Czech Technical University in Prague

Faculty of Electrical Engineering

Department of Control Engineering

Master's Thesis

Trajectory Tracking for Autonomous Vehicles

Author: **Jan Filip**

Supervisor: **Martin Vajnar**

Academic Year: **2017/2018**

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Filip Jan** Personal ID number: **406516**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Control Engineering**

Study program: **Cybernetics and Robotics**

Branch of study: **Cybernetics and Robotics**

II. Master's thesis details

Master's thesis title in English:

Trajectory Tracking for Autonomous Vehicles

Master's thesis title in Czech:

Sledování trajektorie pro autonomní vozidla

Guidelines:

1. Research the state-of-the art approaches to trajectory tracking for autonomous and semi-autonomous road vehicles.
2. Perform a survey of available high-fidelity vehicle dynamics simulators for model-in-the-loop verification of the controller.
3. Develop and verify a mathematical model that captures important features of the kinematics and dynamics of the vehicle for design of trajectory tracking controller.
4. Create a simulation environment for design and verification of the trajectory tracking controller. Use model-based design software, such as Matlab & Simulink.
5. Design a control system for longitudinal and lateral trajectory tracking that can steer the vehicle along a predefined trajectory.
6. Verify the controller experimentally.

Bibliography / sources:

[1] Rajamani, R., Vehicle Dynamics and Control, Springer US, ISBN 978-1-4614-1432-2
[2] Kiencke, U. and Nielsen, L., Automotive Control Systems for Engine, Driveline and Vehicle, SAE International, ISBN 0-7680-0505-1,2000
[3] Schramm, D., Hiller, M. and Bardini, R., Vehicle Dynamics: Modeling and Simulation, Springer-Verlag Berlin Heidelberg, ISBN 978-94-017-8532-7

Name and workplace of master's thesis supervisor:

Ing. Martin Vajnar, CIIRC, ČVUT v Praze

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **15.01.2018** Deadline for master's thesis submission: **25.05.2018**

Assignment valid until: **30.09.2019**

Ing. Martin Vajnar Supervisor's signature	prof. Ing. Michael Šebek, DrSc. Head of department's signature	prof. Ing. Pavel Ripka, CSc. Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt	Student's signature

Abstract

Development of autonomous cars is shaping the future of transportation. High-performance motion controllers are necessary to deploy the technology under a wide range of driving conditions. This thesis addresses trajectory tracking for autonomous vehicles, with the goal of developing a racing controller. We factor the task into the subproblems of vehicle dynamics modeling and model-based control design.

In the first part, we review the single-track models of vehicle motion, analyze their properties, such as response variation with speed, and limitations, such as prediction mismatch of the linear model under high lateral acceleration.

In the second part, we decompose the control problem into independent longitudinal and lateral controllers. The core contribution of the thesis is the formulation of the tracking controllers under the servomechanism problem framework, allowing a unified design approach using state-space methods. In particular, we compared the performance of reactive (LQR) and predictive (MPC) control strategies, and analyzed the structure of the unconstrained MPC, drawing the link between the works based on the classical control design and the recent optimization-based approaches.

The last part presents the test results achieved on a handling track in a vehicle dynamics simulator. We designed a minimum-time trajectory to operate the vehicle at the handling limits, demonstrated the functionality of the proposed solution and the peak performance achievable using the traditional decoupled control architecture, linear models, and constrained predictive controllers.

Keywords: vehicle dynamics, steering control, cruise control, autonomous vehicle, trajectory tracking, servomechanism, model predictive control, autonomous racing.

Abstrakt

Rozvoj autonomních vozidel utváří budoucnost dopravy. Přesné a spolehlivé řízení pohybu je nutné pro nasazení samořiditelných vozidel v rozličných jízdních podmínkách. Tato práce se zabývá návrhem sledování trajektorie pro samořiditelná vozidla. Cílem práce je vyvinout kontrolér pro závodní aplikace. Úloha je rozdělena na modelování jízdní dynamiky a následný návrh řízení založený na modelu.

První část práce nabízí přehled jednostopých modelů vozidla, analýzu jejich chování, zejména závislost odezvy na rychlosti, a omezení linearizovaných modelů, zejména nepřesnost předpovědi chování při vyšších hodnotách bočního zrychlení.

V druhé části je návrh řízení rozdělen na nezávislé řízení podélné a příčné dynamiky vozidla. Hlavním přínosem práce je formulace sledování pomocí problému servomechanismu, který sjednocuje návrhu řízení a umožňuje použití stavových metod. Dalším přínosem je srovnání reaktivního (LQR) a prediktivního (MPC) řízení pro danou aplikaci a rozbor zpětnovazební struktury prediktivního řízení bez omezení, která objasňuje souvislosti mezi staršími přístupy, založenými na klasické teorii řízení, a přístupy nedávnými, založenými na optimalizaci.

Závěrečná část prezentuje výsledky dosažené na simulovaném závodním okruhu. Pro experimentální ověření byla navržena trajektorie průjezdu okruhem v minimálním čase, na hranici jízdních možností vozidla. Výsledky dosažené v těchto podmínkách prokazují funkčnost řešení a limity dosažitelného chování pomocí oddělené řídicí architektury založené na prediktivních kontrolérech s omezeními.

Klíčová slova: dynamika vozidla, autonomní vozidlo, sledování trajektorie, servomechanismus, prediktivní řízení.

Declaration

I hereby declare that I wrote the presented thesis on my own and that I cited all the used information sources in compliance with the Methodical instructions about the ethical principles for writing an academic thesis.

Prague, May 25, 2018

Author's signature

Acknowledgement

First, I would like to express my gratitude to all of my former lecturers at the Department of Control Engineering of Czech Technical University in Prague for teaching me the theoretical foundations on which this thesis could expand.

Second, I am profoundly grateful to the Czech Technical University for giving me the opportunity and financial support to spend one and a half year of my studies abroad, gaining an invaluable academical, practical and cultural experience. In this regard, I would like to appreciate Nadání Josefa, Marie a Zdenky Hlávkových for the generous support of my stays abroad.

Third, I would like to thank Martin Vajnar for advising and supportive supervision of my work. My sincere thanks also go to Tomáš Haniš and Martin Grossman of Porsche Engineering Services Prague for extensive consulting on several occasions, practical insights, and for bringing the theoretically-compelling topic of the thesis closer to a possible industrial application.

Last, yet the most, I would like to thank my family as without their continuous support of my studies this journey would not have been possible.

Contents

Acronyms

ACC	Adaptive Cruise Control.
ASM	Active Set Method.
DARPA	Defense Advanced Research Projects Agency.
FIR	Finite Impulse Response.
GNU LGPL	GNU Lesser General Public License.
GPS	Global Positioning System.
IMP	Internal Model Principle.
IMU	Inertial Measurement Unit.
ISO	International Organization for Standardization.
LPV	Linear Parameter-Varying.
LQ	Linear Quadratic.
LQR	Linear Quadratic Regulator.
LTI	Linear Time-Invariant.
LTV	Linear Time-Varying.
MIMO	Multiple-input Multiple-output.
MPC	Model Predictive Control.
PI	Proportional Integral.
PID	Proportional, Integral and Derivative.
QP	Quadratic Programming.
RC	Radio Controlled.
RMS	Root Mean Square.
SAE	Society of Automotive Engineers.
SISO	Single-input Single-output.
SVD	Singular Value Decomposition.

1 | Introduction

The topic of this thesis is model-based design of a trajectory tracking controller for autonomous road vehicles. We focus specifically on the vehicle dynamics involved in the task, going beyond the typical parking and lane keeping applications into the domain of autonomous car racing.

1.1 Motivation

The motivation for developing the controller is found both in the academia and industry. Competitions such as the Formula Student Driverless [1] or the downscaled F1/10 Autonomous Racing Competition [2] emerged recently to foster the academic research in autonomous driving. One of the aims of the thesis is to provide a case study of a vehicle motion controller design for the CTU teams participating in the competitions mentioned above. Autonomous racing may soon become a part of the motorsport mainstream. In 2016, Roborace [3, 4] was announced as a supporting series of the FIA Formula E, aiming to be the first international track championship for autonomously driven cars. In the past, many of the technological advances were adopted from racing onto production cars. The new racing competitions pave the way for further research and development in the areas of autonomous driving, networking, and electric drives.

The thesis was started in collaboration with Porsche Engineering Services, Prague, sharing the interest to adopt the controller for automation of vehicle handling tests. But possible industrial applications surpass the racing tracks. With the progress in sensing, perception algorithms, and vehicle localization, future driver assistance systems can extend beyond vehicle stabilization to offer path tracking. Knowledge of the vehicle position and the desired path can be used to keep the vehicle on the road in emergency situations and reduce the number of road accidents. Control performance of a virtual racing driver can enable the operation of fully autonomous vehicles during worsened weather conditions and reduced adhesion on wet or icy roads.

1.2 Problem Definition

In mobile robotics, the terms path and trajectory are distinguished. A path is defined as a time-independent ordered sequence of positions (or poses) to be traversed with some tolerance. A trajectory additionally includes the speed (and acceleration) at which the path shall be traversed and hence is parametrized in time. Trajectory defines both the positional and the velocity reference, the complete rigid body state. A trajectory is dynamically feasible if the system can generate such state response while satisfying constraints of control inputs and states.

The primary task of the thesis is to design a vehicle motion controller to track a prescribed feasible trajectory at the vehicle handling limits. The secondary task is to design the trajectory based on a given fixed path and a parametrization of the vehicle handling limits. The difficulty of the problem arises from the nonlinearity of the vehicle motion and uncertainties in modeling the tire-road interaction. Longitudinal and lateral vehicle motion dynamics are coupled and require a trajectory to be designed for decoupling and possible linearization. Even when linearized, the control problem remains velocity dependent and thus time-varying.

1.3 Related work

The first milestone in autonomous car racing was the Defense Advanced Research Projects Agency (DARPA) Grand Challenge in 2005, consisting of a 212 km off-road course. Out of the 23 finalists, only five vehicles completed the course. Steering control of the winning robot Stanley [5], created by the Stanford University's team, was based on a kinematic motion model and classical control theory. No information on the future path was used and the controller was responding only to the current tracking errors. The other two medalists, Sandstorm and H1ghlander [6] of the Carnegie Mellon University, relied on a modification of the heuristic-based pure-pursuit path following algorithm [7].

In 2005, the problem of automatic vehicle steering control was analyzed systematically in [8] using a dynamic model of vehicle motion with a nonlinear model of tire force generation. The control of lane change maneuvers for obstacle avoidance on straight roads was formulated using Model Predictive Control (MPC) and started a series of articles applying the MPC approach to traction [9], steering [10], lateral stabilization [11] or lanekeeping [12] control problems. The MPC framework is well-equipped for including the available information about the future path, augmenting the feedback controller with an optimal feedforward input.

In 2009, BMW implemented the TrackTrainer [13] to teach driving at the friction limits. The car was capable of repeating a racing course recorded by a professional racing driver. However, the application was limited by the need of prerecorded data and included no mean of adjusting the speed profile to changes in road surface conditions. The results played an essential role in the successive research of automated highway driving [14].

This thesis was inspired by [15], describing the controllers implemented for Shelly, Stanford's Audi TTS, which completed the first autonomous run up Pikes Peak in Colorado. The control problem was decoupled by designing a minimum-time trajectory, separating the longitudinal and lateral motion dynamics. However, the controllers were designed using classical control methods with multiple control loops in parallel, limiting the performance achievable by repetitive tuning.

In [16] a clothoid-based model predictive controller was designed for Scania trucks. The subsystem structure similar to [15] and use of MPC inspired the research in this work, however, neither the clothoid-based path representation nor the kinematic model was found suitable for the race track driving application. The last major source of inspiration was found in [17], where a lateral motion controller was designed as a constrained linear time-varying MPC controller based on a kinematic vehicle model.

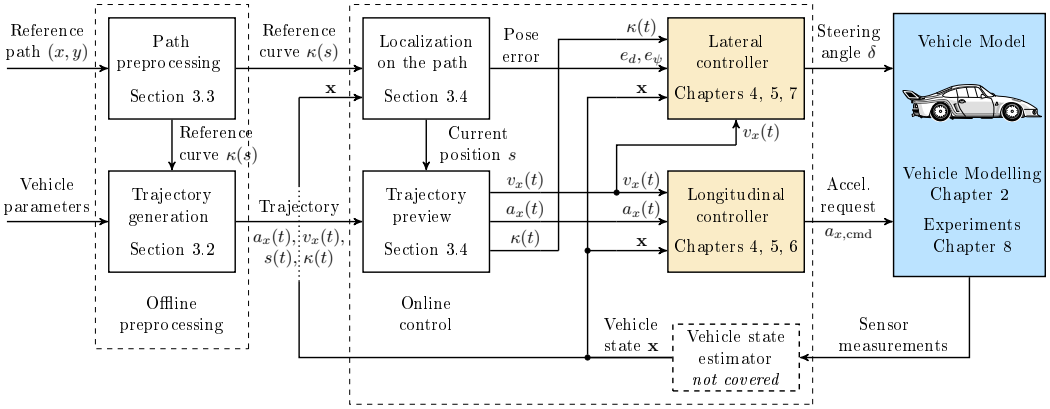


Figure 1.1: System architecture, main topics of the thesis highlighted

1.4 Contributions

The problem was solved by designing a decoupling trajectory and decomposing the problem into separated lateral and longitudinal control similarly as in [15] and [16]. However, the controller design extends the work of [17] using the model predictive control methodology.

The problem of trajectory tracking was formalized as a servomechanism problem and described using a state-space model. The structure allowed a unified design approach for both the lateral and longitudinal controllers, using both state-space and model predictive control methods. Our approach solved the design task, and moreover allowed a further analysis. Structure and performance of reactive (state feedback) and predictive control strategy were compared. Moreover, the approach used in this thesis allowed a broader comparison of the controller structures, drawing a link between the older works based on classical control design methods [5, 15] and the recent optimization-based approaches [17].

Finally, the performance of the controller was verified in a vehicle dynamics simulator, in the scenario of driving the Nardo handling track. A minimum-time trajectory was designed to operate the vehicle at the handling limits to demonstrate the peak performance achievable using the traditional decoupled control architecture, linear models, and predictive controllers with constraints.

1.5 Outline

The main topic of the thesis is model-based control design. System architecture shown in Fig. 1.1 captures the outline. Chapter 2 describes vehicle dynamics modeling in the scope of trajectory tracking. The subsequent control design is based on linear models and state-space methods. Basic principles of race car driving and minimum-time trajectory generation is covered in Chapter 3, together with other components required for implementation of a closed-loop trajectory tracking experiment. The servomechanism state-space structure used for both the lateral and longitudinal tracking controllers is introduced in Chapter 4. The model-predictive formulation of the tracking problem is described in Chapter 5. Longitudinal velocity controller is designed in Chapter 6 and lateral steering controller in Chapter 7. Finally, Chapter 8 presents the experimental results, consisting of simulated handling track driving. The thesis is concluded discussing the achieved results and topics opened for future work in Chapter 9.

2 | Vehicle Modeling

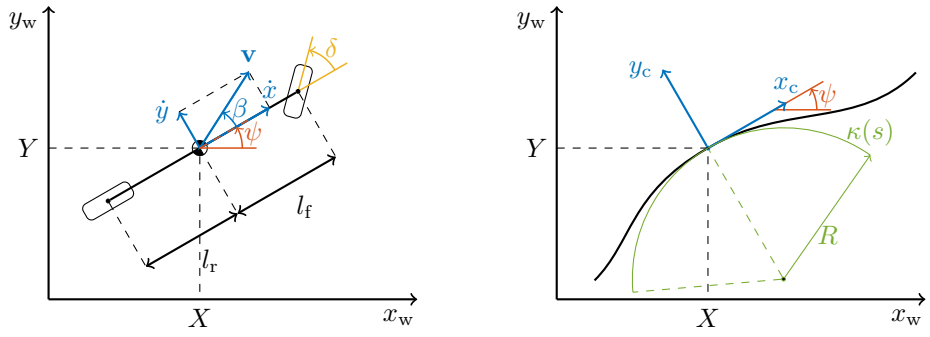
This chapter introduces simplified mathematical models of vehicle motion, which become essential in the later chapters for model-based controller design using model predictive control. In Section 2.2, description of a kinematic bicycle model is given, representing the basic constraints of a car-like motion. Similarly, a differential model of a curvature-continuous curve is described. In Section 2.3, a dynamic single-track planar model of vehicle motion with tire-road force interaction is introduced. The model is developed neglecting the suspension kinematics, body roll, load transfer and the track width with respect to the turn radius. However, the model captures the main dynamic properties for trajectory tracking while being tractable for analysis and controller design. A key part of the dynamic model is the description of the tire-road force interaction. A brief overview of tire modeling and its influence on vehicle handling is given in Section 2.4.

A commercial vehicle dynamics simulator *IPG Carmaker* was used to obtain representative test data, instead of carrying out highly demanding testing on a real full-scale vehicle with the expensive hardware for data acquisition. Assuming that the vehicle model provided in IPG Carmaker captures the behavior of a real vehicle with *high fidelity* if appropriately parametrized, its responses can serve as the reference for modeling. Although the vehicle model provided with IPG Carmaker could not have been validated due to the unavailability of a full-scale vehicle test platform, the behavior observed in simulations is in correspondence with experimental data published in the literature [18], [19, Sec. 6.15], [20, Sec. 14.1]. Therefore, the modeling attempts to capture the simulated behavior using lower-order (and possibly linear) models. Control algorithms are validated on the high-fidelity vehicle model of IPG Carmaker, using model-in-the-loop validation, an approach often utilized by vehicle manufacturers. A survey of available vehicle dynamics simulators is listed in Appendix A and the used parametrization of the vehicle model in Appendix B.

Validation in Section 2.5 shows that the simplified model captures the essential properties for trajectory tracking with a real vehicle. Section 2.6 analyzes the properties of the linear model of lateral motion dynamics, in particular, the response variation with the longitudinal velocity and model discretization. Section 2.7 evaluates the ability of the simplified models to generate open loop predictions of the high-fidelity model response. Finally, Section 2.8 describes separate modeling of the longitudinal dynamics.

