, \bar{a}_Y - v_{X,\min}\omega \leq \bar{a}_{Y,\max}, \right.$$
$$\left. -\bar{a}_Y + v_{X,\min}\omega \leq -\bar{a}_{Y,\min}, -\bar{a}_Y + v_{X,\max}\omega \leq -\bar{a}_{Y,\min} \right\}$$

or equivalently, lifting to the full velocity-acceleration-space,

$$\mathcal{C}_{\bar{a}_Y} := \left\{ \begin{bmatrix} v_X \\ v_Y \\ \omega \\ \bar{a}_X \\ \bar{a}_Y \end{bmatrix} \middle| \begin{bmatrix} 0 & 1 \\ 0 & -1 \\ 1 & -v_{X,\max} \\ 1 & -v_{X,\min} \\ -1 & v_{X,\min} \\ -1 & v_{X,\max} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_X \\ v_Y \\ \omega \\ \bar{a}_X \\ \bar{a}_Y \end{bmatrix} \leq \begin{bmatrix} \omega_{\max} \\ \omega_{\max} \\ \bar{a}_{Y,\max} \\ \bar{a}_{Y,\max} \\ -\bar{a}_{Y,\min} \\ -\bar{a}_{Y,\min} \end{bmatrix} \right\}. \tag{A.7}$$

See Fig. A.2b for a plot of the resulting set. Admissible combinations of accelerations and velocities are then summarized in the set:

$$\mathcal{A} := \mathcal{C}_{\bar{a}_X} \cap \mathcal{C}_{\bar{a}_Y}.$$

E. Linearization of Front Side Slip Angle Constraints

A magnitude constraint on (6.7) is equivalent to a velocity-dependent interval constraint:

$$-\tilde{\alpha}_{\max} + \arctan\left(\frac{l_f\omega + v_Y}{v_X}\right) \leq \delta \leq \tilde{\alpha}_{\max} + \arctan\left(\frac{l_f\omega + v_Y}{v_X}\right)$$

on the front wheel angle δ. The resulting bounds on δ are shown in Fig. A.3a. Working towards linear constraints, introduce:

$$R_3\left(\frac{l_f\omega + v_Y}{v_X}\right) := \arctan\left(\frac{l_f\omega + v_Y}{v_X}\right) - \frac{l_f\omega + v_Y}{v_X}, \tag{A.8}$$

such that:

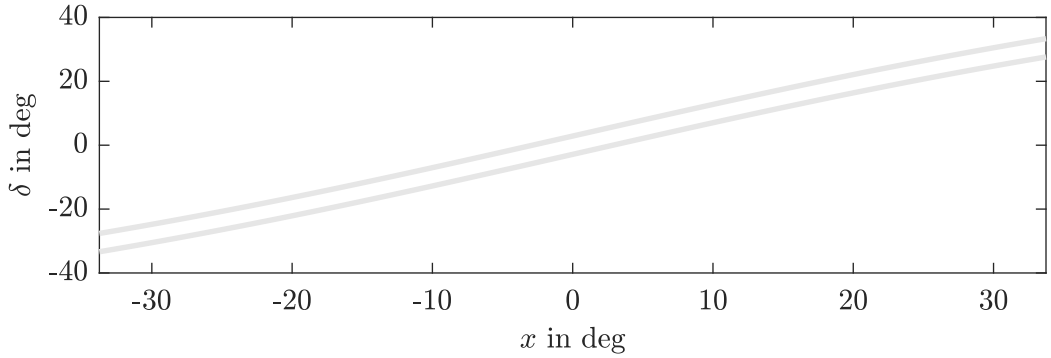
$$\left|\delta - \arctan\left(\frac{l_f\omega + v_Y}{v_Y}\right)\right| = \left|\delta - \frac{l_f\omega + v_Y}{v_X} - R_3\left(\frac{l_f\omega + v_Y}{v_X}\right)\right|,$$

from which it follows that

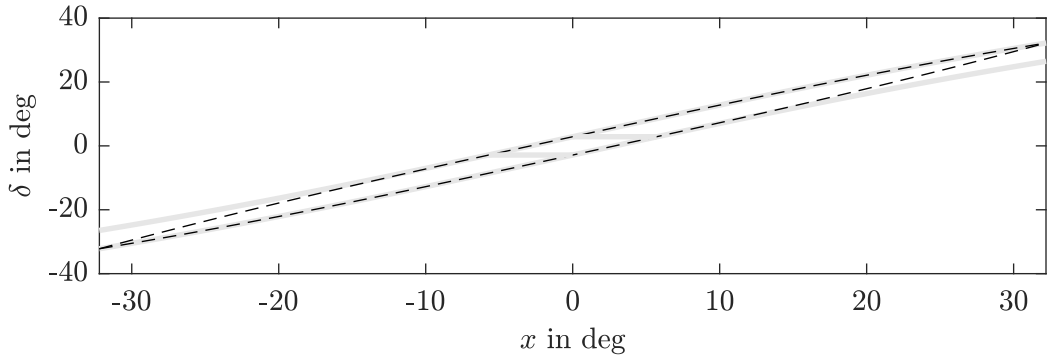
$$\left|\delta - \frac{l_f\omega + v_Y}{v_X}\right| \leq \tilde{\alpha}_{\max} - \left|R_3\left(\frac{l_f\omega + v_Y}{v_X}\right)\right| \Rightarrow \left|\delta - \arctan\left(\frac{l_f\omega + v_Y}{v_X}\right)\right| \leq \tilde{\alpha}_{\max}.$$

Along with the magnitude constraint on δ according to (6.6), this defines the set:

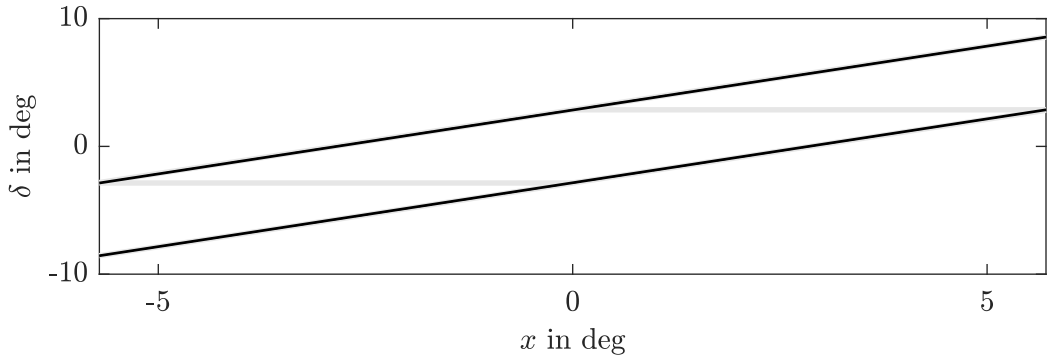
$$\left\{ \begin{bmatrix} \delta \\ x \end{bmatrix} \middle| \, |\delta - x| \leq \tilde{\alpha}_{\max} - |R_3(x)|, |\delta| \leq \delta_{\max} \right\} \tag{A.9}$$



(a) Interval constraints on δ, resulting from constraint on α_f.



(b) Constraints on δ after series expansion (dashed), combined with input magnitude constraints (horizontal lines).



(c) Constraints when using the conservative bound on the series expansion remainder (solid black, zoomed in).

Figure A.3.: Bounds on δ in dependency of $x = \frac{l_\mathrm{f}\omega + v_Y}{v_X}$.

which is shown in Fig. A.3b. In order for this set to be non-empty, it is necessary that

$$\left| R_3 \left(\frac{l_\mathrm{f}\omega + v_Y}{v_X} \right) \right| < \tilde{\alpha}_\mathrm{max}.$$

The relation holds if it holds for the maximum possible value:

$$\left| R_3 \left(\frac{l_\mathrm{f}\omega + v_Y}{v_X} \right) \right| \leq \left| R_3 \left(\frac{l_\mathrm{f}\omega_\mathrm{max} + v_{Y,\mathrm{max}}}{v_{X,\mathrm{min}}} \right) \right| =: \bar{R}_3.$$

As the variable remainder R_3 introduces a dependency of (A.9) on a trigonometric term, it is replaced by the maximum $\bar{R}_3$. Introducing this into the first inequality of (A.9) gives:

$$\frac{l_\mathrm{f}\omega + v_Y}{v_X} - \tilde{\alpha}_\mathrm{max} + \bar{R}_3 \leq \delta \leq \tilde{\alpha}_\mathrm{max} + \frac{l_\mathrm{f}\omega + v_Y}{v_X} - \bar{R}_3.$$

With $\alpha_\mathrm{max} := \tilde{\alpha}_\mathrm{max} - \bar{R}_3$, the constraint (6.8) results, which is depicted in Figure A.3c.

F. Projection of $\mathcal{C}_{v\bar{a}\mu}$ on the Velocity-Acceleration-Space

Computation of the projection (6.11) relies on cylindrical algebraic decomposition (CAD), a procedure which can be used for quantifier elimination in logical expressions involving polynomials [32], implemented in symbolic computation software such as Wolfram Mathematica [152] or Maple [102]. Because CAD is computationally demanding, the original expression (6.10) is simplified prior to application of CAD. First, it is noted that:

$$\mathtt{proj}_{v\bar{a}} \left(\left\{ \begin{bmatrix} v \\ \bar{a} \end{bmatrix} | v \in \mathcal{V}, \begin{bmatrix} v \\ \bar{a} \end{bmatrix} \in \mathcal{A} \right\} \times \mathbb{R}^2 \cap \mathcal{U} \right) = \left\{ \begin{bmatrix} v \\ \bar{a} \end{bmatrix} | v \in \mathcal{V}, \begin{bmatrix} v \\ \bar{a} \end{bmatrix} \in \mathcal{A} \right\} \cap \mathtt{proj}_{v\bar{a}} \left(\mathcal{U} \right). \tag{A.10}$$

Thus, projecting $\mathcal{C}_{v\bar{a}\mu}$ reduces to the problem of projecting the set $\mathcal{U}$ defined in (6.9). In a second step, the following substitutions are made in the definition of (6.9):

$$b := -\frac{l_\mathrm{f}\omega + v_Y}{v_X},\ c := \frac{m}{\bar{c}_Y}\bar{a}_X + \frac{1}{\gamma},\ d := \frac{\omega(l_\mathrm{r} - l_\mathrm{f}) - 2v_Y}{\gamma v_X} - \frac{m}{\gamma \bar{c}_Y}\bar{a}_Y. \tag{A.11}$$

Third, interval bounds on these quantities are derived. Note that these bounds are not necessarily tight, but nonetheless helpful for computations. Considering (6.8), bounds on b can be obtained from:

$$b_\mathrm{max} := \arg\max_{b,\delta} b \ \text{ s.t. } |\delta + b| \leq \alpha_\mathrm{max},\ |\delta| \leq \delta_\mathrm{max}, \tag{A.12}$$

$$b_\mathrm{min} := \arg\min_{b,\delta} b \ \text{ s.t. } |\delta + b| \leq \alpha_\mathrm{max},\ |\delta| \leq \delta_\mathrm{max}, \tag{A.13}$$

giving $b_\mathrm{max} = \alpha_\mathrm{max} - \delta_\mathrm{min}$ and $b_\mathrm{min} = -b_\mathrm{max}$. Similarly,

$$c_\mathrm{max} := \arg\max_{\bar{a}_X, v_Y} \frac{m}{\bar{c}_Y}\bar{a}_X + \frac{1}{\gamma} \text{ s.t. } \begin{bmatrix} \bar{a}_X & v_Y \end{bmatrix}^\mathsf{T} \in \mathcal{C}_{\bar{a}_X},$$

$$c_\mathrm{min} := \arg\min_{\bar{a}_X, v_Y} \frac{m}{\bar{c}_Y}\bar{a}_X + \frac{1}{\gamma} \text{ s.t. } \begin{bmatrix} \bar{a}_X & v_Y \end{bmatrix}^\mathsf{T} \in \mathcal{C}_{\bar{a}_X}.$$

Based on (A.11),

$$d - \frac{b}{\gamma} = \frac{\omega l_\mathrm{r} - v_Y}{\gamma v_X} - \frac{m}{\gamma \bar{c}_Y} \bar{a}_Y.$$

Considering the side-slip constraint (6.2), the bounds on b according to (A.12) and (A.13), and the constraints on $\bar{a}_Y$ according to (A.7), it holds that $d_\mathrm{min} \leq d - \frac{b}{\gamma} \leq d_\mathrm{max}$, with

$$\begin{aligned} d_\mathrm{min} &:= \min_{\bar{a}_Y, \omega} -\frac{\tan(\alpha_\mathrm{max})}{\gamma} - \frac{m}{\gamma \bar{c}_Y} \bar{a}_Y + \frac{b_\mathrm{min}}{\gamma} \text{ s.t. } \begin{bmatrix} \bar{a}_Y & \omega \end{bmatrix}^\mathsf{T} \in \mathcal{C}_{\bar{a}_Y}, \\ d_\mathrm{max} &:= \max_{\bar{a}_Y, \omega} \frac{\tan(\alpha_\mathrm{max})}{\gamma} - \frac{m}{\gamma \bar{c}_Y} \bar{a}_Y + \frac{b_\mathrm{max}}{\gamma} \text{ s.t. } \begin{bmatrix} \bar{a}_Y & \omega \end{bmatrix}^\mathsf{T} \in \mathcal{C}_{\bar{a}_Y}. \end{aligned}$$

Given these bounds, the objective is to search for a set

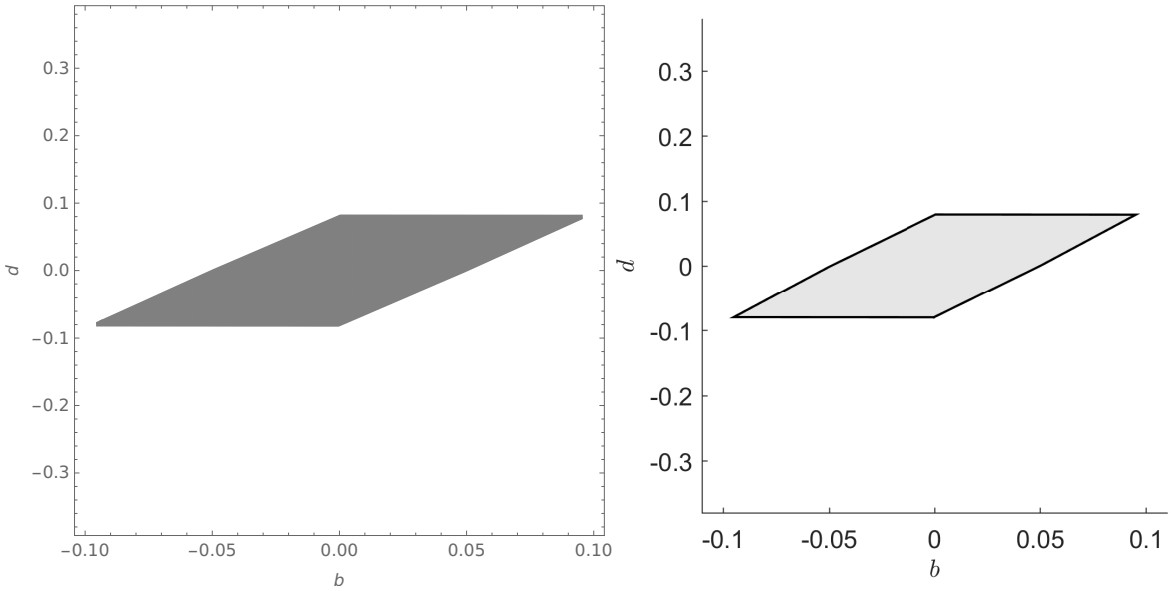
$$\begin{aligned} \mathcal{C}_I := \Bigg\{ \begin{bmatrix} b \\ c \\ d \end{bmatrix} \; & | \exists \delta : |\delta + b| \leq \alpha_\mathrm{max}, \; \left| \delta^2 + b\delta + c - \frac{1}{\gamma} \right| \leq \frac{\bar{c}_X}{\bar{c}_Y} s_{X,\mathrm{max}}, \\ & |\delta \in \mathbb{R}| \leq \delta_\mathrm{max}, \; \delta^3 + b\delta^2 + \delta c + d = 0, \; |b| \leq b_\mathrm{max}, \; c_\mathrm{min} \leq c \leq c_\mathrm{max}, \; d_\mathrm{min} \leq d \leq d_\mathrm{max} \Bigg\}. \end{aligned} \tag{A.14}$$

Using Wolfram Mathematica [152], the following set results, based on the parameter set in Table 5.1 (note: the representation of numbers is owed to Wolfram Mathematica):

$$\mathcal{C}_I = \left\{ \begin{bmatrix} b \\ c \\ d \end{bmatrix} \mid \left(b = -\frac{1329191146657}{20000000\sqrt{116662779049} + 6869720980000} \wedge d = \frac{-20b - 649}{8000} \right) \vee \right.$$
$$\left(-\frac{1329191146657}{20000000\sqrt{116662779049} + 6869720980000} < b \leq -\frac{993}{20000} \wedge \right.$$
$$\left. \frac{-20b - 649}{8000} \leq d \leq \frac{3972000000000b^2 + 137194419600000b + 680191146657}{8000000000000} \right) \vee$$
$$\left(-\frac{993}{20000} < b \leq -\frac{7}{20000} \wedge \right.$$
$$\left. \frac{-20b - 649}{8000} \leq d \leq \frac{3972000000000b^2 + 129994419600000b + 644443146657}{8000000000000} \right) \vee$$
$$\left(-\frac{7}{20000} < b \leq \frac{7}{20000} \wedge \frac{-3972000000000b^2 + 129994419600000b - 644443146657}{8000000000000} \leq d \right.$$
$$\left. \leq \frac{3972000000000b^2 + 129994419600000b + 644443146657}{8000000000000} \right)$$
$$\vee \left(\frac{7}{20000} < b \leq \frac{993}{20000} \wedge \right.$$
$$\left. \frac{-3972000000000b^2 + 129994419600000b - 644443146657}{8000000000000} \leq d \leq \frac{649 - 20b}{8000} \right) \vee$$
$$\left(\frac{993}{20000} < b < \frac{1329191146657}{20000000\sqrt{116662779049} + 6869720980000} \wedge \right.$$
$$\left. \frac{-3972000000000b^2 + 137194419600000b - 680191146657}{8000000000000} \leq d \leq \frac{649 - 20b}{8000} \right) \vee$$
$$\left. \left(b = \frac{1329191146657}{20000000\sqrt{116662779049} + 6869720980000} \wedge d = \frac{649 - 20b}{8000} \right) \wedge \frac{81}{50} \leq c \leq \frac{171}{100} \right\}.$$

Note that this set not necessarily ensures that a unique real root δ of the cubic polynomial in (A.14) exists on the considered interval. In principle, this could be enforced by requiring the discriminant of the polynomial to be positive during the computation, which, however, turned out to be computationally too demanding. Instead, the sign was checked a posteriori using Mathematica.

Also note that the variable c is not coupled with the other variables, such that it suffices to consider the projection of the set on the b-d-plane, as shown in Fig. A.4 (a). Even though this set appears to be convex, its representation is based on nonlinear polynomials,



(a) Projection of the nonlinear set $\mathcal{C}_I$.

(b) Projection of the polytopic inner approximation $\mathcal{C}_{II}$.

Figure A.4.: Projection of the nonlinear set $\mathcal{C}_I$ and of the polytopic inner approximation $\mathcal{C}_{II}$ on the b-d-plane.

whereas the following steps require a polytopic representation. Therefore, a polytopic inner approximation of $\mathcal{C}_I$ is obtained (cf. Fig. A.4(b)):

$$\mathcal{C}_{II} := \left\{ \begin{bmatrix} b \\ c \\ d \end{bmatrix} \mid A_{II} \begin{bmatrix} b \\ c \\ d \end{bmatrix} \leq b_{II} \right\} \subseteq \mathcal{C}_I. \tag{A.15}$$

In a next step, the auxiliary parameters b, c, and d are re-substituted in order to obtain constraints on the original quantities. To that end, (A.11) is rewritten as:

$$\begin{bmatrix} b \\ c \\ d \end{bmatrix} =: \begin{bmatrix} -l_\mathrm{f} & -1 & 0 & 0 \\ 0 & 0 & \frac{m}{\bar{c}_Y} & 0 \\ \frac{l_\mathrm{r}-l_\mathrm{f}}{\gamma} & -\frac{2}{\gamma} & 0 & -\frac{m}{\bar{c}_Y\gamma} \end{bmatrix} \begin{bmatrix} \frac{1}{v_X} & & & 0 \\ & \frac{1}{v_X} & & \\ & & 1 & \\ 0 & & & 1 \end{bmatrix} \begin{bmatrix} \omega \\ v_Y \\ \bar{a}_X \\ \bar{a}_Y \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{\gamma} \\ 0 \end{bmatrix}.$$