2.1 Coordinate Frame Convention

Several standards of defining coordinate frames exist in the automotive industry. The main two standards are Society of Automotive Engineers (SAE) convention and International Organization for Standardization (ISO) convention. Models derived in this thesis are planar. Therefore, the definition of the reference frame in the



(a) kinematic bicycle model (b) curve-fixed coordinate frame

Figure 2.1: Coordinate frame convention used in kinematic models

horizontal plane is of concern. The SAE sign convention [21] defines x-axis pointing forward, y-axis pointing right, z-axis pointing downward and rotation positive in the clockwise direction. The ISO sign convention [22, 23] defines x-axis (longitudinal) pointing forward, y-axis (lateral) pointing left, z-axis (normal) pointing upward and rotation positive in the counter-clockwise direction. The SAE convention is an American National Standard while the ISO standard is commonly used in Europe. Moreover, the ISO frame convention is equivalent to the frame commonly used in mobile robotics [24, 25]. Therefore, the ISO convention is used in this thesis. The only exception to the ISO convention is the definition of tire slip angle, whose sign is changed so that positive tire slip angle induces positive lateral tire force. The SAE and ISO coordinate frames, each with the two possible tire slip sign conventions, are compared in [26, Appendix 1], from which we use the *adapted* ISO option.

2.2 Kinematic Model of Vehicle Motion

This section introduces a kinematic model of vehicle motion based on the steering geometry, which is commonly used for control of maneuvers at low speeds or low levels of lateral acceleration. The presented kinematic model is based on [27]. The key properties of the model are also summarized in [28], comparing the kinematic and dynamic (force-based) models in the context of model-predictive trajectory tracking.

2.2.1 Kinematic Bicycle Model

The key modeling assumption is that the vehicle moves with none of the wheels slipping sideways, meaning that the velocity vector of each wheel is aligned with the longitudinal axis of the wheel. Hence, the motion of the kinematic model is constrained by non-holonomic constraints. This assumption is valid when the lateral acceleration is low, and the lateral force exerted by each tire is negligible. During steady-state turning, the wheels are moving along concentric circular paths, aligned with the circle tangents and with no lateral slip. Therefore, it is possible to represent the two left and right wheels by a single wheel located in the center of the axle. Vehicle motion is assumed planar, the mass of the vehicle is modeled as a point mass located in the center of mass. The coordinate frame convention used in the kinematic model is shown in Fig. 2.1a.

The nonlinear equations of motion are

$$\dot{X} = v\cos(\psi + \beta), \tag{2.1a}$$
$$\dot{Y} = v\sin(\psi + \beta), \tag{2.1b}$$
$$\dot{\psi} = \frac{v\cos(\beta)}{l_\mathrm{f} + l_\mathrm{r}}\tan(\delta), \tag{2.1c}$$
$$\dot{v} = a, \tag{2.1d}$$
$$\beta = \tan^{-1}\left(\frac{l_\mathrm{r}}{l_\mathrm{f} + lr}\tan(\delta)\right), \tag{2.1e}$$

where X, Y are the vehicle coordinates in the global planar reference frame, ψ is the vehicle heading angle, $v = \|\mathbf{v}\|$ is the magnitude of the velocity vector of the center of mass, β is the vehicle side slip angle, the angle between vehicle longitudinal axis and the direction of the velocity vector of the center of mass, l_f, resp. l_r, is the distance between the center of mass and the front, resp. the rear, axle. Only front wheel steering is modeled and δ denotes the steering angle of the front wheel.

The kinematic model describes vehicle motion without considering the forces affecting it. Advantages of using the kinematic model for control design are twofold. First, the model can be used for motion control at low speeds, including standstill. As will be shown later in Section 2.3, any tire model becomes singular at low speeds due to vehicle velocity in the denominator of the wheel slip angle. Secondly, the model is described by a smaller number of parameters, which facilitates system identification and application of the controller to different platforms. Only the wheelbase and the position of the center of mass, described by l_f and l_r, need to be determined.

The model is converted to a Linear Parameter-Varying (LPV) form for the purpose of lateral motion control design, by considering the longitudinal velocity dynamics decoupled and treating the velocity v as a time-varying parameter v_x instead. Goniometric functions are approximated using small-angle assumption: $\sin(x) \approx x$, $\cos(x) \approx 1$, $\tan(x) \approx x$, because the steering angle δ is typically below $|\delta| \leq 20°$ during driving (excluding parking maneuvers). Equations of motion simplify to

$$\dot{\psi} = \frac{v_x}{L}\tan(\delta) = v_x\kappa \approx \frac{v_x}{L}\delta, \tag{2.2a}$$
$$\beta = \frac{l_\mathrm{r}}{L}\delta, \tag{2.2b}$$

where $L = l_\mathrm{f} + l_\mathrm{r}$ is the wheelbase, and $\kappa = \frac{1}{R}$ is the instantaneous curvature of motion, assuming the vehicle is moving along a circle of radius R. The small-angle approximation is valid for sufficiently large turn radius, meaning that $R \gg L$.

2.2.2 Model of a Curve

A curve can be described using its curvature as a function of the distance traveled along the curve, the arc length s. The coordinate frame used for the differential description of a curve is shown in Fig. 2.1b. The kinematic bicycle model of Eq. (2.2) represents a geometric curve. Rewriting Eq. (2.2a) using the chain rule

$$\dot{\psi} = \frac{\mathrm{d}\psi}{\mathrm{d}t} = \frac{\mathrm{d}\psi}{\mathrm{d}s}\frac{\mathrm{d}s}{\mathrm{d}t} = \kappa\dot{s}, \tag{2.3}$$

the velocity of the motion along the curve $\dot{s}$ is the time derivative of the arc length s. Curvature κ is the derivative of the curve tangential heading ψ with respect to the arc length s.

A curve used commonly in road design is a clothoid, a spiral whose curvature is an affine function of its arc length [27, 29]. The path must have continuous curvature to be kinematically feasible. While not imposed by Eq. (2.2) directly, in practice steering angle rate $\dot{\delta}$ (slew rate of the steering servo) is limited and the steering angle cannot be changed instantaneously while moving along the curve. Therefore, curvature continuity is an essential property of a curve representing the desired path of the vehicle. The motion of a particle along a curvature-continuous curve can be described by

$$\dot{\kappa} = u, \tag{2.4a}$$
$$\dot{\psi} = \kappa v_{\mathrm{ref}}, \tag{2.4b}$$

where input u is the rate of curvature change and v_{ref} is a time-varying parameter representing the tangential velocity of the motion along the curve $v_{\mathrm{ref}} = \dot{s}$. This differential representation of a curve is equivalent to the linearized kinematic bicycle model. The Cartesian coordinates of the curve can be recovered by integration of Eq. (2.1a)–(2.1b) with $\beta = 0$.

2.3 Dynamic Model of Vehicle Motion

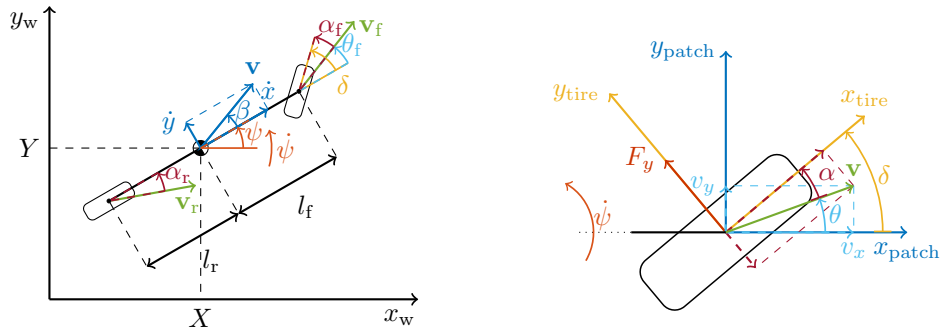
This section introduces a mathematical model of vehicle motion based on force interactions. The aim is to obtain a model which covers a larger envelope in terms of lateral acceleration than the kinematic model of Section 2.2. At a constant steering angle, the turning radius of a car typically changes with speed. This behavior is important for path tracking but is not captured in the kinematic model. The dynamic model presented in this section improves path prediction accuracy and is suitable for controller design using linear system methods.

As in the previous section, the motion of the vehicle is assumed planar; the vehicle is assumed to be a rigid body with mass m and moment of inertia I_z with respect to the axis normal to the horizontal plane at the center of mass. Since tires are the only part in contact with the road surface during normal operation, modeling tire forces is the essential part of a vehicle motion model. As in Section 2.2 we restrict ourselves to a single-track model, with the left and right tires lumped into one, representing the whole axle. By doing so, the roll dynamics of the suspension and lateral weight transfer are neglected, and a large turning radius is assumed, such that the left and right tires undergo approximately the same lateral deflection as the virtual tire in the middle. The coordinate frame and angular variables used in the dynamic bicycle model are shown in Fig. 2.2a.

A rigid body constrained to the planar movement has three degrees of freedom and its motion is governed by the following Newton-Euler equations

$$\ddot{x} = m^{-1}\left(F_{x,\mathrm{r}} + F_{x,\mathrm{f}}\cos(\delta) - F_{y,\mathrm{f}}\sin(\delta)\right) + \dot{\psi}\dot{y}, \tag{2.5a}$$
$$\ddot{y} = m^{-1}\left(F_{y,\mathrm{r}} + F_{x,\mathrm{f}}\sin(\delta) + F_{y,\mathrm{f}}\cos(\delta)\right) - \dot{\psi}\dot{x}, \tag{2.5b}$$
$$\ddot{\psi} = I_z^{-1}\left(l_{\mathrm{f}}F_{x,\mathrm{f}}\sin(\delta) + l_{\mathrm{f}}F_{y,\mathrm{f}}\cos(\delta) - l_{\mathrm{r}}F_{y,\mathrm{r}}\right), \tag{2.5c}$$

where $\dot{x}, \dot{y}$ are components of the velocity in the body-fixed coordinate frame and $\dot{\psi}$ is the yaw rate, the angular velocity of the rigid body with respect to the normal axis at the center of mass. Only the front wheel is assumed steered with δ denoting the steering angle. Tire forces are modeled and expressed in the respective tire-fixed coordinate frames, with the convention depicted in Fig. 2.2b. Therefore, the



(a) dynamic bicycle model (b) tire coordinate frames

Figure 2.2: Coordinate frame convention used in the dynamic model

forces related to the front steered wheel are transformed to the body-fixed coordinate frame by rotation of δ about the normal axis at the tire point of contact. The same procedure can be used for modeling steered rear wheel or incorporating all four tires in the twin-track model. The position of the vehicle center of mass in the global coordinate frame is obtained by integrating

$$\dot{X} = \dot{x}\cos(\psi) - \dot{y}\sin(\psi) = v\cos(\psi + \beta), \tag{2.6a}$$

$$\dot{Y} = \dot{x}\sin(\psi) + \dot{y}\cos(\psi) = v\sin(\psi + \beta), \tag{2.6b}$$

where

$$v = \sqrt{\dot{x}^2 + \dot{y}^2} \tag{2.7}$$

is the magnitude of the velocity vector at the vehicle's center of mass and

$$\beta = \arctan\left(\frac{\dot{y}}{\dot{x}}\right) \tag{2.8}$$

is the vehicle side slip angle, the direction of the velocity vector at the center of mass with respect to the body-fixed coordinate frame. To obtain the yaw angle, the heading of the vehicle, Eq. (2.5c) is integrated twice. From Eq. (2.5b), the lateral acceleration at the center of mass of the vehicle is given as

$$a_y = \ddot{y} + \dot{\psi}\dot{x}. \tag{2.9}$$

2.4 Tire Modeling

Characterizing the force interaction between the vehicle tires and the road surface is the major challenge in modeling and control of vehicle motion. The actual physical interaction of an elastically deformable pneumatic tire sliding over the road surface is very complex and still not fully understood [26]. In this thesis, we focus on simplified models of the interaction for control design and vehicle motion simulation, describing the forces exerted in the longitudinal and lateral directions as a function of the vehicle states. This section provides a brief description of tire properties based on [27, Chapter 13], [30, Chapter 2], and [26, Chapter 1].

The most important tire characteristic for trajectory tracking is the generation of lateral force. The ability to generate lateral force is strongly dependent on tire contact patch, determined not only by tire dimensions but also by the vertical load

and inflation pressure. The force is generated by adhesion of the two surfaces. The adhesion depends mainly on road surface conditions, being reduced on a wet or icy road. Centrifugal force pushes the vehicle outwards of the turn during cornering. The adhesion between the tire and the road surface supplies the centripetal force and prevents the tire contact patch from sliding outwards. As a result, the tire deforms elastically between the contact patch and the wheel rim, causing lateral deflection and twisting of the tire tread at the tire contact patch [31]. According to the elastic foundation model for lateral force generation, first developed in [32] by Fiala, the elasticity of the tire opposes the deflection with a certain stiffness, exerting lateral force. During cornering, an angle exists between the wheel pointing direction and the velocity vector of the tire contact patch. The angle is denoted as the tire slip angle α and is used as the input variable in modeling of tire lateral force. For the front tire of the single-track model, the tire slip angle α_f is defined as

$$\alpha_\mathrm{f} = \delta - \theta_\mathrm{f} = \delta - \arctan\left(\frac{\dot{y} + l_\mathrm{f}\dot{\psi}}{v_x}\right), \tag{2.10}$$

where θ_f is the direction of the velocity vector at the tire contact patch in a body-fixed coordinate frame and δ is the wheel pointing direction, transforming θ_f into the tire-fixed coordinate frame. Rear tire slip angle α_r is calculated similarly, substituting axle arm l_f with $-l_\mathrm{r}$ and with a zero steering angle. Due to the presence of the tire slip angle, a force is generated opposing the non-zero lateral velocity, which contributes to the lateral deflection of the tread.

Three ranges of operation with respect to tire slip angle exist: *elastic* (or linear), *transitional*, and *frictional* [26, 30]. In the elastic range, the lateral force is approximately linear with respect to the tire slip angle. In the transitional range, tire sliding starts to occur and the elastic capability of the tire is becoming saturated. In the frictional range, the adhesion limit is exceeded and the surfaces begin to slide against each other. The source of the lateral force is no longer an elastic, but a frictional force.

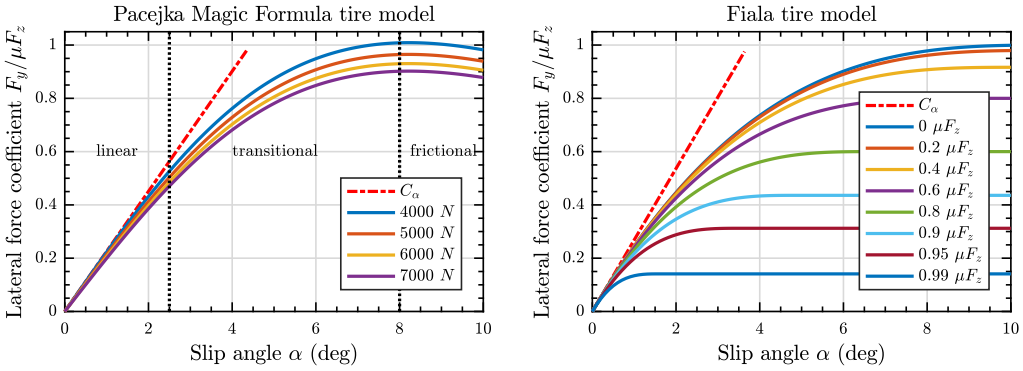
The key behavior the lateral tire model shall cover is the following: the force increment decreases as the tire slip is increased, the maximum available force is limited, and the maximum varies with tire-road friction and the vertical loading force of the tire. The above operational ranges are captured in the most widely used tire models, Pacejka [33, 26] and Fiala [32], both semi-empirical. Each of the models describes an analytic curve which defines the typical shape of the tire cornering force characteristic. The model is usually fitted to data obtained by measuring the force exerted by the tire at different values of tire slip angle and under different values of vertical loading force in a tire testing rig. The models vary in the number of parameters and hence in the flexibility of the fit.

2.4.1 Pacejka Magic Formula

The Magic Formula tire model introduced in [33] and covered in detail in [26] provides an analytic formula for an accurate description of measured steady-state tire force behavior. The formula for the lateral force with zero horizontal and vertical shift is

$$F_y(\alpha) = D\sin\left[C\arctan\{B\alpha - E\left(B\alpha - \arctan B\alpha\right)\}\right], \tag{2.11}$$

with B stiffness factor, C shape factor, D peak value and E curvature factor, determining the curve shape [27]. The slope of the curve at the origin, the tire cornering stiffness C_α, is given as BCD.



(a) Pacejka, for varying F_z (b) Fiala, for varying F_x

Figure 2.3: Normalized lateral force curves of the selected tire models

A typical shape of the lateral force coefficient (lateral force normalized by the vertical load) is shown in Fig. 2.3a, with parameter values selected arbitrarily to show the typical shape as in [30]. The curve is anti-symmetric and was plotted only for positive values of tire slip angle α. The Pacejka Magic formula captures all three ranges of tire operation and with four parameters offers sufficient freedom to fit the curve to measured tire data. Tire load sensitivity modeled using $D \propto F_z^{0.8}$ is shown, creating the typical decrease of lateral force coefficient with increasing tire loading force. While the curve is plotted for the use case of lateral force generation, Eq. (2.11) can be used for a description of longitudinal force or tire self-aligning moment generation. In contrast to Fiala tire model shown later, Pacejka tire model can capture the reduction of lateral force in the frictional range of operation.

2.4.2 Fiala Tire Model

During combined generation of lateral and longitudinal force, the total force exerted in the horizontal plane is limited by the adhesion μF_z, where F_z is the normal load applied to the tire and μ is the friction coefficient. The relation between longitudinal and lateral force limit can be modeled using the *friction* or *Kamm's circle* of forces

$$\mu F_z \geq \sqrt{F_x^2 + F_y^2}, \tag{2.12}$$

neglecting factors such as the tire load sensitivity. Tire models describing the combined force generation in more detail using combined slip are described in [26, 27, 30].

In this thesis, a simpler model of combined force generation will be used, the modified Fiala tire model as described in [34], where the model was used to analyze drift equilibria of the nonlinear single-track vehicle model and to design a controller for sustained drifting. The model with the adapted ISO sign convention is given as

$$F_y = \begin{cases} C_\alpha \tan(\alpha) - \frac{C_\alpha^2}{3F_{y,\max}} |\tan(\alpha)| \tan(\alpha) + \frac{C_\alpha^3}{27F_{y,\max}^2} \tan(\alpha)^3, & |\alpha| < \alpha_{\text{sl}}, \\ \text{sgn}(\alpha) F_{y,\max}, & |\alpha| \geq \alpha_{\text{sl}}, \end{cases} \tag{2.13}$$

where C_α is the tire cornering stiffness, α is the tire slip angle and

$$\alpha_{\text{sl}} = \frac{3F_{y,\max}}{C_\alpha} \tag{2.14}$$

is the tire slip angle at which the exerted force reaches the available maximum and becomes saturated. The maximum lateral tire force is

$$F_{y,\max} = \nu \mu F_z. \tag{2.15}$$

The modified Fiala tire model of Eq. (2.13) incorporates the coupling between the lateral, longitudinal, and normal force as a function $F_y(\alpha, F_x, F_z)$, meaning that the lateral force F_y is dependent on the forces acting in the remaining x, z directions. The total force in the horizontal direction is restricted by the friction circle Eq. (2.12) and so the available lateral force is reduced by coefficient ν

$$\nu(F_x, F_z) = \frac{\sqrt{\mu^2 F_z^2 - F_x^2}}{\mu F_z} \in \langle 0,\, 1 \rangle, \tag{2.16}$$

assuming $|F_x| \leq \mu F_z$.