Introducing this into (A.15) leads to a set $\mathcal{C}_{III}$:

$$\mathcal{C}_{III} := \left\{ \begin{bmatrix} v_X \\ v_Y \\ \omega \\ \bar{a}_X \\ \bar{a}_Y \end{bmatrix} \mid A_{III} \begin{bmatrix} \omega \\ v_Y \\ \bar{a}_X v_X \\ \bar{a}_Y v_X \end{bmatrix} \leq b_{III} \cdot v_X \right\}$$

with

$$A_{III} := A_{II} \begin{bmatrix} -l_\mathrm{f} & -1 & 0 & 0 \\ 0 & 0 & \frac{m}{\bar{c}_Y} & 0 \\ \frac{l_\mathrm{r}-l_\mathrm{f}}{\gamma} & -\frac{2}{\gamma} & 0 & -\frac{m}{\bar{c}_Y \gamma} \end{bmatrix}, \qquad b_{III} := \left(b_{II} - A_{II} \begin{bmatrix} 0 \\ \frac{1}{\gamma} \\ 0 \end{bmatrix} \right),$$

which can equivalently be expressed as:

$$\mathcal{C}_{III} = \left\{ \begin{bmatrix} v_X \\ v_Y \\ \omega \\ \bar{a}_X \\ \bar{a}_Y \end{bmatrix} \mid \begin{bmatrix} A_{III}[:, 1:2] & -b_{III} \end{bmatrix} \begin{bmatrix} \omega \\ v_Y \\ v_X \end{bmatrix} \leq -A_{III}[:, 3:4] \begin{bmatrix} \bar{a}_X \\ \bar{a}_Y \end{bmatrix} v_X \right\}. \tag{A.16}$$

Working towards a polytopic set representation, the nonlinear products of accelerations and longitudinal velocity are linearized by finding lower bounds on the right-hand side:

$$\begin{aligned} \min_{v_X \in [v_{X,\min}, v_{X,\max}]} & - A_{III}[i, 3:4] \begin{bmatrix} \bar{a}_X \\ \bar{a}_Y \end{bmatrix} v_X \\ &= \begin{cases} -A_{III}[i, 3:4] \begin{bmatrix} \bar{a}_X \\ \bar{a}_Y \end{bmatrix} v_{X,\min} & \text{if } -A_{III}[i, 3:4] \begin{bmatrix} \bar{a}_X \\ \bar{a}_Y \end{bmatrix} > 0 \\ -A_{III}[i, 3:4] \begin{bmatrix} \bar{a}_X \\ \bar{a}_Y \end{bmatrix} v_{X,\max} & \text{if } -A_{III}[i, 3:4] \begin{bmatrix} \bar{a}_X \\ \bar{a}_Y \end{bmatrix} < 0 \end{cases}. \end{aligned}$$

Replacing row-wise the right-hand side of (A.16) with these minima gives:

$$\begin{aligned} \mathcal{C}_{IV} :&= \left\{ \begin{bmatrix} v_X \\ v_Y \\ \omega \\ \bar{a}_X \\ \bar{a}_Y \end{bmatrix} \mid \begin{bmatrix} A_{III}[:, 1:2] & -b_{III} & A_{III}[:, 3:4] v_{X,\min} \\ A_{III}[:, 1:2] & -b_{III} & A_{III}[:, 3:4] v_{X,\max} \end{bmatrix} \begin{bmatrix} \omega \\ v_Y \\ v_X \\ \bar{a}_X \\ \bar{a}_Y \end{bmatrix} \leq 0 \right\} \\ &= \left\{ \begin{bmatrix} v_X \\ v_Y \\ \omega \\ \bar{a}_X \\ \bar{a}_Y \end{bmatrix} | A_{IV} \begin{bmatrix} v_X \\ v_Y \\ \omega \\ \bar{a}_X \\ \bar{a}_Y \end{bmatrix} \leq 0 \right\}, \end{aligned}$$

where

$$A_{IV} := \begin{bmatrix} -b_{III} & A_{III}[:, 2:1] & A_{III}[:, 3:4] v_{X,\min} \\ -b_{III} & A_{III}[:, 2:1] & A_{III}[:, 3:4] v_{X,\max} \end{bmatrix},$$

and clearly,

$$\mathcal{C}_{IV} \subset \mathcal{C}_{III}.$$

In a last step, the intersection with the sets $\mathcal{A}$ and $\mathcal{V}$ according to (A.10) is carried out, yielding:

$$\mathcal{C}_{v\bar{a}} := \left\{ \begin{bmatrix} v \\ \bar{a} \end{bmatrix} | v \in \mathcal{V}, \begin{bmatrix} v \\ \bar{a} \end{bmatrix} \in \mathcal{C}_{IV} \cap \mathcal{A} \right\}.$$

Parameter	p_1	p_2	p_3	p_4	p_5	p_6	w_2
Correspondence	$\dot{s}$	$\dot{\theta}$	$\ddot{s}$	$\dot{\theta}\dot{s}$	$\frac{1}{v_X}$	$\frac{\dot{s}}{v_X}$	$\frac{m}{\Theta} l_\mathrm{f} \dot{\theta}\dot{s} - \frac{\bar{c}_Y l_\mathrm{r}(l_\mathrm{f}+l_\mathrm{r})}{\Theta} \frac{\dot{\theta}}{v_X}$

Table A.1.: Parameters and corresponding expressions.

G. Derivation of Parameter Bounds

The objective of this section is to determine a polytope $\mathcal{P}$ as used in Theorem 7.2, requiring to over-approximate the range of the non-convex function f_param as defined in (7.9) on the convex domain $\mathcal{E} \times \tilde{\mathcal{C}}_{\bar{x}}$. Starting point is the following:

Theorem A.1 (Parameter set $\mathcal{P}$)**.** *With $\mathcal{E}$ according to Assumption 6.1 and $\mathcal{C}_{\bar{x}} \subset \mathbb{R}^7$ from (6.16), denote by $\mathcal{Y}$ the image of $\mathcal{E} \times \mathcal{C}_{\bar{x}}$ under a function $f_{param} : \mathcal{E} \times \mathcal{C}_{\bar{x}} \to \mathbb{R}^4$. Further, assume to be given a set $\mathcal{I} = \{1, 2, 3, 4\}$ and matrices A_i and vectors b_i, defining polytopes*

$$\mathcal{P}_i := \left\{ \begin{bmatrix} \bar{x} \\ z_i \end{bmatrix} | \bar{x} \in \mathcal{C}_{\bar{x}},\ z_i \in \mathbb{R},\ A_i \begin{bmatrix} \bar{x} \\ z_i \end{bmatrix} \leq b_i \right\} \subset \mathbb{R}^8,\ i \in \mathcal{I},$$

for which

$$\begin{bmatrix} \bar{x} \\ f_{param,i}(e, \bar{x}) \end{bmatrix} \in \mathcal{P}_i\ \forall e \in \mathcal{E}\ \forall \bar{x} \in \mathcal{C}_{\bar{x}}.$$

Then, with the set

$$\mathcal{P} := \mathtt{proj}_{z_1,z_2,z_3,z_4} \bigcap_{i \in \mathcal{I}} \left\{ \begin{bmatrix} \bar{x}^\mathsf{T} & z_1 & z_2 & z_3 & z_4 \end{bmatrix}^\mathsf{T} | \begin{bmatrix} \bar{x} \\ z_i \end{bmatrix} \in \mathcal{P}_i,\ z_j \in \mathbb{R} \text{ for } j \neq i \right\}, \quad \text{(A.17)}$$

it holds that

$$\mathcal{Y} \subseteq \mathcal{P}.$$

Proof. For each element of the image set it holds that:

$$y \in \mathcal{Y} \Rightarrow \exists e \in \mathcal{E}, \bar{x} \in \tilde{\mathcal{C}}_{\bar{x}} :\ y = f_\mathrm{param}(e, \bar{x}).$$

By requirement, this implies that

$$\begin{bmatrix} \bar{x} \\ y_i \end{bmatrix} \in \mathcal{P}_i\ \forall\ i \in \mathcal{I},$$

which is equivalent to:

$$\begin{bmatrix} \bar{x} \\ y \end{bmatrix} \in$$
$$\left\{ \begin{bmatrix} \bar{x} \\ y \end{bmatrix} | \begin{bmatrix} \bar{x} \\ y_i \end{bmatrix} \in \mathcal{P}_i,\ i \in \mathcal{I} \right\} = \bigcap_{i \in \mathcal{I}} \left\{ \begin{bmatrix} \bar{x}^\mathsf{T} & y_1 & y_2 & y_3 & y_4 \end{bmatrix}^\mathsf{T} | \begin{bmatrix} \bar{x} \\ y_i \end{bmatrix} \in \mathcal{P}_i,\ y_j \in \mathbb{R} \text{ for } j \neq i \right\},$$

from which (A.17) follows.

□

The computation of the polytopes $\mathcal{P}_i$ is detailed in the following. Note that it is simple to obtain the polytope $\mathcal{P}_1$, as it contains admissible values of $f_{\text{param},1} \equiv \ddot{s} = \bar{x}_{[3]}$ according to (5.2), which is already constrained by $\mathcal{C}_{\bar{x}}$. Thus, the (low-dimensional) set

$$\mathcal{P}_1 := \left\{ \begin{bmatrix} \bar{x} \\ z_1 \end{bmatrix} | \bar{x} \in \mathcal{C}_{\bar{x}}, z_1 = \bar{x}_{[3]} \right\}$$

fulfills the requirements.

Polytope $\mathcal{P}_2$

In order to derive bounds on $\frac{1}{v_X}$ (the second component of f_{param}) note that according to (5.31),

$$v_X = \begin{bmatrix} 1 & 0 \end{bmatrix} R(e_\psi)^\mathsf{T} \begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_{\text{pos}} \\ \dot{e}_{\text{pos}} \end{bmatrix} + \dot{s} \cos e_\psi \leq \left|\left| \begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_{\text{pos}} \\ \dot{e}_{\text{pos}} \end{bmatrix} \right|\right| + |\dot{s} \cos e_\psi|.$$

While this expression depends on the errors as well as the two reference trajectory quantities $\dot{\theta}$ and $\dot{s}$, the impact of $\dot{\theta}$ is small. With a set $\mathcal{E}$ of admissible errors as in Assumption 6.1, the need to further account for both the errors and $\dot{\theta}$ is eliminated by noting that

$$\left|\left| \begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_{\text{pos}} \\ \dot{e}_{\text{pos}} \end{bmatrix} \right|\right| \leq \max_{\bar{x} \in \mathcal{C}_{\bar{x}}} \left|\left| \begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \end{bmatrix} \right|\right| \cdot \frac{1}{\xi^2} = \frac{1}{\xi^2} \sqrt{1 + \dot{\theta}_{\max}^2} =: n_1,$$

and especially, $\dot{s} \cos e_\psi \gg n_1 \ \forall \ e_\psi \in \texttt{proj}_{e_\psi} \mathcal{E}$ because $e_{\psi,\max} := \max \texttt{proj}_{e_\psi} \mathcal{E}$ is small. Therefore,

$$\cos(e_{\psi,\max}) \dot{s} - n_1 \leq v_X \leq \dot{s} + n_1$$

and consequently

$$\frac{1}{\dot{s} + n_1} \leq \frac{1}{v_X} \leq \frac{1}{\cos(e_{\psi,\max}) \dot{s} - n_1} \ \forall e \in \mathcal{E}. \tag{A.18}$$

Computation of the McCormick envelope [107] of the graph of the upper and lower bounds of $\frac{1}{v_X}$ on $\texttt{proj}_{\dot{s}} \mathcal{C}_{\bar{x}}$ then gives the set shown in Figure A.5, which can be lifted to the full space of $\bar{x}$ to give the desired polytope $\mathcal{P}_2$. Note that this can be done automatically, e.g. relying on Yalmip [96].