A shape typical for the Fiala tire model is shown in Fig. 2.3b with the tire cornering stiffness indicated in red. Compared to the Pacejka tire model of Fig. 2.3a, the increment of the lateral force coefficient (slope of the curve) decreases faster and its decrease rate cannot be controlled by a parameter. Behavior in the frictional range cannot be parametrized, the lateral force coefficient becomes saturated. Although Fiala tire model curve is not as flexible, it can be parametrized more easily using a single parameter C_α. While Fiala model does not include longitudinal dynamics of the wheel (wheel rotational inertia and generation of longitudinal slip), it describes the reduction of the available lateral force when a longitudinal force is applied during braking or accelerating.

Since the interface for trajectory tracking controller shall be the longitudinal acceleration command, the modified Fiala tire is a viable validation model without the need of using tire models of higher complexity. Additionally, through F_z the model can take into account longitudinal weight transfer as a result of longitudinal acceleration or increase of dynamic load due to aerodynamic downforce. Therefore, Fiala tire model was selected for use in the nonlinear single-track model of vehicle dynamics.

2.4.3 Linear Tire Model

For classical control design, a linear approximation of the cornering force characteristic is used, resulting in a linear model and allowing controller design using linear systems methods. However, the cornering force characteristic is nonlinear and the approximation holds only for small slip angles, resp. up to medium levels of lateral acceleration, approximately for $|a_y| \leq 4\,\mathrm{m/s^2}$. For the controller design, the nonlinear Eq. (2.5) of vehicle motion is linearized. The linearized model is known in the literature as *Linear bicycle model of lateral vehicle dynamics*.

To obtain linear models, the longitudinal and lateral dynamics shall be decoupled. The lateral dynamics model is usually derived with the assumption of constant longitudinal velocity $\dot{x}$, neglecting the longitudinal dynamics of Eq. (2.5a), and for the range of tire slip angles, where the tire cornering force characteristic is approximately linear. A linear model of longitudinal vehicle dynamics is described separately in Section 2.8.

Decoupling is done by assuming that the longitudinal velocity dynamics are handled by an independent controller. The trajectory is planned below the maximum force limit to avoid the stronger coupling in force saturation. Under such assumptions, the longitudinal velocity $\dot{x}$ is replaced by a time-varying parameter v_x and Eq. (2.5a) is not further used in linearizing the model of lateral dynamics.

Second, the small angle approximation is used for the steering angle δ, approximating $\sin(\delta) \approx \delta$, $\cos(\delta) \approx 1$, and for the slip angle of each tire in Eq. (2.10)

$$\alpha_{\mathrm{f}} = -\arctan\left(\frac{\dot{y} + l_{\mathrm{f}}\dot{\psi}}{v_x}\right) + \delta \approx -\frac{\dot{y} + l_{\mathrm{f}}\dot{\psi}}{v_x} + \delta, \tag{2.17a}$$

$$\alpha_{\mathrm{r}} = -\arctan\left(\frac{\dot{y} - l_{\mathrm{r}}\dot{\psi}}{v_x}\right) \approx -\frac{\dot{y} - l_{\mathrm{r}}\dot{\psi}}{v_x}. \tag{2.17b}$$

The lateral tire force is approximated as proportional to the tire slip angle α for small slip angles [27], resp. in the elastic range of tire operation [26]. A first-order approximation of the tire force characteristic is obtained from Taylor expansion of the tire cornering force characteristic

$$C_\alpha = \frac{\partial F_y}{\partial \alpha}\Big|_{\alpha:=0}. \tag{2.18}$$

Proportionality constant C_α equals the tire cornering stiffness of the Fiala tire model Eq. (2.13), or BCD of the Pacejka tire model Eq. (2.11). Lastly, the effect of the forward wheel tractive force $F_{x,\mathrm{f}}$ to lateral motion dynamics $\ddot{y}$ is neglected by taking $F_{x,\mathrm{f}}\sin(\delta) \approx 0$.

Tire cornering forces are expressed as linear functions of slip angles, respectively as linear functions of the vehicle states, rewriting Eq. (2.5b) and (2.5c) into

$$\ddot{y} = \frac{F_{y,\mathrm{r}} + F_{y,\mathrm{f}}}{m} - \dot{\psi}v_x = \frac{1}{m}\left[-C_{\alpha_{\mathrm{r}}}\left(\frac{\dot{y} - l_{\mathrm{r}}\dot{\psi}}{v_x}\right) - C_{\alpha_{\mathrm{f}}}\left(\frac{\dot{y} + l_{\mathrm{f}}\dot{\psi}}{v_x} - \delta\right)\right] - \dot{\psi}v_x, \tag{2.19a}$$

$$\ddot{\psi} = \frac{l_{\mathrm{f}}F_{y,\mathrm{f}} - l_{\mathrm{r}}F_{y,\mathrm{r}}}{I_z} = \frac{1}{I_z}\left[l_{\mathrm{r}}\,C_{\alpha_{\mathrm{r}}}\left(\frac{\dot{y} - l_{\mathrm{r}}\dot{\psi}}{v_x}\right) - l_{\mathrm{f}}\,C_{\alpha_{\mathrm{f}}}\left(\frac{\dot{y} + l_{\mathrm{f}}\dot{\psi}}{v_x} - \delta\right)\right] \tag{2.19b}$$

and collecting the terms into a state-space form with state vector $\mathbf{x} = [\dot{y},\, \dot{\psi}]^\top$

$$\begin{bmatrix}\ddot{y}\\ \ddot{\psi}\end{bmatrix} = \underbrace{\begin{bmatrix} -\dfrac{C_{\alpha_{\mathrm{r}}} + C_{\alpha_{\mathrm{f}}}}{mv_x} & \dfrac{l_{\mathrm{r}}C_{\alpha_{\mathrm{r}}} - l_{\mathrm{f}}C_{\alpha_{\mathrm{f}}}}{mv_x} - v_x \\ \dfrac{l_{\mathrm{r}}C_{\alpha_{\mathrm{r}}} - l_{\mathrm{f}}C_{\alpha_{\mathrm{f}}}}{I_z v_x} & -\dfrac{l_{\mathrm{r}}^2 C_{\alpha_{\mathrm{r}}} + l_{\mathrm{f}}^2 C_{\alpha_{\mathrm{f}}}}{I_z v_x} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix}\dot{y}\\ \dot{\psi}\end{bmatrix} + \underbrace{\begin{bmatrix}\dfrac{C_{\alpha_{\mathrm{f}}}}{m}\\ \dfrac{l_{\mathrm{f}}C_{\alpha_{\mathrm{f}}}}{I_z}\end{bmatrix}}_{\mathbf{B}}\delta. \tag{2.20}$$

2.5 Validation and Identification

In this section, the nonlinear single-track model of Section 2.3 with Fiala tire model described in Subsection 2.4.2 and with the linear tire model of Subsection 2.4.3 are identified and validated with respect to a high-fidelity model of vehicle dynamics provided in the commercial software *IPG Carmaker*. The focus of identification is steering behavior and lateral dynamics of the vehicle. Since obtaining experimental measurements with a real vehicle is cost prohibitive for an academic thesis, a high-fidelity model is used as a reference instead. The goal is to fit the simplified models to the simulated results of the high-fidelity model. The physical parameters, such as mass, its distribution, and moment of inertia can be obtained based on a multibody model from computer-aided design software available to the vehicle manufacturers. The main parametric uncertainty comes from the nonlinear tire-road interaction, which is to be described using a first-order approximation.

Firstly, the lateral transient response is identified using the ISO 7401 step steer test method [35] at speed of 80 km/h. The simulation results are used to parametrize the simplified models derived in the previous section. Subsequently, the ISO 7401 step steer test at two different speeds, namely at 60 km/h and 100 km/h as suggested in [35], is used to validate the model with identified parameters. Many phenomena are influencing the steering characteristics of a real vehicle which were not included in the simplified models, such as lateral load transfer, tire camber induced by vehicle rolling motion, by tire self-aligning moment, or by suspension kinematics. All these phenomena influence the lateral transient response of the vehicle.

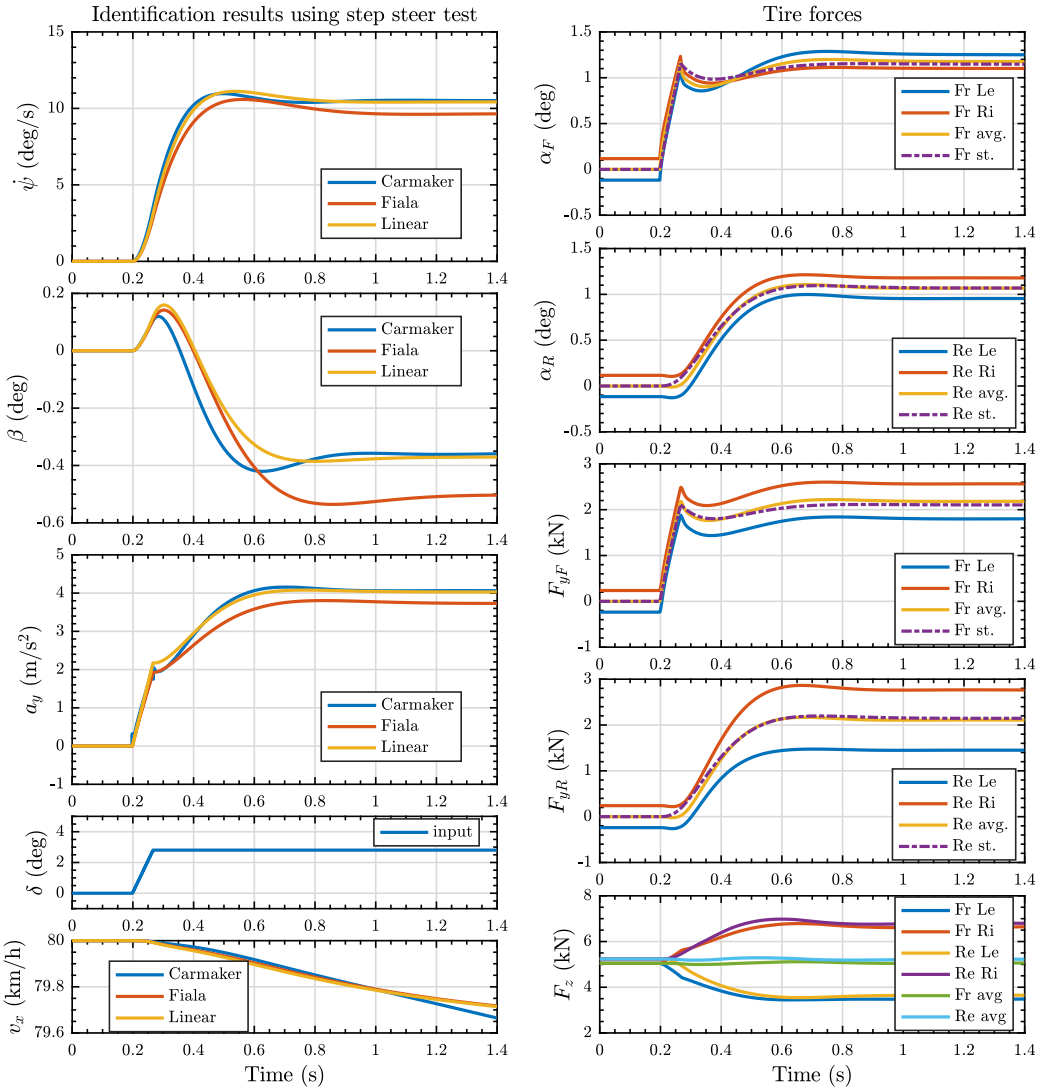
The tire cornering stiffness of the simplified single-track model cannot be determined based on the tire cornering force characteristic and the static load distribution alone. As discussed in [26], front and rear *axle effective side force characteristics* are introduced. The wheels are steered not only by the steering wheel, but a steering angle exists due to body roll through suspension kinematics, steer compliance, initial toe and camber angle of the wheels. Therefore, the values of tire cornering stiffness have to be adjusted to account for the unmodeled phenomena to improve the ability of the simplified model predicting the behavior of a higher-order, high-fidelity vehicle model. The effective cornering stiffness of an axle is defined as the ratio of the axle side force and the *virtual axle slip angle* [26], which is defined with respect to the center of the axle. Results of the ISO 7401 step steer test were used to lower the values of the front and rear cornering stiffness to match the effective values.

The identification test was executed at a constant speed of 80 km/h. The steering wheel was abruptly moved to an angle at which the steady-state lateral acceleration reaches $a_{y,\mathrm{ss}} = 4\,\mathrm{m/s^2}$. The required position of the steering wheel angle is determined beforehand, using a slow ramp steer or steady-state circular motion. Mandatory measured signals of the experiment, as defined by the ISO 7401 standard, are shown in Fig. 2.4a in blue color.

Lateral force is generated only at the front axle during the rising edge of the step. The slip angle at the rear axle occurs with a time delay, and as it increases, lateral force is generated at the rear axle. The vehicle side slip angle is initially positive, but as the rear axle slip angle increases, its sign changes, reaching a negative steady-state value. In the steady-state cornering conditions, the tire cornering forces and the centrifugal force are in balanced. Similarly, the moments of the tire cornering forces are in balance with respect to the center of mass.

The test is used to match the steady state yaw rate gain $\delta \to \dot{\psi}$, lateral acceleration gain $\delta \to a_y$ and time-constants of the steering response. Simulated results were compared against the response of the linear model Eq. (2.20) and $C_{\alpha_\mathrm{r}}, C_{\alpha_\mathrm{f}}$ values were adjusted to minimize the Root Mean Square (RMS) errors in β and $\dot{\psi}$, each normalized by the respective steady-state value. Note that $\dot{y} = v_x\beta$ for the linear model. The response of the identified linear model is shown in Fig. 2.4a in yellow. The response of the nonlinear single-track model with Fiala tire model of the same tire cornering stiffness is shown in orange. Values of the identified cornering stiffness are summarized in Tab. B.1 of Appendix B.

Both simplified models show similar behavior as the high-fidelity model, gains in yaw rate and lateral acceleration channels are well-matched. The high-fidelity model has a slightly different transient response in the side slip angle. Since the increment of tire force decreases with increasing tire slip angle for the Fiala tire model, the simulated response of the nonlinear model has a lower yaw rate and lateral acceleration gain. A larger value of side slip angle β has to be developed to achieve a balance of the front and rear tire forces during the steady-state cornering.

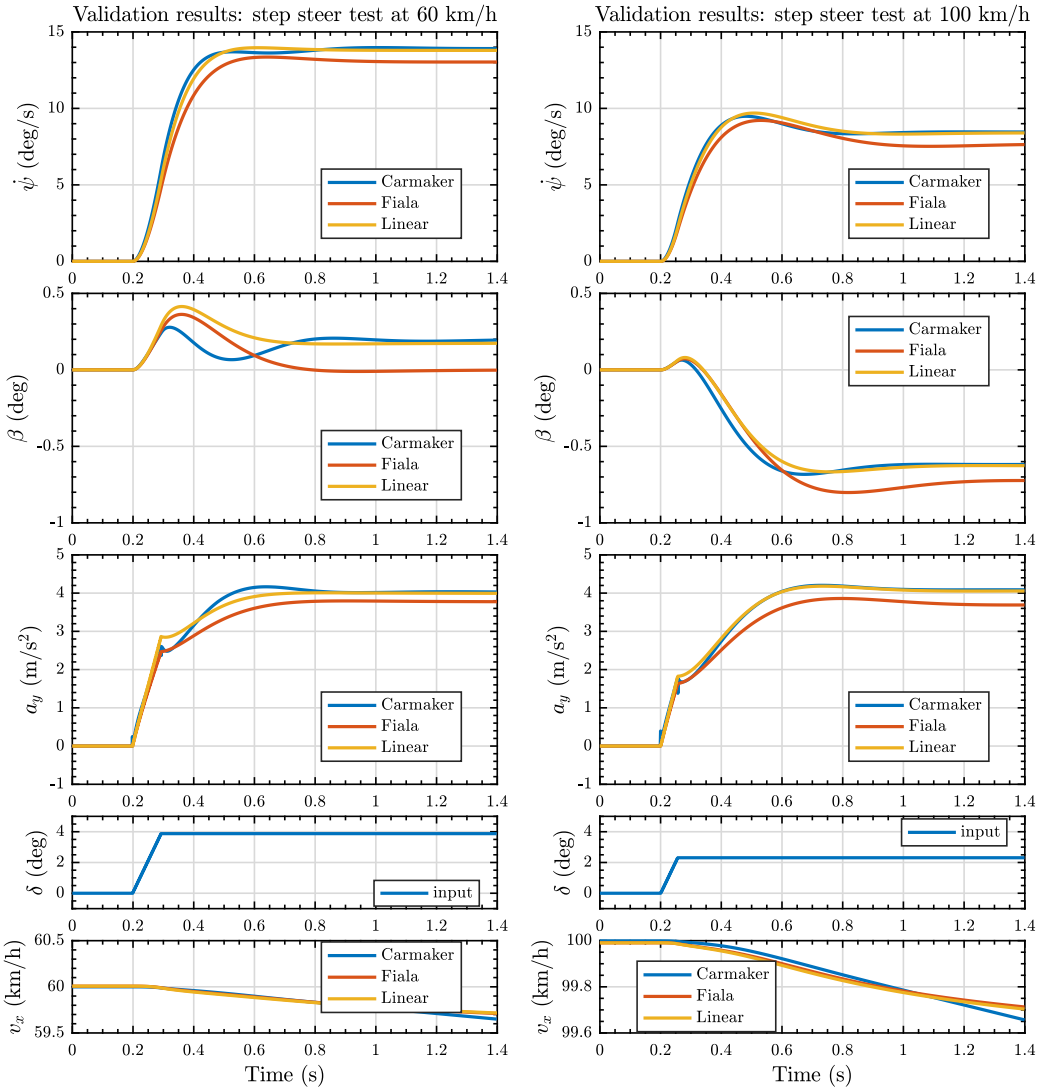


(a) Model identification (b) nonlinear single-track

Figure 2.4: Prediction using simplified dynamic models

However, in Section 2.7, the tire nonlinearity plays an important role when predicting the response during more aggressive maneuvers with higher peak lateral acceleration.