Polytope $\mathcal{P}_3$

Bounds on the third component of f_{param}, $\frac{\dot{s}}{v_X}$, are obtained similarly to the computation of $\mathcal{P}_2$. From (A.18), it follows that

$$\frac{\dot{s}}{\dot{s} + n_1} \leq \frac{\dot{s}}{v_X} \leq \frac{\dot{s}}{\cos(e_{\psi,\max}) \dot{s} - n_1} \ \forall \ e \in \mathcal{E}.$$

Because the upper (lower) bound is a convex (concave) function, a polytopic over-approximation is obtained easily, cf. Figure A.6, which can again be lifted to the full parameter space.

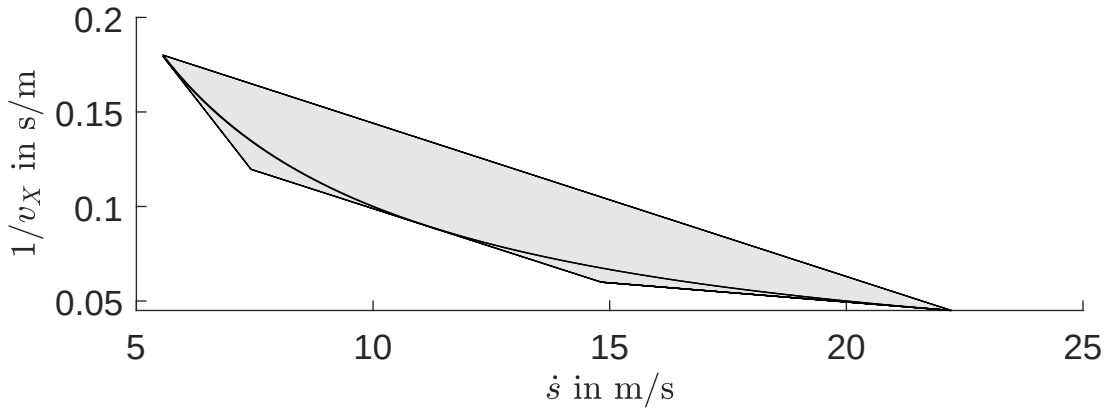


Figure A.5.: McCormick-envelope of $\frac{1}{v_X}$.

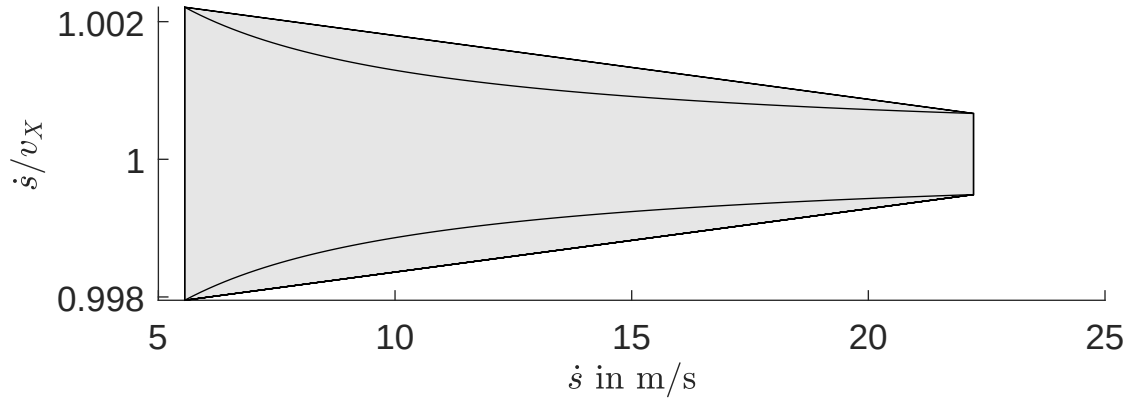


Figure A.6.: Polytopic over-approximation of the bounds on the graph of $\frac{\dot{s}}{v_X}$.

Polytope $\mathcal{P}_4$

The fourth component of f_{param}:

$$\frac{m}{\Theta} l_{\text{f}} \dot{\theta} \dot{s} - \frac{\bar{c}_Y l_{\text{r}} (l_{\text{f}} + l_{\text{r}})}{\Theta} \frac{\dot{\theta}}{v_X} \tag{A.19}$$

depends on both $\dot{s}$ and $\dot{\theta}$, if v_X is not perceived as function as in (5.31), but as parameter which is bounded along with $\bar{x}$ by $\mathcal{P}_1$. Plotting (A.19) for minimum and maximum values of v_X on $\mathcal{P}_1$ gives the non-convex set shown in Figure A.7, along with a (manually obtained) bounding polytope. Again, a lifting gives $\mathcal{P}_4$.

H. Bounds on $\tilde{w}_1$

Theorem A.2. *Given the function $\tilde{w}_1(e, \bar{x}, \Delta a)$ as defined in (7.11) and the corresponding domain $\mathcal{D}_{yaw}$, it holds that*

$$\tilde{w}_1(e, \bar{x}, \Delta a) \leq w_{1,\max}\left(f_{param}(e, \bar{x})\right) \ \forall \left[e^{\mathsf{T}} \bar{x}^{\mathsf{T}} \Delta a^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathcal{D}_{yaw},$$

where according to (7.10),

$$w_{1,\max}(p) := c_1 + c_2 \cdot p_5,$$

with

$$c_1 := \Delta a_{\psi,\max} + \ddot{\theta}_{\max} + \left|\left|\begin{bmatrix}\cos e_{\psi,\max} - 1 & \sin e_{\psi,\max} - e_{\psi,\max}\end{bmatrix}\right|\right| \dots$$
$$+ \frac{m}{\Theta} l_f \cdot \max_{\bar{x} \in \mathcal{C}_{\bar{x}}} \left|\left| \left(-K + \begin{bmatrix} \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix} \right) P_{pos}^{-\frac{1}{2}} \right|\right| + \max_{p \in \mathcal{P}} \left|\left| \begin{bmatrix} \frac{m}{\Theta} l_f p_4 \\ -\frac{m}{\Theta} l_f p_3 - \frac{\bar{c}_Y (l_f + l_r)}{\Theta} p_6 \end{bmatrix} \right|\right|$$

and

$$c_2 := \frac{\bar{c}_Y (l_f + l_r)}{\Theta} \cdot \max_{\bar{x} \in \mathcal{C}_{\bar{x}}} \left|\left| \begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \end{bmatrix} P_{pos}^{-\frac{1}{2}} \right|\right|. \tag{A.20}$$

Proof. From the definition of $\tilde{w}_1$ in (7.11) and the triangle inequality, it holds that:

$$\begin{aligned}
\tilde{w}_1(e, \bar{x}, \Delta a) &\leq ||\tilde{w}_1(e, \bar{x}, \Delta a)|| \\
&\leq \left|\left|\begin{bmatrix}\cos e_\psi - 1 & \sin e_\psi - e_\psi\end{bmatrix}\right|\right| \cdot \left|\left| \begin{bmatrix} \frac{m}{\Theta} l_{\text{f}} \dot{\theta} \dot{s} \\ -\frac{m}{\Theta} l_{\text{f}} \ddot{s} - \frac{\bar{c}_Y (l_{\text{f}} + l_{\text{r}})}{\Theta} \frac{\dot{s}}{v_X} \end{bmatrix} \right|\right| + |\Delta a_\psi| + |\ddot{\theta}| \dots \\
&+ \frac{m}{\Theta} l_{\text{f}} \left|\left| \left(-K + \begin{bmatrix} \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix} \right) \begin{bmatrix} e_{\text{pos}} \\ \dot{e}_{\text{pos}} \end{bmatrix} \right|\right| \dots \\
&+ p_5 \frac{\bar{c}_Y (l_{\text{f}} + l_{\text{r}})}{\Theta} \left|\left| \begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_{\text{pos}} \\ \dot{e}_{\text{pos}} \end{bmatrix} \right|\right| \\
&\leq c_2 \cdot p_5 + c_1
\end{aligned}$$