Fig. 2.4b shows the tire slip angles and tire forces, comparing the response of the high-fidelity and the linearized bicycle model. The quantities are plotted per tire, meaning that the lateral force of the whole axle of the single-track model was divided by two. The First two letters in the legends of Fig. 2.4b distinguish front and rear tires, the second two letters indicate the side of the vehicle, the averaged value or the single-track model response (denoted st.). While the rear tire slip angle and rear tire lateral force of the linearized model match well the averaged quantities of the two rear tires, the front axle effective cornering stiffness had to be lowered by a factor of 0.46. In Fig. 2.4b, the front axle slip angle α_{f} was multiplied by the same factor to match the averaged value of tire slip of the high-fidelity model. The front axle effective cornering stiffness appears lower in the overall response as result of the suspension kinematics. Fig. 2.4b also shows the normal forces loading the tires, resp. the lateral weight transfer, which becomes even more significant with increasing lateral acceleration.



(a) at 60 km/h (b) at 100 km/h

Figure 2.5: Validation experiment using ISO 7401 step steer maneuver executed at two different speeds.

Pneumatic tires do not behave according to the laws of dry friction and the increase in the lateral force is below-proportional to the increase in the normal force. Lateral weight transfer decreases the total available lateral force. The nonlinear tire model helps to account for the *tire load sensitivity* as the increment of the tire cornering force decreases with increasing tire slip angle in the transitional range of operation, which is induced by higher levels of lateral acceleration.

Validation experiments in Fig. 2.5 show that the simplified models are valid at different speeds and that the linear parameter-varying model, parametrized by the velocity, can be used to predict the behavior of the vehicle at different speeds, with other parameters fixed. Steady-state gains in the yaw rate and lateral acceleration match. Transient behavior of the side slip angle is slightly different, but the simplified models can approximate the response shape, including the occurrence of a non-minimum phase response at higher speeds.

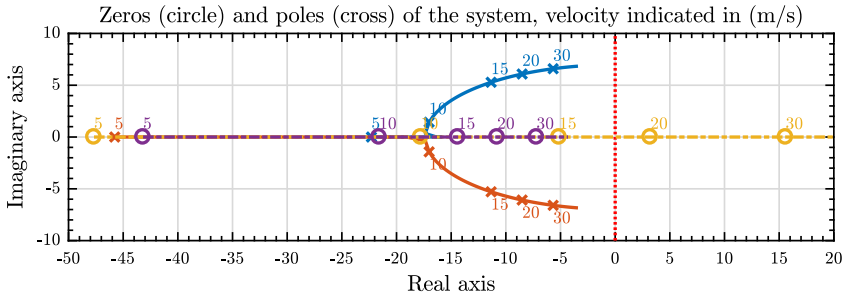


Figure 2.6: Variation of system poles and zeros with longitudinal velocity

2.6 Properties of Linear Model of Lateral Dynamics

The previous section showed that the linear dynamic bicycle model of Eq. (2.20) captures some of the properties important for trajectory tracking, which the kinematic model does not. The required steady-state value of the steering angle to maintain a circular trajectory of curvature κ at longitudinal velocity v_x is described by

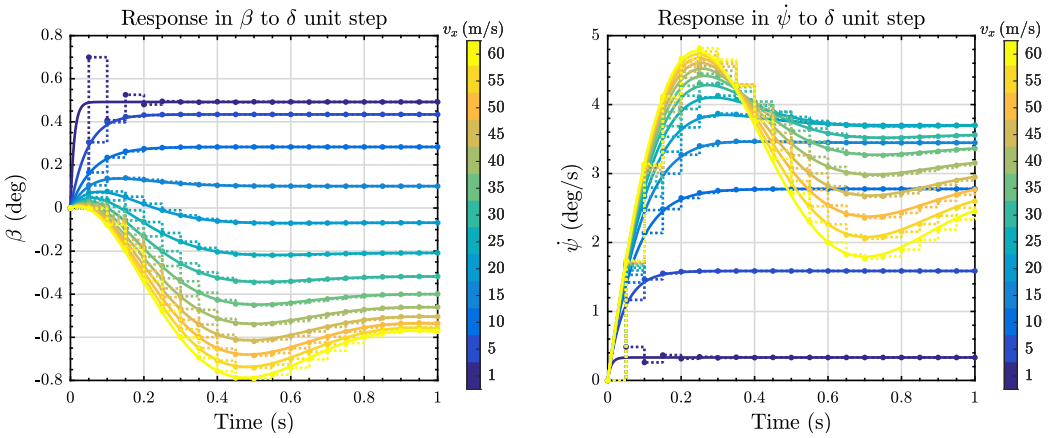
$$\delta = L\kappa + \underbrace{\left(\frac{m_\mathrm{f}}{C_{\alpha_\mathrm{f}}} - \frac{m_\mathrm{r}}{C_{\alpha_\mathrm{r}}}\right)}_{K_V} v_x^2 \kappa, \tag{2.21}$$

where K_V is the *understeer gradient*. Based on the sign of K_V, three types of cars exist: *neutral steer* ($K_V = 0$), *understeer* ($K_V > 0$) and *oversteer* ($K_V < 0$). These terms refer to the typical behavior of the vehicle during steady-state cornering and in the linear region of tire operation. For example, an understeering vehicle executes an arc of larger radius as v_x is increased while keeping δ fixed. Most passenger cars are designed as understeer, since an oversteer vehicle becomes unstable above the *critical velocity* when $L = -K_v v_x^2$. Understeer gradient describes the speed dependency of the steady-state yaw rate gain and is vital for correct prediction of the vehicle turning radius at different speeds and levels of lateral acceleration.

2.6.1 Variation of Transient Response with Longitudinal Velocity

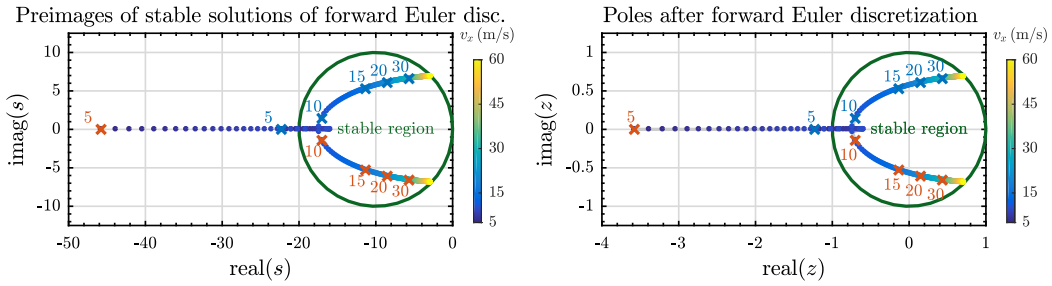
Not only the steady-state properties vary with velocity, but also the transient response. First, let us analyze the map of poles and zeros of the continuous-time linear dynamic bicycle model (with an understeer parametrization) in Fig. 2.6. The system has a pair of poles and transfer function of each output channel has one distinct zero. At low velocities ($v_x \leq 5\,\mathrm{m/s}$), the dynamics are relatively stiff, with both poles real. As the velocity is increased, both poles and zeros shift towards right-hand side, meaning a slower transient response. For $v_x \geq 10\,\mathrm{m/s}$, the poles become complex conjugate, meaning an oscillatory response. The damping is further reduced with increasing velocity. As the velocity is further increased, one of the zeros transitions into the right-hand side of the plane, resulting in the non-minimum phase response.

The response of the system to a unit step of $\delta = 1°$ is shown in Fig. 2.7. Under the small angle assumption, Eq. (2.8) simplifies into $\beta = \dot{y}/v_x$ and so β can be viewed as normalized lateral velocity and used as the output in place of $\dot{y}$. The results match the behavior expected based on the pole-zero map. As the velocity is increased, the response becomes oscillatory with decreasing damping. The non-minimum phase response caused by the positive real part zero is visible in the side



(a) vehicle side slip angle (b) yaw rate

Figure 2.7: Linear model response to a unit step of $\delta = 1°$. Response of the continuous-time model compared with the system discretized using the bilinear transform sampled at $f_s = 20$ Hz.



(a) preimages of stable solutions (b) discretized system poles

Figure 2.8: Discretization using Forward Euler method yields unstable discrete-time system at low speeds, shown for sampling with $f_s = 10$ Hz.

slip response in Fig. 2.7a. Approximately for velocities $v_x \geq 20$ m/s, the response is initially positive and changes the sign during settling. This characteristic non-minimum phase behavior was also observed in the case of the higher-complexity IPG Carmaker vehicle model in Fig. 2.4a and in Fig. 2.5b, while at lower speed in Fig. 2.5a, the β response was minimum phase.

2.6.2 Discretization

Since the controller in this thesis will operate at discrete time steps, the continuous time dynamics of the linear parameter-varying system need to be discretized. As seen in Fig. 2.6, the system response is relatively *stiff* at low velocities, meaning that the associated time constants of the response are fast. Therefore, care must be taken not only because the model becomes singular at low velocities, but also because the sampling rate might be insufficient or the selected discretization method may yield incorrect results. This thesis addresses variation of poles and zeros location with longitudinal velocity, which has the main influence. A root locus analysis with respect to other parameters of the single-track model is described in [36].

Fig. 2.8b shows the poles of the continuous-time linear system parametrized at different speeds and the region for which discretization using forward Euler approximation results in a stable solution of the difference equation. Forward Euler

discretization maps only the poles inside of the circle in Fig. 2.8a to stable discrete-time images. At low speeds, the discretized system would be unstable and thus not usable for predictions in the controller. At high speeds, the boundary of the stable region is approached. A similar analysis was ignored in [28] and the dynamic bicycle model was discretized using forward Euler discretization method at $f_\mathrm{s} = 10\,\mathrm{Hz}$ and even at $f_\mathrm{s} = 5\,\mathrm{Hz}$!

A different discretization method must be used, such as the backward Euler method or the bilinear transform (Tustin's method). The latter ensures a one-to-one mapping between stable regions of the Laplace and $\mathcal{Z}$-transform complex planes, while backward Euler method may stabilize some unstable dynamics. In this thesis, the bilinear transform is used, which requires calculation of a matrix inverse to discretize the plant.

Bilinear transform discretization of a linear state-space model is obtained by approximating the integrals in the solution of the state-transition equation

$$\mathbf{x}_{k+1} - \mathbf{x}_k = \int_{t_k}^{t_{k+1}} \dot{\mathbf{x}}\,\mathrm{d}t = \mathbf{A}\int_{t_k}^{t_{k+1}} \mathbf{x}\,\mathrm{d}t + \mathbf{B}\int_{t_k}^{t_{k+1}} \mathbf{u}\,\mathrm{d}t \tag{2.22}$$

using the trapezoidal rule

$$\begin{aligned}
\mathbf{x}_{k+1} - \mathbf{x}_k =& \mathbf{A}\frac{T_\mathrm{s}}{2}\left[\mathbf{x}_{k+1} + \mathbf{x}_k\right] + \mathbf{B}\frac{T_\mathrm{s}}{2}\left[\mathbf{u}_{k+1} + \mathbf{u}_k\right], \\
\left[\mathbf{I} - \mathbf{A}\frac{T_\mathrm{s}}{2}\right]\mathbf{x}_{k+1} =& \left[\mathbf{I} + \mathbf{A}\frac{T_\mathrm{s}}{2}\right]\mathbf{x}_k + \mathbf{B}\frac{T_\mathrm{s}}{2}\left(\mathbf{u}_{k+1} + \mathbf{u}_k\right), \\
\mathbf{x}_{k+1} =& \underbrace{\left[\mathbf{I} - \mathbf{A}\frac{T_\mathrm{s}}{2}\right]^{-1}\left[\mathbf{I} + \mathbf{A}\frac{T_\mathrm{s}}{2}\right]}_{\mathbf{A}_\mathrm{d}}\mathbf{x}_k + \underbrace{\left[\mathbf{I} - \mathbf{A}\frac{T_\mathrm{s}}{2}\right]^{-1}\mathbf{B}\frac{T_\mathrm{s}}{2}}_{\mathbf{B}_\mathrm{d}}\left[\mathbf{u}_{k+1} + \mathbf{u}_k\right].
\end{aligned} \tag{2.23}$$

To obtain a causal discretization scheme, the first-order hold at the input is reduced to a zero-order hold, assuming $\mathbf{u}_{k+1} \approx \mathbf{u}_k$ and so

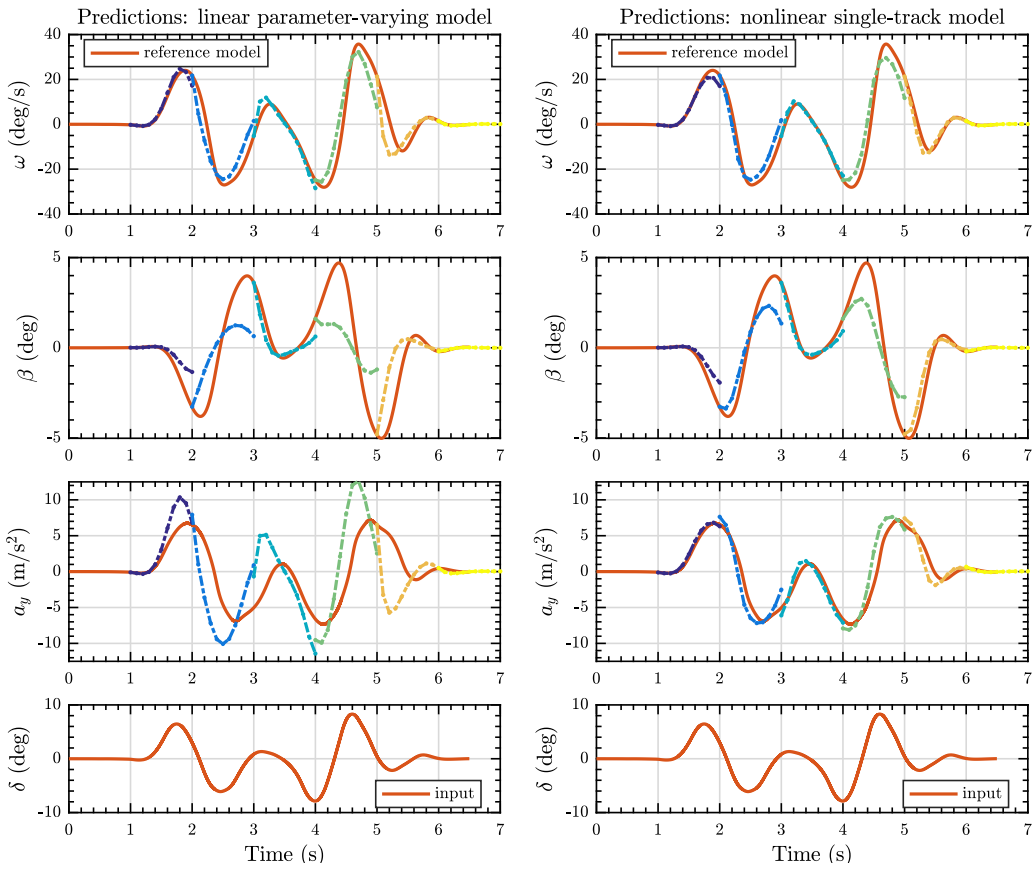
$$\frac{\mathbf{u}_{k+1} + \mathbf{u}_k}{2} \approx \mathbf{u}_k \quad \rightarrow \quad \mathbf{B}_\mathrm{d} = \left[\mathbf{I} - \mathbf{A}\frac{T_\mathrm{s}}{2}\right]^{-1}\mathbf{B}T_\mathrm{s}. \tag{2.24}$$

When approximating the integrals, the time-varying model is assumed as parameter-varying instead, with the values of parameters fixed between the sampling steps. Therefore, system matrices can be factored out of the integrals.

Response of the system discretized at $f_\mathrm{s} = 20\,\mathrm{Hz}$ using Eq. (2.23) and (2.24) is shown in Fig. 2.7 using dot markers. The discretized response is stable, however, some difference exists at low speeds $v_x \leq 5\,\mathrm{m/s}$, an overshoot and oscillations in the discretized response. Therefore, an insufficient controller performance at low speeds may also be caused by a low sampling frequency.

2.7 Model-based Prediction

In this section, the derived models of vehicle dynamics are used to predict the trajectory of the high-fidelity model of IPG Carmeker. Since this thesis deals with model-based control design and in particular with model predictive control, accurate predictions of system state response are necessary. If the prediction of system states is not accurate, the controller cannot account for the effect of its actions in the future. Simulating open-loop predictions helps us to determine a reasonable



(a) lateral, linear parameter-varying (b) nonlinear single-track

Figure 2.9: Prediction using simplified dynamic models

prediction horizon length. Using an excessive length of the prediction horizon increases the computational burden and does not bring any useful information as the prediction no longer correspond to what the actual future behavior of the system will be. Similarly, an accurate model is required to design a state-estimator for calculation of unmeasured observable system states. Model accuracy has a significant impact on the quality of state estimation based on noisy measurements and limits the achievable control performance.

The scope of the application is controlling the vehicle motion in the linear region and possibly in the transitional range of tire operation, up to the limits of adhesion. For verification of the open-loop predictions, ISO 3888-1 double lane change maneuver is used. It is a closed-loop, severe lane-change maneuver used for determining the obstacle avoidance performance of a vehicle and its road-holding ability. For more details on the dimensions of the double lane-change track, refer to [37].

The reference data were obtained by simulating the execution of the maneuver in IPG Carmaker using the IPG Driver. The maneuver was executed at a speed of $v_x = 90$ km/h. Subsequently, an open-loop simulation of the simplified model was executed, initialized from the vehicle true state. If the model prediction is accurate, the simulated response shall correspond with the development of the reference model. Comparison of the responses is shown in Fig. 2.9. The model was simulated for 1 s, applying the steering input δ. Then, the state was re-initialized and simulation of the prediction window was repeated. Each 1 s window is shown in a distinct color.

Fig. 2.9a shows predictions using the LPV model of Eq. (2.20) discretized using

the bilinear approximation at $f_\mathrm{s} = 20\,\mathrm{Hz}$. During the maneuver, the vehicle is operating past its linear range. While the predicted yaw rate $\dot{\psi}$ approximately matches the reference model, predicted lateral acceleration a_y is overestimated. This is due to the unmodeled transitional region of tire operation, tire load sensitivity and lateral load transfer. The capability of the tires to generate lateral force decreases in the transitional region, and the tire is not capable of producing the lateral force required to counteract the centrifugal force. Therefore, steering angle δ and side slip angle β of the reference model are higher to generate the required lateral force. As a result, the linear model predicts a higher lateral acceleration (also with faster responsiveness) and a lower side slip angle for the given input δ.

The prediction is improved in the case of the nonlinear single-track model with Fiala tire model, which covers the transitional range of tire operation. As a result, predictions of side slip angle β and lateral acceleration a_y in Fig. 2.9b match the IPG Carmaker model more closely and approximate the response over the whole prediction window. Inspecting the response, lateral force generation capability of the IPG Carmaker model decreases more during the peaks of lateral acceleration at $t \in \langle 2,\, 3\rangle$ s and for $t \in \langle 4,\, 5\rangle$ s. Predictions are more accurate outside these intervals when the lateral acceleration is lower and the handling limits are not approached.

In conclusion, the prediction provided by the linear model is only approximate in the transitional range of operation. While the model may still be usable for the linear parameter-varying formulation of Model Predictive Control (LPV MPC), longer prediction horizons are unlikely to increase the controller performance. The nonlinear model matches well the response of the reference model even during abrupt maneuvers, high levels of lateral acceleration and in the transitional range of tire operation. The nonlinear model may be used to improve control performance in Nonlinear MPC or for nonlinear state estimation using an Unscented Kalman Filter.

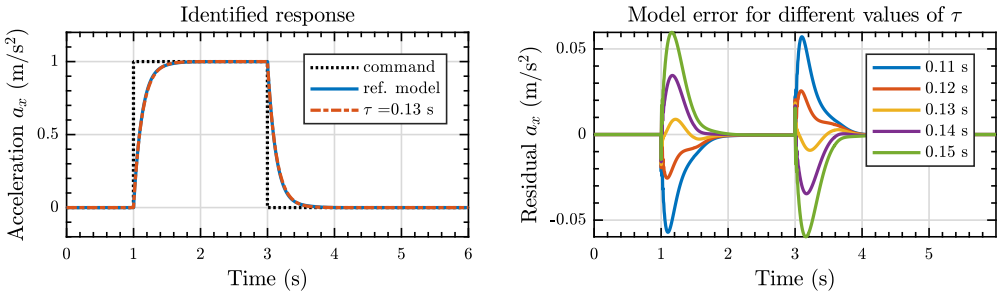
2.8 Linear Model of Longitudinal Dynamics

In this section, longitudinal dynamics of Eq. (2.5a) are simplified into a linear model. For trajectory tracking, we design a higher-level velocity controller to track the desired velocity trajectory while interacting with a lower-level powertrain controller through the interface of desired longitudinal acceleration $a_{x,\mathrm{des}}$. The interface using $a_{x,\mathrm{des}}$ is commonly used in Adaptive Cruise Control (ACC). The underlying powertrain controller is a Proportional Integral (PI) type controller of the desired acceleration $a_{x,\mathrm{des}}$, which controls the brakes and the accelerator pedal, generating longitudinal forces $F_{x,\mathrm{r}}$, $F_{x,\mathrm{f}}$. Thus, Eq. (2.5a) is collected into

$$\ddot{x} = \underbrace{m^{-1}\left(F_{x,\mathrm{r}} + F_{x,\mathrm{f}}\cos(\delta) - F_{y,\mathrm{f}}\sin(\delta)\right)}_{a_x} + \dot{\psi}\dot{y} \approx a_x, \tag{2.25}$$

where $\dot{\psi}\dot{y}$ and $F_{y,\mathrm{f}}$ are neglected for decoupling. The integrative character of the lower-level controller compensates the unmodeled slowly-varying dissipative forces, such as rolling resistance or aerodynamic drag. The higher-level controller must ensure that the constraints of the applicable longitudinal acceleration a_x are respected so that the predictive control is based on a feasible strategy. A traction controller is optionally recommended as a part of the lower-level controller to improve the effectiveness of longitudinal force generation.

Dynamics of the lower-level controller and the powertrain are commonly modeled using a first-order transfer function, called the generalized vehicle longitudinal



(a) identified response

(b) residuals for different τ_L

Figure 2.10: Identified acceleration response

dynamic system in [38]

$$a_x = \frac{K_\mathrm{L}}{\tau_\mathrm{L} s + 1} a_{x,\mathrm{des}}, \tag{2.26}$$

with the system gain usually $K_\mathrm{L} = 1$ and a time constant τ_L. In [38], the model was used for a predictive multi-objective vehicular ACC. A similar modeling approach was applied in [39] for model predictive control of transitional maneuvers for adaptive vehicle cruise control. A first-order approximation of the lower-level controller is also suggested in [27, Chapter 5].

Collecting Eq. (2.25) and Eq. (2.26), the linear state space model of longitudinal dynamics is

$$\begin{bmatrix} \dot{a}_x(t) \\ \dot{v}_x(t) \end{bmatrix} = \underbrace{\begin{bmatrix} -\tau_\mathrm{L}^{-1} & 0 \\ 1 & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} a_x(t) \\ v_x(t) \end{bmatrix} + \underbrace{\begin{bmatrix} \tau_\mathrm{L}^{-1} \\ 0 \end{bmatrix}}_{\mathbf{B}} a_{x,\mathrm{des}}(t), \tag{2.27}$$

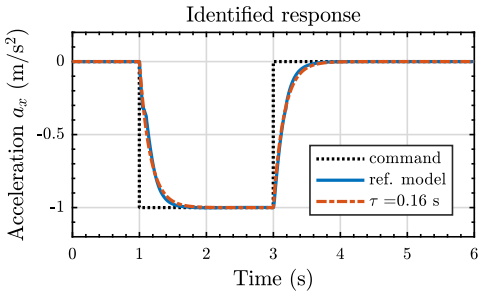
with state vector $\mathbf{x} = [a_x, v_x]^\top$ and input $a_{x,\mathrm{des}}$.

2.8.1 Longitudinal Dynamics Identification

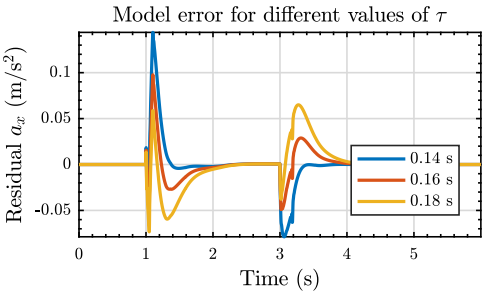
In the Eq. (2.27) model, the time constant τ_L has to be determined. An identification experiment was executed by periodically commanding impulse acceleration up to the maximum speed, followed by periodically commanding impulse deceleration using $a_{x,\mathrm{des}}$ while driving straight. Provided that the desired acceleration of 1 m/s^2 was feasible at given speed and power limit, the observed behavior was approximately linear and not varying with speed.

An acceleration impulse is shown in Fig. 2.10a. The response was fitted with a first-order transfer function, with a time constant of $\tau_\mathrm{L} = 0.13$ s resulting in the lowest RMS error. Error between the reference and the identified model is shown in Fig. 2.10b. A deceleration impulse is shown in Fig. 2.11a. Compared to acceleration, the falling edge during deceleration is not as smooth and contains some nonlinear artifacts, but a first-order transfer function is still a reasonable approximation. The response was fitted with a time constant of $\tau_\mathrm{L} = 0.16$ s, which results in the lowest RMS error but is different from the time constant identified for positive acceleration commands. Error between the reference and identified model is shown in Fig. 2.11b, revealing the artifacts in the edges, compared to Fig. 2.10b.

To obtain a simple model, a single time constant is used for both cases. Best match in terms of RMS error in velocity was achieved for $\tau_\mathrm{L} = 0.14$ s. A validation experiment is shown in Fig. 2.12, combining acceleration followed by deceleration. Modeled response shows that the first-order approximation Eq. (2.26) is sufficient

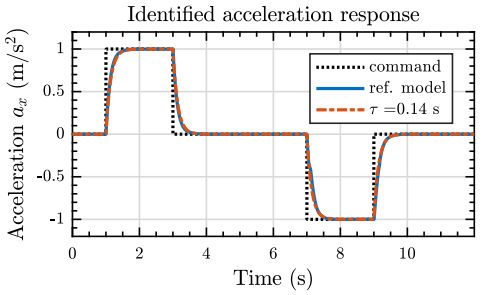


(a) identified response

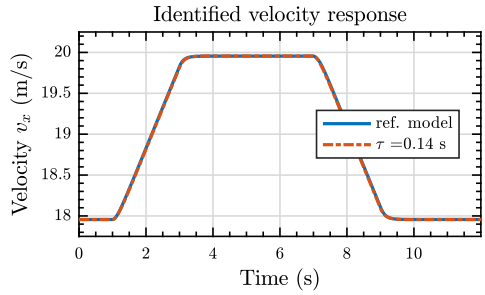


(b) residuals for different τ_{L}

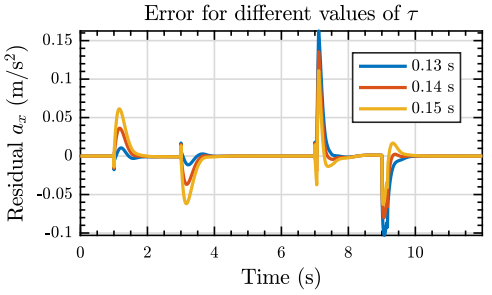
Figure 2.11: Identified deceleration response



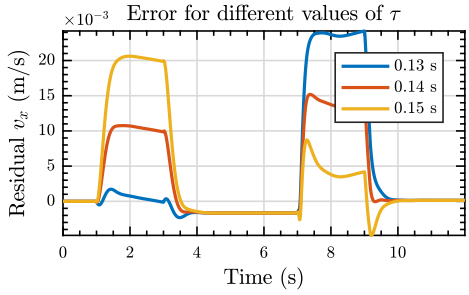
(a) aceleration model



(b) velocity model



(c) residual a_x for different τ_{L}



(d) residual v_x for different τ_{L}

Figure 2.12: Validation of linear model of longitudinal dynamics

to abstract the behavior of the lower-level longitudinal controller and Eq. (2.27) provides an accurate velocity prediction when disturbances (such as road slope) are not present. Modeled longitudinal dynamics of the electric powertrain vehicle are relatively simple, with no dead-time or combustion engine lag.

Lateral motion models are typically formulated with the wheel steering angle as the input, neglecting the dynamics of the steering mechanism. The lower-level is typically vehicle-dependent, and we did not include it explicitly in the lateral motion model. Steering rate limits are assumed during control design to reflect some of the mechanical constraints. Nonetheless, the underlying steering mechanism dynamics can be added to our framework in the same way as the dynamics of the lower-level acceleration controller if needed.

3 | Trajectory Generation

This chapter describes the basic concepts of race car driving and presents an algorithm for generating a minimum-time velocity trajectory based on an idealized vehicle model. A velocity trajectory is necessary for decoupling of the controllers and the possible application in racing motivated using the trajectory that results in the fastest lap time. Typically, the minimum-time trajectory requires operating the vehicle close to its handling limit, testing the performance of the decoupled control architecture under a significant model mismatch. Nonetheless, the controllers are independent of the trajectory generation and can be used in other applications, such as lane keeping and cruise control, tracking trajectories generated with different performance objectives, including ride comfort, safety, and efficiency.

3.1 Race Car Driving

The problem imposed by racing is well-summarized by [30, Chapter 1]. The technical objective in racing is to traverse a given course in minimum time. Moving along the track, the velocity vector of the vehicle is tangent to the path of the vehicle's center of gravity. A race circuit consists of a number of straight and curved segments, and, as the vehicle progresses along the track, the velocity vector is changing both its magnitude and its direction. Driving performance can be quantified in terms of the change of the velocity vector in time, using the total acceleration. Tire adhesion limits the total available force, recalling the friction circle Eq. (2.12). Summing the force contributions of all four tires and normalizing by weight of the vehicle leads to the concept of the vehicle gg diagram. The diagram captures the operational envelope of the vehicle expressed by the combined acceleration in the longitudinal and lateral directions, optionally normalized to multiples of the gravitational acceleration g.

To maximize the average speed, the maximum longitudinal acceleration a_x is applied at the exit of each turn to utilize the straight segment. Similarly, braking with the maximum effort shall be initiated just ahead of a turn. Cornering results in lateral acceleration a_y due to changing the velocity vector direction. Each corner shall be executed at the maximum velocity, limited by the available tire cornering force, which translates to the maximum lateral acceleration. Hence, the goal of a racing driver is to control the car along the gg diagram boundary, maximizing the total acceleration $a_{\text{tot}} = \sqrt{a_x^2 + a_y^2}$, fully utilizing the available tire adhesion.

Exceeding the envelope relates to reaching the peak of the tire force characteristic and the frictional range of tire operation, resulting in a reduction of the available force, loss of adhesion, and possibly sliding. Although a skilled racing driver is capable of controlling the car in a slide, the dynamics become strongly coupled, and the operation of the two controllers is no longer independent. Therefore, the trajectory is designed near the boundary of the gg diagram, but with some safety margin to keep the assumption of decoupled longitudinal and lateral dynamics applicable.

3.2 Trajectory Generation Algorithm

The problem of racing trajectory generation can be formalized as finding control inputs $\mathbf{u}(t)$ that minimize the time t_{finish} when the vehicle crosses the finish line, defined by the distance s_{finish} traveled from the start

$$\begin{aligned} \underset{\mathbf{u}(t)}{\text{minimize}} \quad & t_{\text{finish}} && \text{(3.1a)} \\ \text{subject to} \quad & s(t_{\text{finish}}) = s_{\text{finish}}, && \text{(3.1b)} \\ & \dot{\mathbf{x}}(t) = f\left(\mathbf{x}(t), \mathbf{u}(t), t\right), && \text{(3.1c)} \\ & \mathbf{g}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq \mathbf{0}, && \text{(3.1d)} \end{aligned}$$

where Eq. (3.1c) represents a nonlinear model of the vehicle dynamics and constraints Eq. (3.1d) can represent vehicle handling limits, track borders and other constraints, including the boundary conditions of the problem.

The minimum lap time problem was originally a research interest of professional racing teams. Approaches capable of simultaneously optimizing both the path and the speed profile exist: based on a single-track model with load transfer in [40] or using a twin-track vehicle model in [41, 42], both formulated using nonlinear programming. Local optimization based on a nonlinear MPC maximizing the progress along the track was described in [43].

In this thesis, we consider the path fixed and aim to find a velocity trajectory that results in traversing the path in minimum time. The reference path can be obtained by a kinematic planner as a minimum curvature path, or it can be determined by a human driver and repeated by the autonomous controller as in [11]. Minimum-time speed optimization over a fixed path was formulated as a convex optimization problem in [44]. This thesis utilizes a simpler iterative integration scheme similar to the one introduced in [45] and utilized by [15, 46]. Instead of clothoid curves, we use the algorithm with a uniform discretization of a general reference curve.

3.2.1 Simplified model

For the optimization problem Eq. (3.1), we assume the path to be a curvature-continuous curve, described by curvature κ along the arc length s in the form $\kappa(s)$. The vehicle model is idealized as a point-mass particle moving along the curve with a single degree of freedom

$$f\left(\mathbf{x}(t), \mathbf{u}(t), t\right) = \begin{bmatrix} \dot{s} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ a_x(\kappa(s), v) \end{bmatrix}, \tag{3.2}$$

where the state $\mathbf{x} = [s, v]^\top$ consists of the cumulative distance s travelled along the curve and the tangential velocity v. A single input $a_x\left(\kappa(s), v\right)$ controls the motion, the available longitudinal acceleration, a nonlinear function of the vehicle velocity and track curvature. The road contact model is idealized as a single isotropic tire, with combined force-generating capabilities of the four tires, neglecting lateral and longitudinal load transfer. For simplicity, the effect of the road bank angle and the road grade is neglected. Formulas with the road inclination included can be found in [15]. The maximum transferable force is limited by the tire force using the friction circle model, with the normal force F_z and the friction coefficient μ

$$F_x^2 + F_y^2 \leq (\mu F_z)^2, \tag{3.3}$$

where

$$F_x = ma_x + bv + \frac{1}{2}\rho A c_{\mathrm{D}} v^2 = ma_x + F_{\mathrm{dis}}, \tag{3.4a}$$

$$F_y = mv^2\kappa, \tag{3.4b}$$

$$F_z = mg + \frac{1}{2}\rho A c_{\mathrm{L}} v^2, \tag{3.4c}$$

in which a_x is the input longitudinal acceleration of the vehicle, $b \geq 0$ is the coefficient of linear rolling resistance, ρ is air density, A is the frontal reference aerodynamic area, $c_{\mathrm{D}} \geq 0$ is the drag coefficient, m is the vehicle mass, and $c_{\mathrm{L}} \geq 0$ is the coefficient of the aerodynamic downforce (negative lift).

3.2.2 Solving the Optimization

An approximate solution of the optimization Eq. (3.1) with the idealized motion model of Eq. (3.2) and Eq. (3.3) is obtained using a two-pass iterative forward-backward algorithm. The nonlinear model Eq. (3.2) is discretized by approximating the derivatives using forward differences and by assuming the input piecewise constant $a_k \leftarrow a_x(\kappa(s_k), v_k)$ at time step k with a step length of Δt

$$v_{k+1} = v_k + a_k \Delta t, \tag{3.5}$$

$$s_{k+1} = s_k + v_k \Delta t + \frac{1}{2} a_k (\Delta t)^2 = s_k + \Delta s. \tag{3.6}$$

The optimization is solved by discretizing the curve $\kappa(s)$ at a fixed step Δs in arc length s. Substituting Δt from Eq. (3.5) into Eq. (3.6) yields

$$\Delta s = v_k \left(\frac{v_{k+1} - v_k}{a_k} \right) + \frac{a_k}{2} \left(\frac{v_{k+1} - v_k}{a_k} \right)^2 = \frac{v_{k+1}^2 - v_k^2}{2a_k}, \tag{3.7}$$

from which the update of velocity after step of Δs is

$$v_{k+1} = \sqrt{v_k^2 + 2a_k \Delta s}. \tag{3.8}$$

With the assumption of piecewise-constant acceleration, the distance increment is

$$\Delta s = v_k \Delta t + \frac{1}{2} a_k (\Delta t)^2 = v_k \Delta t + \frac{\Delta v}{2} \Delta t = \frac{v_{k+1} + v_k}{2} \Delta t, \tag{3.9}$$

where $a_k \Delta t = \Delta v = v_{k+1} - v_k$ is the velocity increment and $\Delta t = t_{k+1} - t_k$ is the time increment. The time update is obtained from Eq. (3.9) as

$$t_{k+1} = t_k + \frac{2\Delta s}{v_{k+1} + v_k}. \tag{3.10}$$

The formulas of Eq. (3.8) and Eq. (3.10) are applied to approximate the solution of Eq. (3.2) along a curve. The curve is sampled at N steps using a fixed discretization step of Δs. The arc length samples form an array $s(k), k \in \{1, \ldots, N\}$, and the curvature samples form an array $\kappa(k) \leftarrow \kappa(s(k))$.