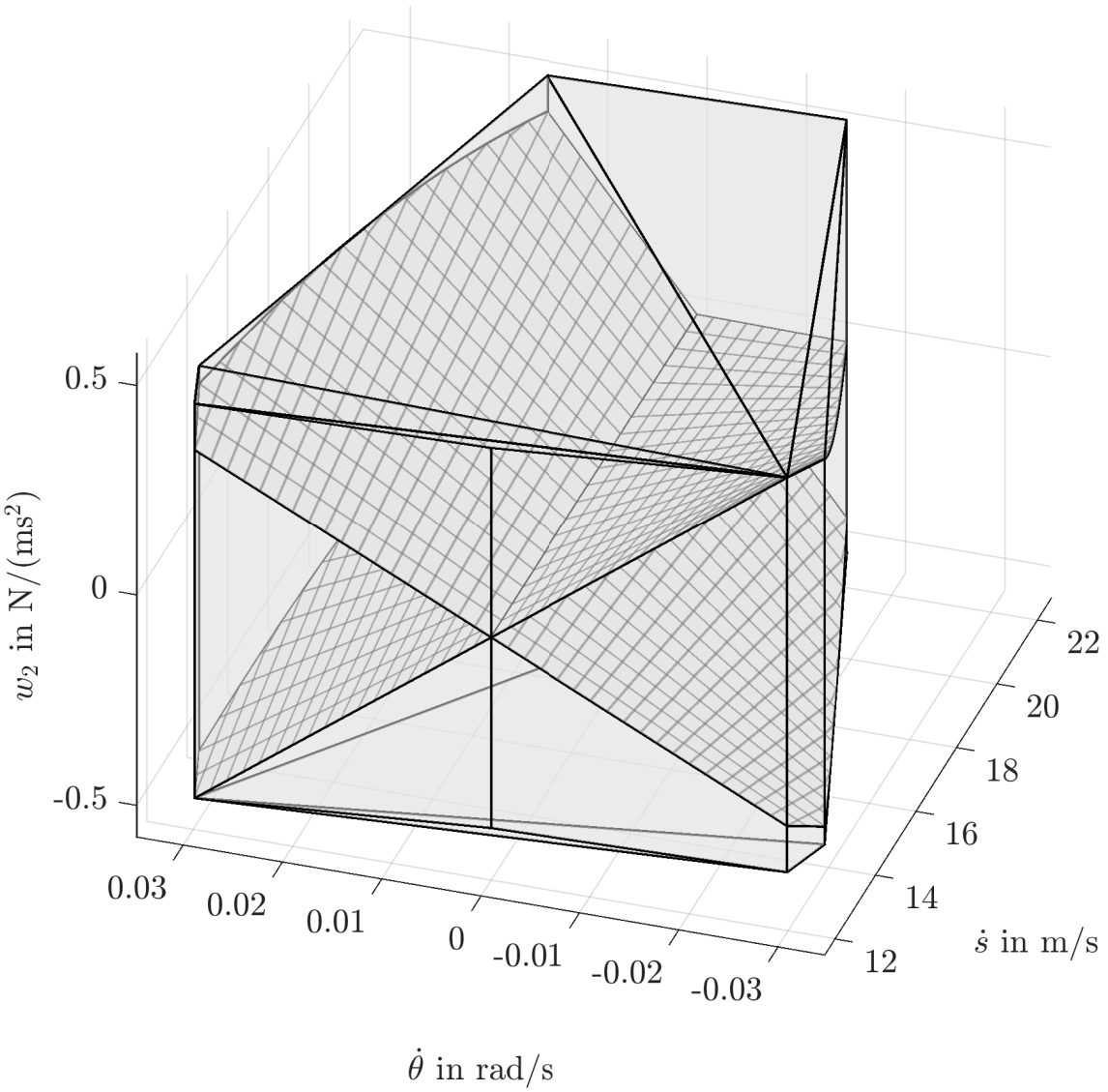


Figure A.7.: Component $f_{\text{param}[4]}$ (gridded; shown for extremal values of v_X) and a bounding polytope.

where c_1 and c_2 encode the maximum values:

$$c_2 := \max \frac{\bar{c}_Y(l_\mathrm{f}+l_\mathrm{r})}{\Theta} \cdot \left|\left|\begin{bmatrix} 0 & -\dot{\theta} & 1 & 0 \\ \dot{\theta} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_\mathrm{pos} \\ \dot{e}_\mathrm{pos} \end{bmatrix}\right|\right|$$

subject to

$$\bar{x} \in \mathcal{C}_{\bar{x}}, \quad \begin{bmatrix} e_\mathrm{pos} & \dot{e}_\mathrm{pos} \end{bmatrix}^\mathsf{T} P_\mathrm{pos} \begin{bmatrix} e_\mathrm{pos} \\ \dot{e}_\mathrm{pos} \end{bmatrix} \leq 1.$$

The solution is given by (A.20), where the maximum of the matrix norm expression can be computed explicitly for a given P_pos. Similarly,

$$c_1 := \max |\Delta a_\psi| + |\ddot{\theta}| + \left|\left|\begin{bmatrix} \cos e_\psi - 1 & \sin e_\psi - e_\psi \end{bmatrix}\right|\right| \cdot \left|\left|\begin{bmatrix} \frac{m}{\Theta} l_\mathrm{f} p_4 \\ -\frac{m}{\Theta} l_\mathrm{f} p_3 - \frac{\bar{c}_Y(l_\mathrm{f}+l_\mathrm{r})}{\Theta} p_6 \end{bmatrix}\right|\right| \cdots + \frac{m}{\Theta} l_\mathrm{f} \left|\left| \left(-K + \begin{bmatrix} \dot{\theta}^2 & \ddot{\theta} & 0 & 2\dot{\theta} \\ -\ddot{\theta} & \dot{\theta}^2 & -2\dot{\theta} & 0 \end{bmatrix} \right) \begin{bmatrix} e_\mathrm{pos} \\ \dot{e}_\mathrm{pos} \end{bmatrix} \right|\right|$$

subject to

$$\begin{bmatrix} e_\mathrm{pos} & \dot{e}_\mathrm{pos} \end{bmatrix}^\mathsf{T} P_\mathrm{pos} \begin{bmatrix} e_\mathrm{pos} \\ \dot{e}_\mathrm{pos} \end{bmatrix} \leq 1, \; p \in \mathcal{P}, \quad \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} \in \mathtt{proj}_{\dot{\theta},\ddot{\theta}} \mathcal{C}_{\bar{x}}, \; e_\psi \in \mathtt{proj}_{e_\psi} \mathcal{E}.$$

The single terms can be maximized separately and mostly without a problem, as for the low-dimensional expressions, even graphical solutions are possible. The exception is:

$$\max_{p \in \mathcal{P}} \left|\left|\begin{bmatrix} \frac{m}{\Theta} l_\mathrm{f} p_4 \\ -\frac{m}{\Theta} l_\mathrm{f} p_3 - \frac{\bar{c}_Y(l_\mathrm{f}+l_\mathrm{r})}{\Theta} p_6 \end{bmatrix}\right|\right|,$$

which is a non-convex optimization problem, but nonetheless solvable to global optimality by [63].

□

I. Parameter Values

Table A.2.: Parameter values used in planning and tracking example.

Param.	$v_{X,\min}$	$v_{X,\max}$	$\dot{v}_{X,\min}$	$\dot{v}_{X,\max}$	$\dot{v}_{Y,\min}$	$\dot{v}_{Y,\max}$	$\beta_\max$	$\delta_\min$
Value	$20\,\mathrm{km\,h^{-1}}$	$80\,\mathrm{km\,h^{-1}}$	$-5\,\mathrm{m\,s^{-2}}$	$5\,\mathrm{m\,s^{-2}}$	$-5\,\mathrm{m\,s^{-2}}$	$5\,\mathrm{m\,s^{-2}}$	$3\,\mathrm{deg}$	$-3\,\mathrm{deg}$
Param.	$v_{Y,\min}$	$v_{Y,\max}$	$\omega_\min$	$\omega_\max$	$s_{X,\min}$	$s_{X,\max}$	$\alpha_\max$	$\delta_\max$
Value	$-10\,\mathrm{km\,h^{-1}}$	$10\,\mathrm{km\,h^{-1}}$	$-20\,\mathrm{deg\,/s}$	$20\,\mathrm{deg\,/s}$	-0.1	0.1	$3\,\mathrm{deg}$	$3\,\mathrm{deg}$

List of Symbols

Functions

$\delta^H(\bullet_1, \bullet_2)$ Hausdorff-distance between two compact, non-empty sets $\bullet_1$ and $\bullet_2$

inv map assigning invariants to a phase of a hybrid automaton

ρ distance between a point and a set

Θ map assigning the set of inbound transitions to a phase of a hybrid automaton

$f : \mathbb{R}^{n_x} \times \mathcal{U} \times \texttt{proj}_w \mathcal{W} \to \mathbb{R}^{n_x}$ flow function of a hybrid automaton

$f_i : \mathbb{R}^{n_{x_i}} \times \mathbb{R}^{n_{u_i}} \times \mathbb{R}^{n_{w_i}} \to \mathbb{R}^{n_{x_i}}$ model of the dynamics of vehicle i

g map assigning a guard set to a transition

$g_{\mathcal{P}} : \mathbb{R}^{n_x} \to \mathbb{R}$ support function of a polytope $\mathcal{P}$

$k_i : \mathbb{R}^{n_{x_i}} \times \mathbb{R}^{\bar{n}_x} \to \mathbb{R}^{n_{u_i}}$ controller of vehicle i

General

$\bullet$ wildcard character

$\bullet_{[:,j]}$ jth column of matrix $\bullet$

$\bullet_{[\mathcal{I},:]}$ entries of all rows of matrix $\bullet$ addressed by index set $\mathcal{I}$ (ordered by magnitude of indices)

$\bullet_{[i,:]}$ ith row of matrix $\bullet$

$\bullet_{[i,j]}$ entry in the ith row, jth column of matrix $\bullet$

$\bullet_{[i]}$ ith entry of vector $\bullet$

$\partial\mathcal{P}$ boundary of polytope $\mathcal{P}$

Operators

$\texttt{convh}\,(\bullet)$ gives the convex hull of the vectors in set $\bullet$

$\texttt{diag}\,(\bullet_1, \bullet_2, \dots)$ diagonal matrix with diagonal entries $\bullet_1$, $\bullet_2$,...

$\texttt{diag}\,(\bullet)$ main diagonal entries of matrix $\bullet$

$\texttt{d}(\bullet)$ polar dual of a polyhedron $\bullet$

$\texttt{f}\,(\bullet)$ gives the set of outward-pointing unit normal vectors of the facets of polyhedron $\bullet$

$\texttt{int}(\bullet)$ gives the interior of polytope $\bullet$

$\oplus$ Minkowski sum

$\texttt{post}(\bullet)$ given controllable sets, it indicates to which set a set is controllable

$\texttt{pre}$ operator used for computation of controllable sets

$\texttt{pre}_\text{c}(\bullet)$ returns states controllable in one step to target $\bullet$ without phase transitions

$\texttt{pre}_\text{d}(\bullet)$ returns states controllable in one step to target $\bullet$ with phase transitions

$\texttt{proj}(\bullet)$ projection of set $\bullet$

$\times$ Cartesian product of two sets (if not used to indicate dimension of a matrix)

$\texttt{tri}\,(\bullet)$ triangulation of a finite set of points

$\texttt{vol}\,(\bullet)$ volume of simplex $\bullet$

$\texttt{v}\,(\bullet)$ gives the the set of vertices of polytope $\bullet$

Scalars and Physical Quantities

α_bss Side-slip angle

$\bar{p}_x(t)$ longitudinal reference position at time t

$\bar{p}_y(t)$ lateral reference position at time t

$\ddot{\theta}$ Yaw acceleration (reference trajectory)

$\ddot{s}$ Tangential acceleration (reference trajectory)

δ Front wheel steering angle

$\dot{\theta}$ Yaw rate (reference trajectory)

$\dot{s}$ Tangential velocity (reference trajectory)

$l_{x,\text{safe}}$ longitudinal safety distance between two vehicles

ω Yaw rate of a vehicle

ω_i weight of vertex i in Warren's algorithm

ψ Vehicle heading (global coordinate system)

$p_x^{(i)}(t)$ longitudinal position of vehicle i at time t

$p_y^{(i)}(t)$ lateral position of vehicle i at time t

Θ Moment of inertia (vertical vehicle axis; through center of gravity)

t_k, k quantities indicating discrete time instances

T_s sampling time

$u_{x,\min}$, $u_{x,\max}$ ($u_{y,\min}$, $u_{y,\max}$) bounds on admissible longitudinal (lateral) acceleration