The optimization is split into two passes through the arrays. First, the backward algorithm iterates the sampled curve in the reverse order, starting from the finish, determining the maximum velocity in turns and the necessary braking distance before each turn. Second, the forward algorithm iterates in the forward order from the start, maximizing the speed and determining the optimal times of transition between acceleration and braking.

Algorithm 1: Backward pass

Data: $\kappa(k)$, v_{lim}, Δs
Result: $v_{\text{bwd}}(k)$, $v_{\text{max}}(k)$
$k \leftarrow \text{length}(N)$;
$v_{\text{bwd}}(k) \leftarrow v_{\text{lim}}$;
while $k > 1$ **do**
  $v_{\text{max}}(k-1) \leftarrow$ solution of Eq. (3.3) for $\kappa(k-1)$;
  $v_{\text{max}}(k-1) \leftarrow \min(v_{\text{max}}(k-1), v_{\text{lim}})$;
  $a_{\text{lim}} \leftarrow [v^2_{\text{max}}(k-1) - v^2_{\text{bwd}}(k)]/(2\Delta s)$; Eq. (3.7)
  $a(k-1) \leftarrow -h(\kappa(k), v_{\text{bwd}}(k), -1)$; Eq. (3.14)
  $a(k-1) \leftarrow \min(a(k-1), a_{\text{lim}})$;
  $v_{\text{bwd}}(k-1) \leftarrow \sqrt{v_{\text{bwd}}(k)^2 + 2a(k-1)\Delta s}$; Eq. (3.8)
  $k \leftarrow (k-1)$;
end

Backward Pass

The steps of the backward algorithm are summarized in Algorithm 1. At each step, the algorithm calculates the maximum velocity for the previous sample $v_{\text{max}}(k-1)$ by solving Eq. (3.3) for $\kappa(k-1)$ and $a_x = 0$, taking the real positive solution of the quartic equation in velocity v. The result is saturated by a user-defined limit v_{lim}.

Next, the backward pass determines the deceleration required to achieve velocity $v_{\text{max}}(k-1)$ at step $(k-1)$ and the maximum available braking deceleration using function h, which will be described in Subsection 3.2.3. Note that a_k is taken as positive due to integration backward in time. The lower of the two accelerations is applied and next iteration continues by moving backward along the curve.

Forward Pass

The steps of the forward pass are described using Algorithm 2. The forward pass generates the time-optimal velocity $v_{\text{fwd}}(k)$ and acceleration $a(k)$ profiles. When below the velocity profile $v_{\text{bwd}}(k)$ of the backward pass, the maximum available positive acceleration is applied to increase the speed. Once $v_{\text{bwd}}(k)$ is reached, the optimal braking or coasting is applied as in $v_{\text{bwd}}(k)$. During the forward pass, the profile parametrized originally in discrete distance steps is reparametrized using time steps, forming the trajectory parametrized in time.

As a result, the optimal transition times between acceleration and braking are determined to execute the track in the minimum time and under the acceleration constraints. The described iterative scheme results in a *bang-bang* type of trajectory, applying the maximum available input at each step with a single change of input polarity per one turn. The bang-bang control is a typical solution in the minimum-time problems with bounded control inputs.

3.2.3 Powertrain model

Both the backward and the forward pass algorithm utilized a function $h(\kappa, v, d)$ to determine the maximum available longitudinal acceleration in direction $d = \pm 1$, with forward positive. Function h also depends on the vehicle and environmental parameters, such as μ, g, ρ, but these are assumed constant and omitted in the list of the function arguments. Using κ_k, v_k at step k, the available longitudinal

Algorithm 2: Forward pass

Data: $\kappa(k)$, $v_{\text{bwd}}(k)$, Δs, v_0
Result: $v_{\text{fwd}}(k)$, $a(k)$, $t(k)$
$k \leftarrow 1$;
$t(k) \leftarrow 0$;
$v_{\text{fwd}}(k) \leftarrow v_0$;
while $k < N$ **do**
  $a_{\text{lim}} \leftarrow [v_{\text{bwd}}^2(k+1) - v_{\text{fwd}}^2(k)]/(2\Delta s)$; Eq. (3.7)
  $a(k) \leftarrow h(\kappa(k), v_{\text{fwd}}(k), +1)$; Eq. (3.14)
  $a(k) \leftarrow \min(a(k), a_{\text{lim}})$;
  $v_{\text{fwd}}(k+1) \leftarrow \sqrt{v_{\text{fwd}}(k)^2 + 2a(k)\Delta s}$; Eq. (3.8)
  $t(k+1) \leftarrow t(k) + 2\Delta s[v_{\text{fwd}}(k+1) + v_{\text{fwd}}(k)]^{-1}$; Eq. (3.10)
  $k \leftarrow (k+1)$;
end

acceleration is determined from Eq. (3.3) as

$$a_x(\kappa, v, d) = \frac{1}{m}\left(d\,F_x - F_{\text{dis}}\right) = \frac{1}{m}\left(d\sqrt{\mu^2 F_z^2(v) - mF_y(\kappa, v)} - F_{\text{dis}}(v)\right), \quad (3.11)$$

based on the maximum available tire force in the longitudinal direction.

When $d = 1$, the value of a_x is further limited by the vehicle powertrain. In case of an electric-drive vehicle with N_{mtr} motors, the angular speed of each electric motor is calculated based on the vehicle translational speed (neglecting possible wheel slipping), and the available torque is transformed to the wheel tractive force F_{trc}

$$\omega_i(v) = \frac{k_{i,\text{trans}}}{120\pi r}\, v, \quad (3.12)$$

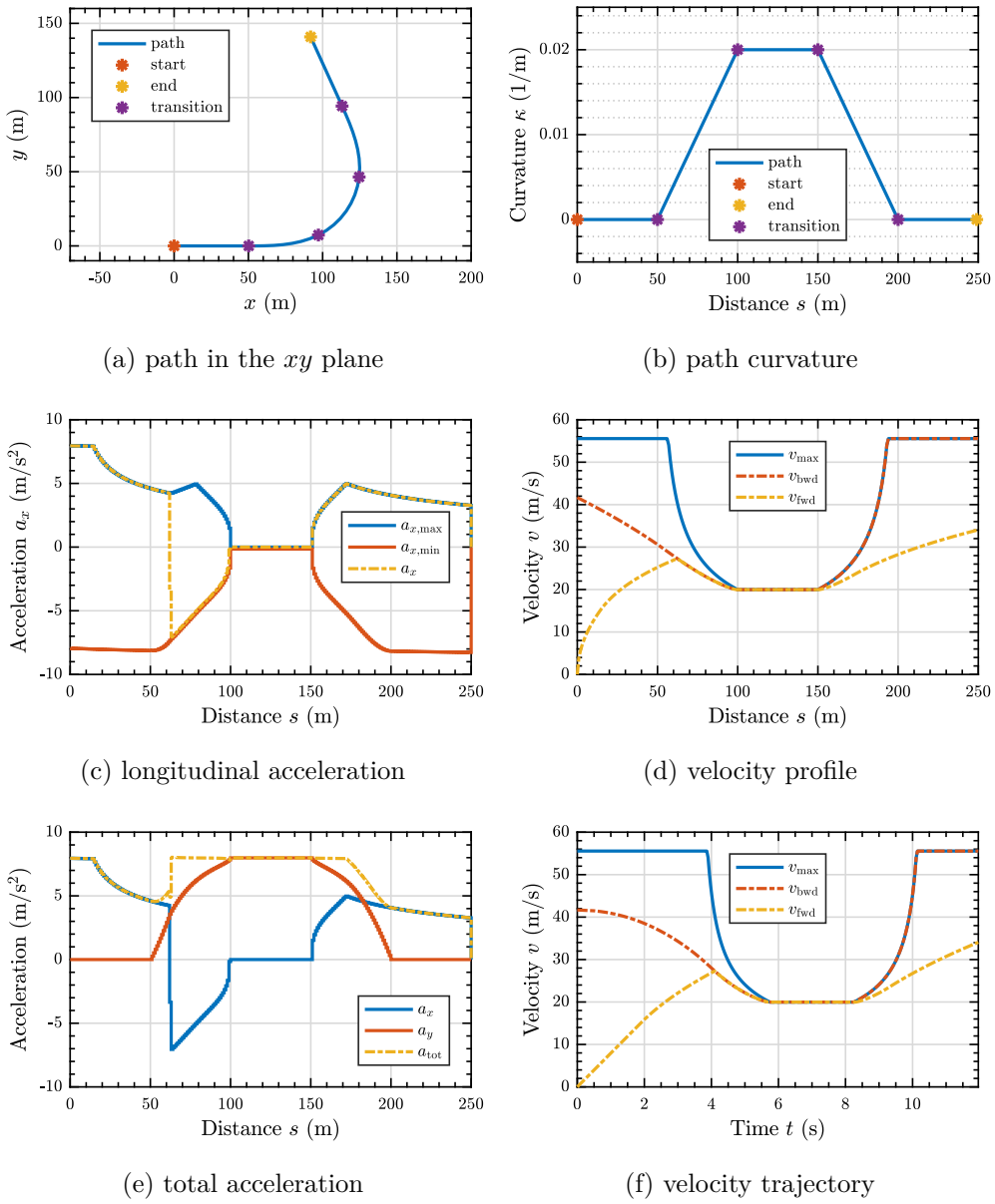
$$F_{\text{trc}}(v) = \frac{1}{r}\sum_{i=1}^{N_{\text{mtr}}} k_{i,\text{eff}}\, k_{i,\text{trans}}\, \tau_i(\omega_i), \quad (3.13)$$

where r is the wheel rolling radius, k_{trans} is the transmission ratio, k_{eff} is the transmission efficiency, and $\tau(\omega)$ represents the torque-rpm curve of the motor. The maximum available longitudinal acceleration is given by

$$h(\kappa, v, d) = \begin{cases} \frac{1}{m}\left[-F_{\text{dis}}(v) + \min\left(\sqrt{\mu^2 F_z^2(v) - F_y^2(\kappa, v)},\, F_{\text{trc}}(v)\right)\right], & \text{if } d = +1, \\ \frac{1}{m}\left[-F_{\text{dis}}(v) - \sqrt{\mu^2 F_z^2(v) - F_y^2(\kappa, v)}\right], & \text{if } d = -1. \end{cases} \quad (3.14)$$

3.2.4 Solution for a Single Turn

An example trajectory generated using the two-pass algorithm is shown in Fig. 3.1. The path is shown in Fig. 3.1a and consists of two straight segments, a circular segment, and two transitional segments of increasing curvature, as indicated in Fig. 3.1b. Calculated longitudinal acceleration a_x is shown in Fig. 3.1c. As vehicle speed increases, the available acceleration decreases due to the maximum power limitation of the motors. As curvature starts to increase at $s = 50\,\text{m}$, the lateral acceleration starts increasing in Fig. 3.1e. When $v_{\text{fwd}} = v_{\text{bwd}}$, braking before the turn starts by following the profile calculated during the backward pass. A constant speed, resulting in the maximum lateral acceleration, is maintained during the



(a) path in the xy plane

(b) path curvature

(c) longitudinal acceleration

(d) velocity profile

(e) total acceleration

(f) velocity trajectory

Figure 3.1: Generated minimum time acceleration and velocity trajectory

turn. As the curvature starts to decrease at $s = 150\,\text{m}$, longitudinal acceleration is gradually applied, keeping the total acceleration at the available limit. Fig. 3.1f shows the resulting velocity trajectory, parametrized in time. Vehicle parameters used for the trajectory generation are summarized in Appendix B and correspond to the parameters of Tesla Model S from IPG Carmaker. For a race car, the effects of aerodynamics would typically be more significant. In Fig. 3.1c the effect is visible as the slight increase of $|a_{x,\min}|$ at higher speeds.

3.3 Path Interpolation

The previous section described the generation of the time-optimal velocity trajectory given the curvature description of the vehicle path. In some scenarios, such as repeating a path driven by a human driver or traversing a path consisting of

user-defined waypoints, curvature description is not directly available. This section presents optional steps of recovering the curve based on the positions sampled from it. The sampling in position is assumed to be accurate, and hence the measured positions are interpolated with a curve. In case of inaccurate position measurement and multiple traversals of the same path, the task can be reformulated using smoothing instead of interpolation.

The resulting curve shall be continuous in curvature. Therefore, the points are interpolated using a cubic spline [47], a piecewise cubic polynomial curve with C^2 continuity (up to the second derivative) at all the internal knots (interpolated points) and with the *not-a-knot* end conditions. The spline is the interpolating solution with the minimum curvature among all interpolating functions having continuous derivatives up to the second one. As the positions are not sampled equidistantly, cumulative *chordal distance* (Euclidean distance of the neighboring knots) is used as the interpolation parameter s, approximating the true arc length.

Curvature of a planar curve in the xy plane is defined as

$$\kappa(s) = \frac{x'(s)y''(s) - y'(s)x''(s)}{\left[(x'(s))^2 + (y'(s))^2\right]^{\frac{3}{2}}}, \tag{3.15}$$

where $x'(s) = \frac{\mathrm{d}}{\mathrm{d}s}x(s)$, $x''(s) = \frac{\mathrm{d}^2}{\mathrm{d}s^2}x(s)$ denote the derivatives with respect to the arc length s. Using the interpolating splines $x(s)$, $y(s)$, the curvature can be obtained analytically by substitution of the derivatives into Eq. (3.15). Since $x(s)$, $y(s)$ are cubic polynomials, the derivatives are also polynomial and continuous. The resulting curvature Eq. (3.15) is a rational function, continuous except at the roots of the denominator, which is always nonzero provided that the interpolated positions do not contain subsequent duplicate values and that the points are spaced sufficiently. Both can be assured by an appropriate preprocessing of the measured positions. Using the spline interpolation, the position and the curvature can be resampled along the curve for an arbitrary value of s, for example at a fixed step of Δs to obtain the discretized curve for the velocity profile generation.

3.4 Trajectory Preview

At each control steps, the position of the vehicle in the world coordinate frame is converted to a path-fixed coordinate frame to evaluate the tracking errors. The origin of the path-fixed coordinate frame is selected as the point on the curve that minimizes the Euclidean distance to the origin of the body-fixed coordinate frame. The solution is approximated by finding the closest point on a piecewise-linear approximation of the spline curve. The accuracy of the approximation can always be improved by resampling the spline $[x(s),\, y(s)]^\top$ with a finer step in s.

Given the vehicle position $\mathbf{v}$ in the the world coordinate frame, and n-th candidate line segment, defined by endpoints $\mathbf{w}_n$ and $\mathbf{w}_{n+1}$ in the the world coordinate frame, with segment length d_n and normalized line vector $\mathbf{s}_n$

$$d_n = \|\mathbf{w}_{n+1} - \mathbf{w}_n\|_2, \qquad \mathbf{s}_n = d_n^{-1}(\mathbf{w}_{n+1} - \mathbf{w}_n), \tag{3.16}$$

the closest point $\mathbf{c}_n$ on the segment n is obtained by evaluating the projection of

vector $\mathbf{p}_n$ onto the segment line vector $\mathbf{s}_n$

$$\mathbf{p}_n = \mathbf{v} - \mathbf{w}_n, \tag{3.17}$$
$$\hat{k}_n = \mathbf{p}_n \cdot \mathbf{s}_n, \tag{3.18}$$
$$k_n = \text{median}([0,\, \hat{k}_n,\, d_n]), \tag{3.19}$$
$$\mathbf{c}_n = \mathbf{w}_n + k_n\, \mathbf{s}_n, \tag{3.20}$$
$$\mathbf{e}_n = \mathbf{v} - \mathbf{c}_n, \tag{3.21}$$

where $\hat{k}_n$ is limited so that $\mathbf{c}_n$ is located between the segment endpoints. The unsigned distance error is $e_n = \|\mathbf{e}_n\|_2$. To speed-up the calculation, the closest points are evaluated only over a local window of candidate line segments which are likely to contain the closest point. Once segment n^* with the lowest e_{n^*} is found, signed distance error e_d is calculated

$$e_d = \text{sign}\big(\det\left([\mathbf{s}_{n^*}, \mathbf{e}_{n^*}]\right)\big)\, e_{n^*}. \tag{3.22}$$

Heading of the curve at a given point can either be obtained by numeric integration of $\kappa(s)$, or, to avoid possible offset due to integration, by approximating the curve heading using the segment directions. The heading at $\mathbf{w}_n$ is

$$\theta_n = \text{atan2}(\mathbf{s}_{n,y},\, \mathbf{s}_{n,x}). \tag{3.23}$$

Reference heading at $\mathbf{c}_{n^*}$ is obtained by linear interpolation

$$\psi_{\text{ref}} = \theta_{n^*} + k_{n^*}(\theta_{n^*+1} - \theta_{n^*}), \tag{3.24}$$

and wrapped

$$\psi_{\text{ref}} \leftarrow \text{mod}(\psi_{\text{ref}} - (\psi - \pi),\, 2\pi) + (\psi - \pi), \tag{3.25}$$

so that the heading error $e_\psi = (\psi_{\text{ref}} - \psi) \in \pm\pi$.

By determining the position in the path coordinate frame, current arc length s is obtained as the cumulative choral length

$$s_{\text{now}} = k_{n^*} + \sum_{n=1}^{n^*-1} d_n. \tag{3.26}$$

To preview the precalculated trajectory, the reference time corresponding to the current arc length s_{now} is obtained using linear interpolation with $s(k)$ and $t(k)$. Next, a time vector sampling the trajectory at uniform time steps T_s is created

$$t_{\text{now}} \leftarrow t(s_{\text{now}}), \qquad \mathbf{t} = t_{\text{now}} + \Big[0,\, T_\text{s},\, \ldots,\, T_\text{s} N_\text{p}\Big]^\top, \tag{3.27}$$

and the reference values are obtained by linear interpolation in the precalculated discrete trajectory. The steps can be implemented using two look-up tables, the first recovering $t(s_{\text{now}})$ and the second sampling the reference trajectory $\mathbf{p}(t)$ and the longitudinal acceleration limits $\mathbf{a}_{\text{lim}}(t)$

$$\mathbf{p}(t) = \Big[a_x(t),\, v_{\text{fwd}}(t),\, \kappa(t)\Big]^\top, \qquad \mathbf{a}_{\text{lim}}(t) = \Big[a_{x,\min}(t),\, a_{x,\max}(t),\Big]^\top, \tag{3.28}$$

at time values of $\mathbf{t}$ to obtain $\mathbf{p}(t_{\text{now}})$ containing the reference generator states a_{ref}, v_{ref}, κ_{ref} and a preview over the prediction horizon.

4 | Controller Structure

This chapter defines the general structure of a tracking controller. In control theory, the goal of tracking is to follow a non-zero reference signal, usually with some guarantees of steady-state tracking errors for different classes of reference signals. The reference signal may be defined either explicitly, as a desired output of the system, or, more compactly, as a response of another dynamic system. The second option is used in this thesis, formulating the tracking task as a *servomechanism problem*.

The servomechanism problem approach allows using the same model structure for abstraction of different tracking controllers and feedback design using the state space methods, such as a Linear Quadratic Regulator (LQR) or a linear Model Predictive Controller (MPC). The structure can abstract the problems of asymptotic tracking, known disturbance rejection, or model matching in a unified approach. Moreover, the identical model structure used for the design of both LQR and MPC helps to compare when the two controllers behave identically and what are the benefits of a predictive control strategy.

4.1 Servomechanism Problem

Linear continuous-time servomechanism problem is defined as controlling error output $\mathbf{e}$ of the dynamic system

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{x}}_1(t) \\ \dot{\mathbf{x}}_2(t) \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{u}(t), \\ \mathbf{e}(t) &= \begin{bmatrix} -\mathbf{C}_1 & \mathbf{C}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix} - \mathbf{D}_1 \mathbf{u}(t), \end{aligned} \tag{4.1}$$

where $\mathbf{x}_1 \in \mathbb{R}^{n_p}$ is the state of the controlled system $(\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1, \mathbf{D}_1)$ with $(\mathbf{A}_1, \mathbf{B}_1)$ stabilizable, $\mathbf{x}_2 \in \mathbb{R}^{n_g}$ is the state of the reference generator, which is a linear autonomous system with $(\mathbf{A}_2, \mathbf{C}_2)$ observable. No direct feedthrough needs to be modeled in the vehicle motion controllers, and so $\mathbf{D}_1 = 0$ is assumed for simplicity.

The output of the controlled system $\mathbf{C_1 x}_1$ shall track the output $\mathbf{C_2 x}_2$ of the reference generator system. The reference output trajectory is generated as an unforced response to an initial condition of $\mathbf{x}_2$. To change the reference, state $\mathbf{x}_2$ is reset, which is possible as the reference generator is a virtual system within the controller. Changing the reference results in a transient response to the new initial condition.

The reference generator can be viewed either as a *command generator* or as a *disturbance model*. The command generator can be a simplified model of the controlled system or a generator of a selected class of signals. For example, the model of a constant output reference is a single integrator. Similarly, a double integrator models ramp reference signals.

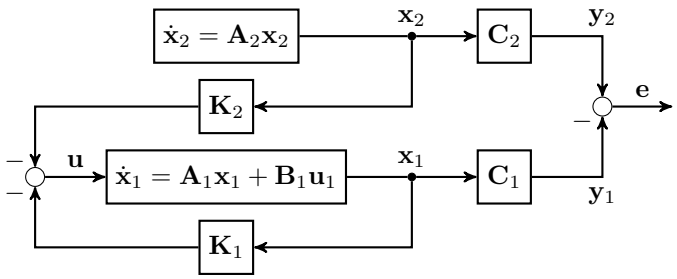


Figure 4.1: State feedback within the servomechanism problem structure

4.2 Structure with State Feedback

For the servomechanism system of Eq. (4.1), a linear state-feedback controller can be designed

$$\mathbf{u} = -\mathbf{K}_1\mathbf{x}_1 - \mathbf{K}_2\mathbf{x}_2, \tag{4.2}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ is the control input and $\mathbf{K}_1 \in \mathbb{R}^{n_u \times n_p}$, $\mathbf{K}_2 \in \mathbb{R}^{n_u \times n_g}$ are the feedback matrices. Structure of the servomechanism problem with state feedback is shown in Fig. 4.1.

Tracking error $\mathbf{e}$ converges to zero if the Internal Model Principle (IMP) is satisfied [48, 49]. In a simplification, the principle requires the presence of all the unstable modes of the reference generator $\mathbf{A}_2$ in the controlled system $\mathbf{A}_1$, and that the unstable modes common to both subsystems are mapped to the error outputs with corresponding degrees. While the internal states are generally non-zero, the tracking error converges to zero if the system satisfies the IMP.

The first step in controller design is to formulate the model according to the servomechanism structure of Eq. (4.1) using continuous-time models. The next step is to discretize the model as a whole, and, finally, the design of the controller is done in discrete time. The block structure of the system remains the same, only the time index is changed to k, and discretized models are assumed instead.

$$\begin{bmatrix} \mathbf{x}_1(k+1) \\ \mathbf{x}_2(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \mathbf{x}_1(k) \\ \mathbf{x}_2(k) \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{B}_1 \\ \mathbf{0} \end{bmatrix}}_{\mathbf{B}} \mathbf{u}(k). \tag{4.3}$$

4.3 Linear Quadratic Servomechanism

The problem of tracking a nonzero reference trajectory is different from the regulation task of controlling $\mathbf{x}(t) \to \mathbf{0}$. To use the formulas developed for LQR, the quadratic performance criterion is formulated to penalize the error $\mathbf{e}$ instead of the state $\mathbf{x}$, so that the criterion value is bounded. The quadratic cost function of the tracking problem over a finite horizon of length N with free final state is

$$J = \frac{1}{2}\mathbf{x}^\top(N)\mathbf{Q}\mathbf{x}(N) + \frac{1}{2}\sum_{k=0}^{N-1} \mathbf{e}^\top(k)\mathbf{Q}_\mathrm{e}\mathbf{e}(k) + \mathbf{u}^\top(k)\mathbf{R}\mathbf{u}(k), \tag{4.4}$$

where $\mathbf{Q}_N \succeq 0$ and $\mathbf{Q}_\mathrm{e} \succeq 0$ are positive semi-definite weighting matrices of the terminal cost, resp. the tracking error cost, and $\mathbf{R} \succ 0$ is a positive definite weighting matrix of the control input cost. In addition to the IMP, the pair $(\mathbf{A}_1, \sqrt{\mathbf{Q}_\mathrm{e}}\mathbf{C}_1)$

must be detectable so that the criterion reflects all states of the controlled system. The input and output variables are selected to allow both $\mathbf{e}(t) \to \mathbf{0}$ and $\mathbf{u}(t) \to \mathbf{0}$ as $t \to \infty$. Therefore, input increments or input rates of change are used as the controlled variable $\mathbf{u}$.

The quadratic cost Eq. (4.4) is expressed using the state $\mathbf{x}$ as

$$J = \frac{1}{2}\mathbf{x}^\top(N)\mathbf{Q}_N\mathbf{x}(N) + \frac{1}{2}\sum_{k=0}^{N-1}\begin{bmatrix}\mathbf{x}^\top(k) & \mathbf{u}^\top(k)\end{bmatrix}\begin{bmatrix}\mathbf{Q} & \mathbf{S}\\ \mathbf{S}^\top & \mathbf{R}\end{bmatrix}\begin{bmatrix}\mathbf{x}(k)\\ \mathbf{u}(k)\end{bmatrix}, \tag{4.5}$$

where $\mathbf{Q} = \mathbf{C}^\top\mathbf{Q}_\mathrm{e}\mathbf{C}$ and $\mathbf{S} = \mathbf{0}$ since $\mathbf{D}_1$ was assumed zero in Eq. (4.1). The general case with $\mathbf{D}_1 \neq \mathbf{0}$ is treated in [49]. The problem of LQ optimal servomechanism is solved by iterating the *Joseph's stabilized Riccati equation*

$$\mathbf{P}(k) = [\mathbf{A} - \mathbf{B}\mathbf{K}(k)]^\top \mathbf{P}(k+1)\,[\mathbf{A} - \mathbf{B}\mathbf{K}(k)] + \mathbf{K}^\top(k)\mathbf{R}\mathbf{K}(k) + \mathbf{Q}, \tag{4.6}$$

where

$$\mathbf{K}(k) = \left(\mathbf{R} + \mathbf{B}^\top\mathbf{P}(k+1)\mathbf{B}\right)^{-1}\left(\mathbf{B}^\top\mathbf{P}(k+1)\mathbf{A}\right), \tag{4.7}$$

is the time-varying optimal feedback gain. The iterations are initialized at time $k = N-1$ with $\mathbf{P}(k+1) = \mathbf{Q}_N$ and evolve backwards in time over the control horizon. Utilizing the structure of the state weighting matrix $\mathbf{Q}$ and of the matrix $\mathbf{P}(k)$

$$\mathbf{Q} = \begin{bmatrix}\mathbf{Q}_{11} & \mathbf{Q}_{12}\\ \mathbf{Q}_{12}^\top & \mathbf{Q}_{22}\end{bmatrix} = \begin{bmatrix}\mathbf{C}_1^\top\mathbf{Q}_\mathrm{e}\mathbf{C}_1 & -\mathbf{C}_1^\top\mathbf{Q}_\mathrm{e}\mathbf{C}_2\\ -\mathbf{C}_2^\top\mathbf{Q}_\mathrm{e}\mathbf{C}_1 & \mathbf{C}_2^\top\mathbf{Q}_\mathrm{e}\mathbf{C}_2\end{bmatrix}, \quad \mathbf{P}(k) = \begin{bmatrix}\mathbf{P}_{11}(k) & \mathbf{P}_{12}(k)\\ \mathbf{P}_{12}^\top(k) & \mathbf{P}_{22}(k)\end{bmatrix}, \tag{4.8}$$

the feedback law can be split in correspondence to the states of the subsystems $\mathbf{x}_1$ and $\mathbf{x}_2$ as in Eq. (4.2). The iterative formulas simplify to

$$\mathbf{K}_1(k) = \left(\mathbf{R} + \mathbf{B}_1^\top\mathbf{P}_{11}(k+1)\mathbf{B}_1\right)^{-1}\left(\mathbf{B}_1^\top\mathbf{P}_{11}(k+1)\mathbf{A}_1\right), \tag{4.9}$$

$$\mathbf{K}_2(k) = \left(\mathbf{R} + \mathbf{B}_1^\top\mathbf{P}_{11}(k+1)\mathbf{B}_1\right)^{-1}\left(\mathbf{B}_1^\top\mathbf{P}_{12}(k+1)\mathbf{A}_2\right), \tag{4.10}$$

which depend only on $\mathbf{P}_{11}$, $\mathbf{P}_{12}$ blocks of $\mathbf{P}$

$$\begin{aligned}\mathbf{P}_{11}(k) &= [\mathbf{A}_1 - \mathbf{B}_1\mathbf{K}_1(k)]^\top \mathbf{P}_{11}(k+1)\,[\mathbf{A}_1 - \mathbf{B}_1\mathbf{K}_1(k)] + \\ &\quad + \mathbf{K}_1^\top(k)\mathbf{R}\mathbf{K}_1(k) + \mathbf{Q}_{11},\\ \mathbf{P}_{12}(k) &= [\mathbf{A}_1 - \mathbf{B}_1\mathbf{K}_1(k)]^\top [\mathbf{P}_{12}(k+1)\mathbf{A}_2 - \mathbf{P}_{11}(k+1)\mathbf{B}_1\mathbf{K}_2] + \\ &\quad + \mathbf{K}_1^\top(k)\mathbf{R}\mathbf{K}_2(k) + \mathbf{Q}_{12},\end{aligned} \tag{4.11}$$

initialized with $\mathbf{P}_{11}(N) = \mathbf{Q}_{11}$, $\mathbf{P}_{12}(N) = \mathbf{Q}_{12}$. The feedback part of $\mathbf{K}_1(k)$, $\mathbf{P}_{11}(k)$ is identical as in the LQR problem, independent of the reference generator.

When the length of optimization horizon N grows, the solution converges to a constant feedback gain $\mathbf{K}_\infty$. Applying the steady state condition $\mathbf{P}(k) = \mathbf{P}(k+1)$ to Eq. (4.6) results in the discrete-time algebraic Riccati equation (DARE). However, it cannot be solved using the conventional solvers (such as `dare` of Matlab Control system toolbox), as the eigenvalues of the reference generator system are uncontrollable. To obtain $\mathbf{K}_\infty$ either iterations of the difference Riccati Eq. (4.6) are used, or more advanced approach of solving the algebraic Riccati equation, based on selecting the maximum stabilizing element from the lattice of all solutions [50], is used. An option for simple systems is to solve for $\mathbf{K}_1$ using `dare` with $(\mathbf{A}_1, \mathbf{B}_1)$ controllable, and then determine $\mathbf{K}_2$ from $\mathbf{K}_1$ such that $\mathbf{u} = \mathbf{0}$ when $\mathbf{e} = \mathbf{0}$ using the system structure.

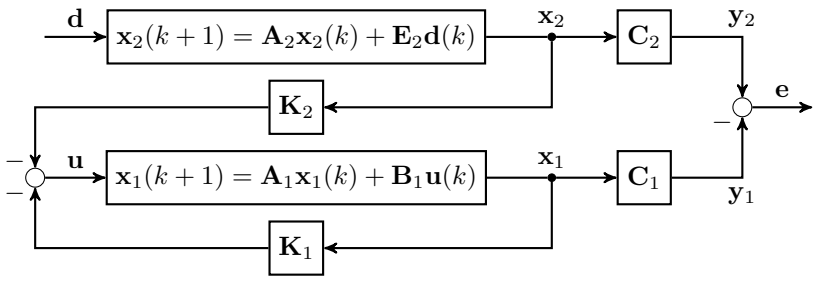


Figure 4.2: Servomechanism structure extended for known disturbance $\mathbf{d}$

In Section 4.4 of [51], the discrete-time LQ tracking problem is derived for explicit output reference signal, resulting in the same feedback gain $\mathbf{K}_1$, but in a different feedforward part, corresponding to *program control* in [49]. This approach offers better tracking of time-varying references but requires storing and prefiltering the whole output reference to obtain the feedforward input. In Section 8.4 of [51], the *command-generator tracker* (CGT) is introduced by incorporating the dynamics of the output reference generation into the control system for the continuous-time case. Tracking for the structure Eq. (4.1) in continuous time is described under the term *regulator with model following*, but with the restrictive assumption of $(\mathbf{A}, \mathbf{B})$ stabilizable, not applicable to the reference generators considered in this thesis.

4.4 Structure with Known Disturbance

Structure of the servomechanism problem can be extended for apriori-known changes in the reference signal generation. The model structure Eq. (4.3) is augmented to

$$\begin{aligned}\begin{bmatrix}\mathbf{x}_1(k+1)\\ \mathbf{x}_2(k+1)\end{bmatrix} &= \underbrace{\begin{bmatrix}\mathbf{A}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_2\end{bmatrix}}_{\mathbf{A}}\begin{bmatrix}\mathbf{x}_1(k)\\ \mathbf{x}_2(k)\end{bmatrix} + \underbrace{\begin{bmatrix}\mathbf{B}_1\\ \mathbf{0}\end{bmatrix}}_{\mathbf{B}}\mathbf{u}(k) + \underbrace{\begin{bmatrix}\mathbf{0}\\ \mathbf{E}_2\end{bmatrix}}_{\mathbf{E}\mathbf{d}(k)=\mathbf{b}(k)}\mathbf{d}(k),\\ \mathbf{e}(k) &= \underbrace{\begin{bmatrix}-\mathbf{C}_1 & \mathbf{C}_2\end{bmatrix}}_{\mathbf{C}}\begin{bmatrix}\mathbf{x}_1(k)\\ \mathbf{x}_2(k)\end{bmatrix},\end{aligned} \tag{4.12}$$

where $\mathbf{d}(k)$ is the known disturbance entering the reference generator system at time step k, altering the generator state and hence the generated reference output to be tracked. Structure of the discrete-time servomechanism problem with the known disturbance is shown in Fig. 4.2.

The knowledge of future disturbance can be utilized in a predictive control strategy. Model structure Eq. (4.12) forms the state-transition equation

$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{b}_k. \tag{4.13}$$

of a single stage of the Linear Time-Varying (LTV) MPC problem described in Chapter 5. In the context of MPC, lower subscripts denotes the discrete time index for notation brevity, while in Chapter 4 the subscripts distinguish the controlled and the reference generator systems.

5 | Model Predictive Control

Model Predictive Controller (MPC) is a control strategy based on solving a finite horizon open-loop optimal control problem. The controller is based on a discrete-time model of the system which is used to predict the system response to the control input along a number of discrete time steps, the prediction horizon. Predicted system response is compared with the reference and both the tracking error and the control input are penalized using a performance cost function. Future behavior of the controlled system is optimized by choosing the best admissible control input, minimizing the performance cost and respecting the input and state constraints.

Feedback control is established by applying only the first input of the calculated optimal input sequence. Applying the rest of the sequence would result in open-loop control, unable to respond to model uncertainties and disturbances. In the next control step, the finite horizon window is shifted and the optimization is repeated, resulting in the control scheme known as the *Receeding horizon control* (RHC). Classical feedback control laws, such as a Proportional, Integral and Derivative (PID) controller, are reactive. Control input is generated in response to the tracking error. On the contrary, MPC computes the predicted response of the system and can utilize a preview of the reference. Control input can be generated even before the tracking error occurs to achieve higher performance during transients.

MPC originates in the 1970s, in control of refining and petrochemical processes, where the system dynamics were relatively slow and allowed to solve the optimization problem within the sampling period. A survey of industrial MPC applications until the year 2000 is given in [52], more recent applications and theoretical results are summarized in [53]. Development of more powerful hardware and specialized solvers enabled the use of MPC in many domains, including fast dynamical systems controlled at sampling rates below the order of milliseconds. In the automotive industry, and the vehicle motion control in particular, MPC was utilized for traction control [9], active steering control of evasive maneuvers [10], yaw and lateral stabilization [11], lane keeping [12], obstacle avoidance and stabilization at the handling limits [54], trajectory tracking for trucks [55] and for passenger vehicles [17], to select several successful applications in the related area.