$u_x^{(i)}$ longitudinal acceleration of vehicle i

$u_y^{(i)}$ lateral acceleration of vehicle i

$v_{X,\min}$, $v_{X,\max}$ ($v_{Y,\min}$, $v_{Y,\max}$) bounds on admissible longitudinal (lateral) velocities

$v_x^{(i)}$ longitudinal velocity of vehicle i

$v_y^{(i)}$ lateral velocity of vehicle i

a_ψ External yaw acceleration of a vehicle

c_Y Lateral tire stiffness

c_X Longitudinal tire stiffness

e_ω Tracking yaw rate error

e_ψ Tracking heading error

H_plan planning horizon

k_pos upper bound on the ∞-norm of the position deviation from a reference

l_r Distance from vehicle center of gravity to rear axle

l_f Distance from vehicle center of gravity to front axle

m Vehicle mass

n degree of continuity of a reference trajectory

p_x Vehicle position (1st direction of global coordinate system)

p_y Vehicle position (2nd direction of global coordinate system)

s Path coordinate (reference trajectory)

s_X Longitudinal tire slip

t_0, t_{plan} start and end time of a trajectory

t_{plan} Duration of a reference trajectory

v_Y Lateral velocity of a vehicle

v_X Longitudinal velocity of a vehicle

Sets

$\bar{\mathcal{P}}$ a shifted version of polyhedron $\mathcal{P}$

$\bar{\mathcal{X}}_{\text{safe}}$ a robust control invariant set used to derive $\mathcal{X}_{\text{safe}}$

$\mathtt{HA}$ a hybrid automaton

$\mathtt{M}$ a maneuver

$\mathbb{N}$ set of natural numbers

$\mathbb{R}$ set of real numbers

$\mathbb{T}$ time domain of a hybrid automaton

$\mathcal{A}$ inner approximation of a controllable set's projection on the continuous state space

$\mathcal{C}$ set of identifiers of cooperating vehicles

$\mathcal{E}_{\text{yaw}}$ Invariant set of the yaw error dynamics

$\mathcal{F}_i$ in Warren's algorithm: index set of incident vertices of facet i

$\mathcal{F}_i(t) \subset \mathbb{R}^2$ set of positions forbidden for vehicle i at time t

$\mathcal{K}_j$ set containing j-step robust controllable sets of a hybrid automaton

$\mathcal{N}$ set of identifiers of non-cooperating, leading vehicles

$\mathcal{P}$ a polyhedron

$\mathcal{Q}$ set of phases of a hybrid automaton

$\mathcal{S}$ a simplex

$\mathcal{T}$ controllable tube (temporal sequence of controllable sets)

$\mathcal{T}$ set of discrete transitions of a hybrid automaton

$\mathcal{U}$ compact set of admissible control inputs of a hybrid automaton

$\mathcal{U}_i$ admissible control inputs of vehicle i

$\mathcal{V}_i$ disturbance set of vehicle i

$\mathcal{W}$ compact, state-dependent set of possible disturbance inputs

$\mathcal{X}_0$, $\mathcal{X}_\mathrm{T}$ sets from a hybrid automaton's continuous state space denoting admissible initial and target states

$\mathcal{X}_\mathrm{safe}$ set of safe distances/velocities between two vehicles

$\mathcal{X}_i$ admissible states of vehicle i

$\tilde{\mathcal{T}}$ inner approximation of a controllable tube

S_i ith set of a triangulation, containing vertex indices

$T_\mathcal{P}(\bullet)$ set of points of tangency for direction $\bullet$ and polytope $\mathcal{P}$

V matrix whose rows contain vertices of a polyhedron

Vectors and Matrices

$0_{n\times m}$ n-by-m matrix of zeros

$1_{n\times m}$ n-by-m matrix of ones

$\bar{C}_\mathrm{pos}^{(i)}$ matrix which extracts position references for vehicle i from $\bar{x}_i$

$\bar{x}$ in Warren's algorithm: coordinate after shift of a polyhedron

$\bar{x}(\cdot)$ Reference trajectory

$\bar{x}_i(t) \in \mathbb{R}^{\bar{n}_x}$ reference value for the controller of vehicle i at time t

$\bar{z}$ reference value for z as used in planning problem

β vector of auxiliary binary variables

$\chi_i(t)$ state vector of vehicle i at time t

Δp positional safety margin

λ barycentric coordinates

μ Vector of vehicle control inputs

$\mu_i(t)$ disturbance vector of vehicle i at time t

$\nu_i(t)$ disturbance vector of vehicle i at time t

θ In part I: tuple of two phases, indicating a possible transition between them; in part II: reference heading angle

$\tilde{\mu}$ Artificial control input in feedback linearization

A, B, B_1, a determining quantities of affine flow function of a hybrid automaton

a_Y External lateral acceleration of a vehicle

a_X External longitudinal acceleration of a vehicle

C_1, D_1 weighting matrices of cost function of planning problem

$C_{\text{pos}}^{(i)}$ matrix which extracts vehicle i's position from its state vector

C_z matrix extracting the vector z from the state vector x

D a matrix

e State vector of the tracking error

e_{n} Tracking position error (normal direction)

e_{pos} Tracking position error state vector

e_{t} Tracking position error (tangential direction)

e_{yaw} Tracking yaw dynamics error state vector

E_i, $i = 1, 2, \ldots, 6$ matrices used in inequality representation of a hybrid automaton

K feedback matrix in tracking controller

q (q_0, q_{T}) vector with one-to-one correspondence to the phase of a hybrid automaton (initial and target phase)

u control input to the flow function of a hybrid automaton

w disturbance input to the flow function of a hybrid automaton

x vector from the continuous state space of a hybrid automaton

z vector of states relevant to the cost function of the planning problem

References

[1] J. Abedor, K. Nagpal, and K. Poolla, "A linear matrix inequality approach to peak-to-peak gain minimization," *International Journal of Robust and Nonlinear Control*, vol. 6, no. 9-10, pp. 899–927, 1996.

[2] J. Ackermann, *Robust control: the parameter space approach*. Springer Science & Business Media, 2012.

[3] A. P. Aguiar and J. P. Hespanha, "Trajectory-tracking and path-following of under-actuated autonomous vehicles with parametric modeling uncertainty," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1362–1379, 2007.

[4] A. Ailon, N. Berman, and S. Arogeti, "On controllability and trajectory tracking of a kinematic vehicle model," *Automatica*, vol. 41, no. 5, pp. 889–896, 2005.

[5] E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia, "Autonomous racing using linear parameter varying-model predictive control (LPV-MPC)," *Control Engineering Practice*, vol. 95, p. 104270, 2020.

[6] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[7] E. Asarin, T. Dang, and A. Girard, "Reachability analysis of nonlinear systems using conservative approximation," in *International Workshop on Hybrid Systems: Computation and Control*. Springer, 2003, pp. 20–35.

[8] H. Atoui, O. Sename, E. Alcala, and V. Puig, "Parameter varying approach for a combined (kinematic + dynamic) model of autonomous vehicles," in *Proceedings of the 21st IFAC World Congress*. IFAC, 2020, pp. 15 280 – 15 285.

[9] P. G. Backes, "Generalized compliant motion task description and execution within a complete telerobotic system," in *Proceedings of the IEEE International Conference on Systems Engineering*. IEEE, 1990, pp. 515–518.

[10] G. Bagschik, M. Nolte, S. Ernst, and M. Maurer, "A system's perspective towards an architecture framework for safe automated vehicles," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2438–2445.

[11] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469–483, 1996.

[12] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1998.

[13] A. Bemporad and C. Filippi, "An algorithm for approximate multiparametric convex programming," *Computational optimization and applications*, vol. 35, no. 1, pp. 87–108, 2006.

[14] A. Bemporad, C. Filippi, and F. D. Torrisi, "Inner and outer approximations of polytopes using boxes," *Computational Geometry*, vol. 27, no. 2, pp. 151–178, 2004.

[15] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[16] P. Bender, Ö. S. Tas, J. Ziegler, and C. Stiller, "The combinatorial aspect of motion planning: Maneuver variants in structured environments." in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2015, pp. 1386–1392.

[17] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific, Belmont, MA, 1995, vol. 1.

[18] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming*. Athena Scientific, 1996.

[19] T. Besselmann and M. Morari, "Hybrid parameter-varying model predictive control for autonomous vehicle steering," *European Journal of Control*, vol. 14, no. 5, pp. 418–431, 2008.

[20] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.

[21] M. Bieshaar, G. Reitberger, S. Zernetsch, B. Sick, E. Fuchs, and K. Doll, "Detecting intentions of vulnerable road users based on collective intelligence," in *Proceedings of the AAET*. ITS Automotive Nord, 2017, pp. 67–87.

[22] R. E. Bixby, "A brief history of linear and mixed-integer programming computation," *Documenta Mathematica*, vol. Optimization Stories, pp. 107–121, 2012.

[23] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer, 2008.

[24] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[25] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[26] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994, vol. 15.

[27] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[28] M. L. Brockman and M. Corless, "Quadratic boundedness of nominally linear systems," *International Journal of Control*, vol. 71, no. 6, pp. 1105–1117, 1998.

[29] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control*. Hemisphere, New York, 1975.

[30] C. Burger and M. Lauer, "Cooperative multiple vehicle trajectory planning using MIQP," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018, pp. 602–607.

[31] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2019.

[32] G. E. Collins and H. Hong, "Partial cylindrical algebraic decomposition for quantifier elimination," *Journal of Symbolic Computation*, vol. 12, no. 3, pp. 299–328, 1991.

[33] M. Corno, G. Panzani, F. Roselli, M. Giorelli, D. Azzolini, and S. M. Savaresi, "An LPV approach to autonomous vehicle path tracking in the presence of steering actuation nonlinearities," *IEEE Transactions on Control Systems Technology*, pp. 1–9, 2020.

[34] T. X. T. Dang, "Verification and synthesis of hybrid systems," Ph.D. dissertation, Institut National Polytechnique de Grenoble, 2000.

[35] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 279–286.

[36] S. Di Cairano and F. Borrelli, "Reference tracking with guaranteed error bound for constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 8, pp. 2245–2250, 2016.

[37] J. Ding, J. Sprinkle, S. S. Sastry, and C. J. Tomlin, "Reachability calculations for automated aerial refueling," in *Proceedings of the IEEE Conference on Decision and Control*. IEEE, 2008, pp. 3706–3712.

[38] E. Donges, "A conceptual framework for active safety in road traffic," *Vehicle System Dynamics*, vol. 32, no. 2-3, pp. 113–128, 1999.

[39] J. Du, J. Masters, and M. Barth, "Lane-level positioning for in-vehicle navigation and automated vehicle location (AVL) systems," in *Proceedings of the International Conference on Intelligent Transportation Systems*. IEEE, 2004, pp. 35–40.

[40] D. Dueri, B. Acikmese, M. Baldwin, and R. S. Erwin, "Finite-horizon controllability and reachability for deterministic and stochastic linear control systems with convex constraints," in *Proceedings of the American Control Conference*. IEEE, 2014, pp. 5016–5023.

[41] M. Egerstedt, X. Hu, and A. Stotsky, "Control of mobile platforms using a virtual vehicle approach," *IEEE Transactions on Automatic Control*, vol. 46, no. 11, pp. 1777–1782, 2001.

[42] J. Eilbrecht, M. Bieshaar, S. Zernetsch, K. Doll, B. Sick, and O. Stursberg, "Model-predictive planning for autonomous vehicles anticipating intentions of vulnerable road users by artificial neural networks," in *Proceedings of the Symposium Series on Computational Intelligence*. IEEE, November 2017, pp. 2869 – 2876.

[43] J. Eilbrecht, M. Jilg, and O. Stursberg, "Distributed H_2-optimized output feedback controller design using the ADMM," in *Proceedings of the 20th IFAC World Congress*. IFAC, 2017, pp. 10 389–10 394.

[44] J. Eilbrecht and O. Stursberg, "Auction-based cooperation of autonomous vehicles using mixed-integer planning," in *Proceedings of the AAET*. ITS Automotive Nord, February 2017, pp. 266–286.

[45] J. Eilbrecht and O. Stursberg, "Cooperative driving using a hierarchy of mixed-integer programming and tracking control," in *Proceedings of the IEEE Intelligent Vehicles Symposium*. IEEE, June 2017, pp. 673–678.

[46] J. Eilbrecht and O. Stursberg, "Optimization-based maneuver automata for cooperative trajectory planning of autonomous vehicles," in *Proceedings of the European Control Conference*, June 2018, pp. 82–88.

[47] J. Eilbrecht and O. Stursberg, "Sichere Trajektorienplanung für autonome Fahrzeuge unter Verwendung steuerbarer und erreichbarer Mengen (in German)," in *Proceedings of the AAET*. ITS Automotive Nord, February 2019, pp. 66–91.

[48] J. Eilbrecht and O. Stursberg, "Challenges of trajectory planning with integrator models on curved roads," in *Proceedings of the 21st IFAC World Congress*. IFAC, 2020, pp. 15 588–15 595.

[49] J. Eilbrecht and O. Stursberg, "Set-based scheduling for highway entry of autonomous vehicles," in *Proceedings of the 21st IFAC World Congress*. IFAC, 2020, pp. 15 396–15 403.

[50] P. Falcone, F. Borrelli, J. Asgari, H. Tseng, and D. Hrovat, "Low complexity MPC schemes for integrated vehicle dynamics control problems," in *Proceedings of the International Symposium on Advanced Vehicle Control*, 2008, pp. 875–880.

[51] P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H. E. Tseng, "A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems," in *Proceedings of the IEEE Conference on Decision and Control*. IEEE, 2007, pp. 2980–2985.

[52] M. S. Floater, "Generalized barycentric coordinates and applications," *Acta Numerica*, vol. 24, pp. 161–214, 2015.

[53] J. V. Frasch, A. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl, "An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles," in *Proceedings of the European Control Conference*. IEEE, 2013, pp. 4136–4141.

[54] E. Frazzoli, "Robust hybrid control for autonomous vehicle motion planning," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[55] E. Frazzoli, M. A. Dahleh, and E. Feron, "A hybrid control architecture for aggressive maneuvering of autonomous helicopters," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 3. IEEE, 1999, pp. 2471–2476.

[56] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler, "Spaceex: Scalable verification of hybrid systems," in *Proceedings of the International Conference on Computer Aided Verification*. Springer, 2011, pp. 379–395.

[57] S. Fuchshumer, K. Schlacher, and T. Rittenschober, "Nonlinear vehicle dynamics control-a flatness based approach," in *Proceedings of the IEEE Conference on Decision and Control*. IEEE, 2005, pp. 6492–6497.

[58] K. Fukuda, "Frequently asked questions in polyhedral computation," ETH, Zürich: https://www.cs.mcgill.ca/~fukuda/soft/polyfaq/node31.html (retrieved 2020-10-15), June 2004.

[59] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin, "Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice," in *Proceedings of the International Conference on Robotics and Automation*. IEEE, 2010, pp. 1649–1654.

[60] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, p. 65, 2010.

[61] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles." *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.

[62] A. Gray, Y. Gao, T. Lin, J. K. Hedrick, H. E. Tseng, and F. Borrelli, "Predictive control for agile semi-autonomous ground vehicles using motion primitives," in *Proceedings of the American Control Conference*. IEEE, 2012, pp. 4239–4244.

[63] Gurobi Optimization, Inc., "Gurobi optimizer," 2020, version 9.0.1.

[64] B. Gutjahr, L. Gröll, and M. Werling, "Lateral vehicle trajectory optimization using constrained linear time-varying MPC," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1586–1595, 2016.

[65] S. Herbert, "Safe real-world autonomy in uncertain and unstructured environments," Ph.D. dissertation, EECS Department, University of California, Berkeley, Aug 2020. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2020/EECS-2020-147.html

[66] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "FaSTrack: a modular framework for fast and guaranteed safe motion planning," in *Proceedings of the IEEE Conference on Decision and Control*. IEEE, 2017, pp. 1517–1522.

[67] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proceedings of the European Control Conference*, 2013, pp. 502–510.

[68] D. Heß, M. Althoff, and T. Sattel, "Comparison of trajectory tracking controllers for emergency situations," in *Proceedings of the IEEE Intelligent Vehicles Symposium*. IEEE, 2013, pp. 163–170.

[69] D. Heß, M. Althoff, and T. Sattel, "Formal verification of maneuver automata for parameterized motion primitives," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1474–1481.

[70] R. Hult, G. R. Campos, E. Steinmetz, L. Hammarstrand, P. Falcone, and H. Wymeersch, "Coordination of cooperative autonomous vehicles: Toward safer and more efficient road transportation," *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 74–84, 2016.

[71] J. hwan Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras, and K. Iagnemma, "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving," in *Proceedings of the American Control Conference*. IEEE, 2013, pp. 188–193.

[72] IBM, "IBM ILOG CPLEX optimization studio," 2019, version 12.10.

[73] A. Isidori, *Nonlinear Control Systems*. Springer, 1995.

[74] M. Jilg, *Hierachical and Cooperative Control of Complex Distributed Systems*. Kassel University Press GmbH, 2018.

[75] C. N. Jones and M. Morari, "Polytopic approximation of explicit model predictive controllers," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2542–2553, 2010.

[76] T. Ju, S. Schaefer, J. D. Warren, and M. Desbrun, "A geometric construction of coordinates for convex polyhedra using polar duals." in *Symposium on Geometry Processing*. Citeseer, 2005, pp. 181–186.

[77] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *The international journal of robotics research*, vol. 5, no. 3, pp. 72–89, 1986.

[78] A. Katriniok and D. Abel, "LTV-MPC approach for lateral vehicle guidance by front steering at the limits of vehicle dynamics," in *Proceedings of the IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 6828–6833.

[79] E. C. Kerrigan, "Robust constraint satisfaction: Invariant sets and predictive control," Ph.D. dissertation, University of Cambridge, 2000. [Online]. Available: http://hdl.handle.net/10044/1/4346

[80] H. K. Khalil, *Nonlinear systems*. Prentice Hall, 2002.

[81] D. Kontny and O. Stursberg, "Fast control using homotopy properties for obstacle-avoidance of systems with input constraints," in *Proceedings of the IEEE Conference on Computer Aided Control System Design*. IEEE, 2016, pp. 654–660.

[82] D. Kontny and O. Stursberg, "Fast optimizing control for non-convex state constraints using homotopy properties," in *Proceedings of the IEEE Conference on Decision and Control*. IEEE, 2016, pp. 4894–4900.

[83] D. Kontny and O. Stursberg, "Online adaption of motion paths to time-varying constraints using homotopies," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3331–3337, 2017.

[84] K. Kurzer, C. Zhou, and J. M. Zöllner, "Decentralized cooperative planning for automated vehicles with hierarchical Monte Carlo tree search," in *Proceedings of the IEEE Intelligent Vehicles Symposium*. IEEE, 2018, pp. 529–536.

[85] Y. Kuwata, "Trajectory planning for unmanned vehicles using robust receding horizon control," Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[86] Y. Kuwata, G. A. Fiore, J. Teo, E. Frazzoli, and J. P. How, "Motion planning for urban driving using RRT," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1681–1686.

[87] L. Lapierre and D. Soetanto, "Nonlinear path-following control of an UAV," *Ocean engineering*, vol. 34, no. 11-12, pp. 1734–1744, 2007.

[88] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[89] C. Le Guernic, "Reachability analysis of hybrid systems with linear continuous dynamics," Ph.D. dissertation, Université Joseph-Fourier - Grenoble I, 2009.

[90] S. Lefèvre, A. Carvalho, and F. Borrelli, "A learning-based framework for velocity control in autonomous driving," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 32–42, 2016.

[91] X. Li, X. Xu, and L. Zuo, "Reinforcement learning based overtaking decision-making for highway autonomous driving," in *Proceedings of the International Conference on Intelligent Control and Information Processing*. IEEE, 2015, pp. 336–342.

[92] Y. Li, X. Chen, and J. Mårtensson, "Linear time-varying model predictive control for automated vehicles: Feasibility and stability under emergency lane change," in *Proceedings of the 21st IFAC World Congress*. IFAC, 2020, pp. 15 928–15 933.

[93] D. J. N. Limebeer and A. V. Rao, "Faster, higher, and greener: Vehicular optimal control," *IEEE Control Systems*, vol. 35, no. 2, pp. 36–56, April 2015.

[94] A. Liniger, "Path planning and control for autonomous racing," Ph.D. dissertation, ETH Zürich, 2018.

[95] Z. Liu and O. Stursberg, "Optimal trajectory planning of hybrid systems by efficient MIQP encoding," in *Proceedings of the IEEE Conference on Decision and Control*. IEEE, 2018, pp. 1548–1553.

[96] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proceedings of the IEEE International Symposium on Computer Aided Control Systems Design*. IEEE, 2004, pp. 284–289.

[97] A. V. Lotov, V. A. Bushenkov, and G. K. Kamenev, *Interactive decision maps: Approximation and visualization of Pareto frontier*. Springer, 2004.

[98] J. Lunze and F. Lamnabhi-Lagarrigue, *Handbook of hybrid systems control: theory, tools, applications*. Cambridge University Press, 2009.

[99] A. Lyapunov, "On the problem of stability of motion," *Stability of Motion*, 1893.

[100] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.

[101] S. Manzinger, M. Leibold, and M. Althoff, "Driving strategy selection for cooperative vehicles using maneuver templates," in *Proceedings of the IEEE Intelligent Vehicles Symposium*. IEEE, 2017, pp. 647–654.

[102] Maplesoft, a division of Waterloo Maple Inc., "Maple," 2020.

[103] T. Marcucci and R. Tedrake, "Mixed-integer formulations for optimal control of piecewise-affine systems," in *Proceedings of the ACM International Conference on Hybrid Systems: Computation and Control*, 2019, pp. 230–239.

[104] L. Markolf, J. Eilbrecht, and O. Stursberg, "Trajectory planning for autonomous vehicles combining nonlinear optimal control and supervised learning," in *Proceedings of the 21st IFAC World Congress*. IFAC, 2020.

[105] R. Matthaei and M. Maurer, "Autonomous driving–a top-down-approach," *at-Automatisierungstechnik*, vol. 63, no. 3, pp. 155–167, 2015.

[106] M. Maurer, J. Christian Gerdes, B. Lenz, and H. Winner, *Autonomous driving: technical, legal and social aspects*. Springer Nature, 2016.

[107] G. P. McCormick, "Computability of global solutions to factorable nonconvex programs: Part I — convex underestimating problems," *Mathematical Programming*, vol. 10, no. 1, pp. 147–175, 1976.

[108] I. M. Mitchell, "Application of level set methods to control and reachability problems in continuous and hybrid systems." Ph.D. dissertation, Stanford University, 2002.

[109] F. Molinari, N. N. Anh, and L. Del Re, "Efficient mixed integer programming for autonomous overtaking," in *Proceedings of the American Control Conference*. IEEE, 2017, pp. 2303–2308.

[110] MOSEK ApS, "Mosek optimization suite 9.0," 2019.

[111] K. Natesan, D.-W. Gu, and I. Postlethwaite, "Design of linear parameter varying trajectory tracking controllers for an unmanned air vehicle," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 224, no. 4, pp. 395–402, 2010.

[112] J. Nilsson and J. Sjöberg, "Strategic decision making for automated driving on two-lane, one way roads using model predictive control," in *Proceedings of the IEEE Intelligent Vehicles Symposium*. IEEE, 2013, pp. 1253–1258.

[113] Y. Ohta, Y. Nagai, and L. Gong, "Beneath-beyond method and construction of Lyapunov functions," in *Proceedings of the International Symposium on Nonlinear Theory and its Applications*. IEICE, 1997, pp. 354–356.

[114] H. Pacejka, *Tire and vehicle dynamics*. Elsevier, 2005.

[115] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[116] G. J. Pappas, J. Lygeros, and D. N. Godbole, "Stabilization and tracking of feedback linearizable systems under input constraints," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 1. IEEE, 1995, pp. 596–601.

[117] J. Park, S. Karumanchi, and K. Iagnemma, "Homotopy-based divide-and-conquer strategy for optimal trajectory planning via mixed-integer programming," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1101–1115, 2015.

[118] D. Payton, "An architecture for reflexive autonomous vehicle control," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 1986, pp. 1838–1845.

[119] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Proceedings of the Conference on Advances in neural information processing systems*, 1989, pp. 305–313.

[120] X. Qian, F. Altché, P. Bender, C. Stiller, and A. de La Fortelle, "Optimal trajectory planning for autonomous driving integrating logical constraints: An MIQP perspective," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2016, pp. 205–210.

[121] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[122] B. Ranft and C. Stiller, "The role of machine vision for intelligent vehicles," *IEEE Transactions on Intelligent vehicles*, vol. 1, no. 1, pp. 8–19, 2016.

[123] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control : theory, computation, and design*, 2nd ed. Nob Hill Pub., 2017.

[124] C. Scherer, P. Gahinet, and M. Chilali, "Multiobjective output-feedback control via LMI optimization," *IEEE Transactions on Automatic Control*, vol. 42, no. 7, pp. 896–911, 1997.

[125] E. Schmitzberger, J.-L. Bouchet, M. Dufaut, D. Wolf, and R. Husson, "Capture of homotopy classes with probabilistic road map," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3. IEEE, 2002, pp. 2317–2322.

[126] T. Schouwenaars, "Safe trajectory planning of autonomous vehicles," Ph.D. dissertation, Massachusetts Institute of Technology, 2006.

[127] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *Proceedings of the European Control Conference*. IEEE, 2001, pp. 2603–2608.

[128] B. Schürmann, A. El-Guindy, and M. Althoff, "Closed-form expressions of convex combinations," in *Proceedings of the American Control Conference*. IEEE, 2016, pp. 2795–2801.

[129] B. Schürmann, D. Heß, J. Eilbrecht, O. Stursberg, F. Köster, and M. Althoff, "Ensuring drivability of planned motions using formal methods," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2017, pp. 1–8.

[130] D. Silver, J. A. Bagnell, and A. Stentz, "Learning autonomous driving styles and maneuvers from expert demonstration," in *Proceedings of the 13th International Symposium on Experimental Robotics*. Springer, 2013, pp. 371–386.

[131] S. Singh, A. Majumdar, J.-J. Slotine, and M. Pavone, "Robust online motion planning via contraction theory and convex optimization," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5883–5890.

[132] R. F. Stengel, "Stochastic optimal control: theory and application." *New York*, 1986.

[133] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[134] O. Stursberg and B. H. Krogh, "Efficient representation and computation of reachable sets for hybrid systems," in *Proceedings of the International Workshop on Hybrid Systems: Computation and Control*. Springer, 2003, pp. 482–497.

[135] O. Stursberg and S. Lohmann, "Synthesizing safe supervisory controllers for hybrid nonlinear systems," in *Proceedings of the 17th IMACS World Congress*, 2005.

[136] Ö. Ş. Taş, F. Kuhnt, J. M. Zöllner, and C. Stiller, "Functional system architectures towards fully automated driving," in *Proceedings of the Intelligent vehicles symposium (IV)*. IEEE, 2016, pp. 304–309.

[137] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback motion planning via sums-of-squares verification," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.

[138] The MathWorks Inc., "MATLAB version 9.4.0.813654 (r2018a)," Natick, Massachusetts, 2018.

[139] C. D. Toth, J. O'Rourke, and J. E. Goodman, *Handbook of discrete and computational geometry*. CRC press, 2017.

[140] P. Usoro, F. Schweppe, D. Wormley, and L. Gould, "Ellipsoidal set-theoretic control synthesis," *Journal of Dynamic Systems, Measurement, and Control*, pp. 331–336, December 1982.

[141] C. Vallon, Z. Ercan, A. Carvalho, and F. Borrelli, "A machine learning approach for personalized autonomous lane change initiation and control," in *Proceedings of the IEEE Intelligent Vehicles Symposium*. IEEE, 2017, pp. 1590–1595.

[142] E. Velenis and P. Tsiotras, "Optimal velocity profile generation for given acceleration limits: Theoretical analysis," in *Proceedings of the American Control Conference*. IEEE, 2005, pp. 1478–1483.

[143] M. P. Vitus, S. L. Waslander, and C. J. Tomlin, "Locally optimal decomposition for autonomous obstacle avoidance with the tunnel-MILP algorithm," in *Proceedings of the IEEE Conference on Decision and Control*. IEEE, 2008, pp. 540–545.

[144] E. L. Wachspress, "Barycentric coordinates for polytopes," *Computers & Mathematics with Applications*, vol. 61, no. 11, pp. 3319–3321, 2011.

[145] J. Wang, J. Liu, and N. Kato, "Networking and communications in autonomous driving: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1243–1274, 2018.

[146] J. Warren, "Barycentric coordinates for convex polytopes," *Advances in Computational Mathematics*, vol. 6, no. 1, pp. 97–108, 1996.

[147] J. Warren, S. Schaefer, A. N. Hirani, and M. Desbrun, "Barycentric coordinates for convex sets," *Advances in computational mathematics*, vol. 27, no. 3, pp. 319–338, 2007.

[148] M. Werling, L. Groll, and G. Bretthauer, "Invariant trajectory tracking with a full-size autonomous road vehicle," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 758–765, 2010.

[149] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenet frame," in *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 987–993.

[150] A. P. White, G. Zhu, and J. Choi, *Linear parameter-varying control for engineering applications*. Springer, 2013.

[151] H. Williams, *Model building in mathematical programming*. Wiley, 1990.

[152] Wolfram Research, "Wolfram Mathematica," 2020, version 10.2.

[153] L. A. Wolsey and G. L. Nemhauser, *Integer and Combinatorial Optimization*. John Wiley & Sons, 1999, vol. 55.

[154] Y. Xu, H. Zheng, W. Wu, and J. Wu, "Robust hierarchical model predictive control for trajectory tracking with obstacle avoidance," in *Proceedings of the 21st IFAC World Congress*. IFAC, 2020, pp. 15 954 – 15 959.

[155] B. Xue, Z. She, and A. Easwaran, "Under-approximating backward reachable sets by polytopes," in *Proceedings of the International Conference on Computer Aided Verification*. Springer, 2016, pp. 457–476.

[156] B. Xue, Z. She, and A. Easwaran, "Under-approximating backward reachable sets by semialgebraic sets," *IEEE Transactions on Automatic Control*, pp. 5185–5197, 2017.

[157] D. Yi, M. A. Goodrich, and K. D. Seppi, "Homotopy-aware RRT*: Toward human-robot topological path-planning," in *Proceedings of the 11th ACM/IEEE Conference on Human Robot Interaction*, 2016, pp. 279–286.

[158] X. Yin and A. Eckert, "A novel strategy for high-performance vehicle lateral motion control," in *Proceedings of the 21st IFAC World Congress*. IFAC, 2020, pp. 14 142–14 148.

[159] Y. Zhang and C. G. Cassandras, "Decentralized optimal control of connected automated vehicles at signal-free intersections including comfort-constrained turns and safety guarantees," *Automatica*, vol. 109, p. 108563, 2019.

[160] L. Zhao, H. Hao, and W. Zhang, "Extracting flexibility of heterogeneous deferrable loads via polytopic projection approximation," in *Proceedings of the IEEE Conference on Decision and Control*. IEEE, 2016, pp. 6651–6656.

[161] J. Zhen and D. Den Hertog, "Computing the maximum volume inscribed ellipsoid of a polytopic projection," *INFORMS Journal on Computing*, vol. 30, no. 1, pp. 31–42, 2017.

[162] G. M. Ziegler, *Lectures on polytopes*. Springer Science & Business Media, 2012, vol. 152.

[163] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for Bertha — A local, continuous method," in *Proceedings of the IEEE Intelligent Vehicles Symposium*. IEEE, 2014, pp. 450–457.