5.1 MPC Problem Formulation

5.1.1 Model used for Predictions

The system model must be descriptive enough to capture the most significant dynamics of the system, yet simple enough to allow solving the optimization problem in real-time. In this thesis, we focus on MPC with a LPV model for response prediction and a quadratic cost function as the performance criterion. LPV model is a special case of a LTV model with the change of the model parameters know and in-

dependent of the model states. Moreover, we focus on the servomechanism problem structure described in Section 4.4. The state-transition and the output equations used in the MPC formulation are

$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{E}_k\mathbf{d}_k, \tag{5.1}$$
$$\mathbf{e}_k = \mathbf{C}_k\mathbf{x}_k, \tag{5.2}$$

with the lower subscript $k \in \mathbb{N}$ indicating the discrete time step index, $\mathbf{x}_k \in \mathbb{R}^{n_x}$ being the system state, $\mathbf{u}_k \in \mathbb{R}^{n_u}$ the control input, $\mathbf{d}_k \in \mathbb{R}^{n_d}$ the disturbance, $\mathbf{e}_k \in \mathbb{R}^{n_e}$ the tracking error, with appropriately sized system matrices $\mathbf{A}_k \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B}_k \in \mathbb{R}^{n_x \times n_u}$, $\mathbf{E}_k \in \mathbb{R}^{n_x \times n_d}$, $\mathbf{C}_k \in \mathbb{R}^{n_e \times n_x}$ of the discrete-time dynamical system. For the parameter-varying formulation, we assume the system matrices $\mathbf{A}_k, \mathbf{B}_k, \mathbf{E}_k, \mathbf{C}_k$ generally different at each time step k, and no input feedthrough term, $\mathbf{D}_k = 0$.

5.1.2 Quadratic Programming Structure

MPC is most often formulated using a quadratic cost function and solved as an instance of Quadratic Programming (QP), the problem of minimizing a quadratic function under linear equality and inequality constraints. Formulating the MPC problem as a QP is a well-established tractable optimization problem and many specialized solvers exist, utilizing the specific structure of QPs arising from MPC. A positive semidefinite quadratic function is convex, its local optimum is also the global optimum, and so the solver cannot get stuck locally at a sub-optimal solution. The MPC problem is arranged into the QP problem structure

$$\underset{\mathbf{z}}{\text{minimize}} \quad \frac{1}{2}\mathbf{z}^\top\mathbf{H}\mathbf{z} + \mathbf{g}^\top\mathbf{z} \tag{5.3a}$$
$$\text{subject to} \quad \mathbf{A}_{\text{eq}}\mathbf{z} - \mathbf{b}_{\text{eq}} = \mathbf{0}, \tag{5.3b}$$
$$\mathbf{b}_{\text{lb}} \leq \mathbf{A}_{\text{iq}}\mathbf{z} \leq \mathbf{b}_{\text{ub}}, \tag{5.3c}$$
$$\mathbf{z}_{\text{lb}} \leq \mathbf{z} \leq \mathbf{z}_{\text{ub}}, \tag{5.3d}$$

where $\mathbf{z} \in \mathbb{R}^{n_z}$ is the optimized variable, n_z is the problem dimension, Eq. (5.3a) is the objective function quadratic in $\mathbf{z}$, with the Hessian of the problem $\mathbf{H} \in \mathbb{R}^{n_z \times n_z}$ and the vector of the linear part $\mathbf{g} \in \mathbb{R}^{n_z}$. Eq. (5.3b) are the equality constraints with $\mathbf{A}_{\text{eq}} \in \mathbb{R}^{n_{\text{eq}} \times n_z}$, $\mathbf{b}_{\text{eq}} \in \mathbb{R}^{n_{\text{eq}}}$, Eq. (5.3c) are the inequality constraints with matrix $\mathbf{A}_{\text{iq}} \in \mathbb{R}^{n_{\text{iq}} \times n_z}$, and $\mathbf{b}_{\text{lb}} \in \mathbb{R}^{n_{\text{iq}}}$ the lower and $\mathbf{b}_{\text{ub}} \in \mathbb{R}^{n_{\text{iq}}}$ the upper bound vector. Eq. (5.3d) are the box constraints imposed on the optimization variable $\mathbf{z}$, with $\mathbf{z}_{\text{lb}} \in \mathbb{R}^{n_z}$ being the lower and $\mathbf{z}_{\text{ub}} \in \mathbb{R}^{n_z}$ the upper bound vector. Some of the constraints may be omitted in particular formulations.

Usually, solvers take advantage of distinguishing particular types of constraints for computational performance reasons. Formally, the notation can be further compacted by including all the constraints under the inequality type

$$\min_{\mathbf{z}} \quad \frac{1}{2}\mathbf{z}^\top\mathbf{H}\mathbf{z} + \mathbf{g}^\top\mathbf{z} \tag{5.4a}$$
$$\text{s.t.} \quad \mathbf{A}_{\text{c}}\mathbf{z} \leq \mathbf{b}_{\text{c}}, \tag{5.4b}$$

with

$$\mathbf{A}_{\text{c}} = \begin{bmatrix} -\mathbf{A}_{\text{eq}}^\top, & \mathbf{A}_{\text{eq}}^\top, & -\mathbf{A}_{\text{iq}}^\top, & \mathbf{A}_{\text{iq}}^\top, & -\mathbf{I}^\top, & \mathbf{I}^\top \end{bmatrix}^\top, \tag{5.5}$$
$$\mathbf{b}_{\text{c}} = \begin{bmatrix} -\mathbf{b}_{\text{eq}}^\top, & \mathbf{b}_{\text{eq}}^\top, & -\mathbf{b}_{\text{lb}}^\top, & \mathbf{b}_{\text{ub}}^\top, & -\mathbf{z}_{\text{lb}}^\top, & \mathbf{z}_{\text{ub}} \end{bmatrix}^\top. \tag{5.6}$$

5.1.3 Quadratic Cost Function

The goal of MPC at time step n is to find a sequence of control inputs

$$\bar{\mathbf{u}} = [\mathbf{u}_n^\top, \dots, \mathbf{u}_{n+N_\mathrm{p}-1}^\top]^\top \in \mathbb{R}^{n_u N_\mathrm{p}}. \tag{5.7}$$

such that the state response

$$\bar{\mathbf{x}} = [\mathbf{x}_{n+1}^\top, \dots, \mathbf{x}_{n+N_\mathrm{p}}^\top]^\top \in \mathbb{R}^{n_x N_\mathrm{p}}, \tag{5.8}$$

predicted based on $\bar{\mathbf{u}}$ and the initial state $\mathbf{x}_n$, minimizes the weighted quadratic cost over the *prediction horizon*, a window of N_p time steps. The quadratic cost at step k, called the *stage cost* or the *running cost*, is defined as

$$L_k(\mathbf{x}_k, \mathbf{u}_k) = \frac{1}{2}\begin{bmatrix}\mathbf{x}_k \\ \mathbf{u}_k\end{bmatrix}^\top \begin{bmatrix}\mathbf{Q}_k & \mathbf{S}_k^\top \\ \mathbf{S}_k & \mathbf{R}_k\end{bmatrix}\begin{bmatrix}\mathbf{x}_k \\ \mathbf{u}_k\end{bmatrix} = \frac{1}{2}\mathbf{x}_k^\top \mathbf{Q}_k \mathbf{x}_k + \frac{1}{2}\mathbf{u}_k^\top \mathbf{R}_k \mathbf{u}_k, \tag{5.9}$$

where $\mathbf{Q}_k \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{Q}_k \succeq 0$ is a positive semi-definite weighting matrix of the state and $\mathbf{R}_k \in \mathbb{R}^{n_u \times n_u}$, $\mathbf{R}_k \succ 0$ is a positive definite weighting matrix of the control input. The cross weighting matrix $\mathbf{S}_k$ between the states and the inputs is considered zero as in Section 4.3. No linear penalties are considered in L_k. The initial and terminal cost functions can be viewed as specialized cases of the running cost L_k. The initial state is given and penalizing it in the initial cost does not change the optimization result. The last input does not influence any of the states within the prediction horizon, and so the controller would not utilize it. Therefore, the input is not included in the terminal cost.

In MPC, the optimization criterion of Eq. (5.3a) is the running cost Eq. (5.9) summed over the prediction horizon. Using the state and input sequences $\bar{\mathbf{x}}$, $\bar{\mathbf{u}}$, the criterion has a similar structure as the running cost

$$J(\mathbf{z}) = \frac{1}{2}\bar{\mathbf{x}}^\top \bar{\mathbf{Q}}\bar{\mathbf{x}} + \frac{1}{2}\bar{\mathbf{u}}^\top \bar{\mathbf{R}}\bar{\mathbf{u}} = \frac{1}{2}\mathbf{z}^\top \begin{bmatrix}\bar{\mathbf{Q}} & \\ & \bar{\mathbf{R}}\end{bmatrix}\mathbf{z} = \frac{1}{2}\mathbf{z}^\top \bar{\mathbf{H}}\mathbf{z}, \tag{5.10}$$

with the optimization variable vector $\mathbf{z} = [\bar{\mathbf{x}}^\top, \bar{\mathbf{u}}^\top]^\top$ and matrices

$$\bar{\mathbf{Q}} = \begin{bmatrix}\mathbf{Q}_{n+1} & & & \\ & \mathbf{Q}_{n+2} & & \\ & & \ddots & \\ & & & \mathbf{Q}_{n+N_\mathrm{p}}\end{bmatrix}, \quad \bar{\mathbf{R}} = \begin{bmatrix}\mathbf{R}_n & & & \\ & \mathbf{R}_{n+1} & & \\ & & \ddots & \\ & & & \mathbf{R}_{n+N_\mathrm{p}-1}\end{bmatrix}, \tag{5.11}$$

of the optimization criterion.

5.1.4 Incorporating State Predictions

Eq. (5.10) defines the control performance criterion. The remaining task is to embed the system dynamics described by Eq. (5.1) into the quadratic programming problem of Eq. (5.3), for which two main approaches exist.

In the first approach, denoted as *sparse*, or *simultaneous*, both the sequence of states and of control inputs form the optimization variable $\mathbf{z} = [\bar{\mathbf{x}}^\top, \bar{\mathbf{u}}^\top]^\top$. The state transition equation is included for each time step in the form of equality constraints Eq. (5.3b). The second formulation, denoted as *primal dense* or *sequential*, uses recursion of the state transition equation Eq. (5.1) to express the predicted sequence $\bar{\mathbf{x}}$ as a function of the control input sequence $\bar{\mathbf{u}}$ and the initial state $\mathbf{x}_n$.

Formulation names refer to the structure of the Hessian matrix $\mathbf{H}$, which has a sparsity pattern for the simultaneous formulation. When the states are eliminated in the primal dense formulation, Hessian size is reduced and contains no zero entries in general. The sparsity pattern is lost and the Hessian becomes dense. Solvers optimized for each formulation exist, therefore, the MPC problem is derived for both cases.

5.1.5 Sparse Formulation

Let us write several steps of the state response prediction using Eq. (5.1)

$$\begin{aligned}
\mathbf{x}_{n+1} &= \mathbf{A}_n\mathbf{x}_n + \mathbf{B}_n\mathbf{u}_n + \mathbf{E}_n\mathbf{d}_n,\\
\mathbf{x}_{n+2} &= \mathbf{A}_{n+1}\mathbf{x}_{n+1} + \mathbf{B}_{n+1}\mathbf{u}_{n+1} + \mathbf{E}_{n+1}\mathbf{d}_{n+1}\\
&= \mathbf{A}_{n+1}\underbrace{(\mathbf{A}_n\mathbf{x}_n + \mathbf{B}_n\mathbf{u}_n + \mathbf{E}_n\mathbf{d}_n)}_{\mathbf{x}_{n+1}} + \mathbf{B}_{n+1}\mathbf{u}_{n+1} + \mathbf{E}_{n+1}\mathbf{d}_{n+1},\\
&= \mathbf{A}_{n+1}\mathbf{A}_n\mathbf{x}_n + \mathbf{A}_{n+1}\mathbf{B}_n\mathbf{u}_n + \mathbf{B}_{n+1}\mathbf{u}_{n+1} + \mathbf{A}_{n+1}\mathbf{E}_n\mathbf{d}_n + \mathbf{E}_{n+1}\mathbf{d}_{n+1},\\
\mathbf{x}_{n+3} &= \mathbf{A}_{n+2}\mathbf{x}_{n+2} + \mathbf{B}_{n+2}\mathbf{u}_{n+2} + \mathbf{E}_{n+2}\mathbf{d}_{n+2}\\
&= \mathbf{A}_{n+2}\underbrace{(\mathbf{A}_{n+1}\mathbf{x}_{n+1} + \mathbf{B}_{n+1}\mathbf{u}_{n+1} + \mathbf{E}_{n+1}\mathbf{d}_{n+1})}_{\mathbf{x}_{n+2}} + \mathbf{B}_{n+2}\mathbf{u}_{n+2} + \mathbf{E}_{n+2}\mathbf{d}_{n+2},
\end{aligned} \tag{5.12}$$

which can be collected into a matrix form

$$\bar{\mathbf{A}}\bar{\mathbf{x}} + \bar{\mathbf{B}}\bar{\mathbf{u}} - \bar{\mathbf{b}} = \mathbf{0}, \tag{5.13}$$

where

$$\bar{\mathbf{A}} = \begin{bmatrix} -\mathbf{I} & & & \\ \mathbf{A}_{n+1} & -\mathbf{I} & & \\ & \mathbf{A}_{n+2} & -\mathbf{I} & \\ & & \ddots & \ddots \end{bmatrix}, \qquad \bar{\mathbf{B}} = \begin{bmatrix} \mathbf{B}_n & & & \\ & \mathbf{B}_{n+1} & & \\ & & \mathbf{B}_{n+2} & \\ & & & \ddots \end{bmatrix}, \tag{5.14}$$

describe the dynamics of the system and the vector

$$\bar{\mathbf{b}} = \begin{bmatrix} -\mathbf{A}_n\mathbf{x}_n - \mathbf{E}_n\mathbf{d}_n \\ -\mathbf{E}_{n+1}\mathbf{d}_{n+1} \\ -\mathbf{E}_{n+2}\mathbf{d}_{n+2} \\ \vdots \end{bmatrix} \tag{5.15}$$

is the influence of the initial condition $\mathbf{x}_n$ and of the disturbance sequence. Using the optimization variable $\mathbf{z} = [\bar{\mathbf{x}}^\top, \bar{\mathbf{u}}^\top]^\top$, Eq. (5.13) is collected into the form of an equality constraint Eq. (5.3b) with

$$\mathbf{A}_{\text{eq}} \leftarrow \begin{bmatrix} \bar{\mathbf{A}} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{B}} \end{bmatrix}, \qquad \mathbf{b}_{\text{eq}} \leftarrow \bar{\mathbf{b}}. \tag{5.16}$$

Together with the cost Eq. (5.10) in place of Eq. (5.3a), the sparse formulation is complete. Alternatively, in the terms of $\bar{\mathbf{x}}$, $\bar{\mathbf{u}}$, the sparse formulation is

$$\begin{aligned}
\underset{\bar{\mathbf{x}},\,\bar{\mathbf{u}}}{\text{minimize}} \quad & \frac{1}{2}\left(\bar{\mathbf{x}}^\top\bar{\mathbf{Q}}\bar{\mathbf{x}} + \bar{\mathbf{u}}^\top\bar{\mathbf{R}}\bar{\mathbf{u}}\right) && \text{(5.17a)}\\
\text{subject to} \quad & \bar{\mathbf{A}}\bar{\mathbf{x}} + \bar{\mathbf{B}}\bar{\mathbf{u}} - \bar{\mathbf{b}} = \mathbf{0}, && \text{(5.17b)}\\
& \mathbf{b}_{\text{lb}} \le \mathbf{A}_{\text{iq}}[\bar{\mathbf{x}}^\top, \bar{\mathbf{u}}^\top]^\top \le \mathbf{b}_{\text{ub}}, && \text{(5.17c)}\\
& \bar{\mathbf{x}}_{\text{lb}} \le \bar{\mathbf{x}} \le \bar{\mathbf{x}}_{\text{ub}}, && \text{(5.17d)}\\
& \bar{\mathbf{u}}_{\text{lb}} \le \bar{\mathbf{u}} \le \bar{\mathbf{u}}_{\text{ub}}. && \text{(5.17e)}
\end{aligned}$$

5.1.6 Primal Dense Formulation

In the primal dense formulation, state variables are eliminated by expressing the sequence of states $\bar{\mathbf{x}}$ from Eq. (5.12) as a function of the control input sequence, of the known disturbances, and of the initial state

$$\underbrace{\begin{bmatrix}\mathbf{x}_{n+1}\\ \mathbf{x}_{n+2}\\ \mathbf{x}_{n+3}\\ \vdots\end{bmatrix}}_{\bar{\mathbf{x}}} = \underbrace{\begin{bmatrix}\mathbf{B}_n & & & \\ \mathbf{A}_{n+1}\mathbf{B}_n & \mathbf{B}_{n+1} & & \\ \mathbf{A}_{n+2}\mathbf{A}_{n+1}\mathbf{B}_n & \mathbf{A}_{n+2}\mathbf{B}_{n+1} & \mathbf{B}_{n+2} & \\ \vdots & \vdots & \vdots & \ddots\end{bmatrix}}_{\hat{\mathbf{B}}} \underbrace{\begin{bmatrix}\mathbf{u}_n\\ \mathbf{u}_{n+1}\\ \mathbf{u}_{n+2}\\ \vdots\end{bmatrix}}_{\bar{\mathbf{u}}} + \dots$$
$$\underbrace{\begin{bmatrix}\mathbf{E}_n & & & \\ \mathbf{A}_{n+1}\mathbf{E}_n & \mathbf{E}_{n+1} & & \\ \mathbf{A}_{n+2}\mathbf{A}_{n+1}\mathbf{E}_n & \mathbf{A}_{n+2}\mathbf{E}_{n+1} & \mathbf{E}_{n+2} & \\ \vdots & \vdots & \vdots & \ddots\end{bmatrix}}_{\hat{\mathbf{E}}} \underbrace{\begin{bmatrix}\mathbf{d}_n\\ \mathbf{d}_{n+1}\\ \mathbf{d}_{n+2}\\ \vdots\end{bmatrix}}_{\bar{\mathbf{d}}} + \underbrace{\begin{bmatrix}\mathbf{A}_n\\ \mathbf{A}_{n+1}\mathbf{A}_n\\ \mathbf{A}_{n+2}\mathbf{A}_{n+1}\mathbf{A}_n\\ \vdots\end{bmatrix}}_{\hat{\mathbf{A}}} \mathbf{x}_n, \tag{5.18}$$

where $\hat{\mathbf{A}} \in \mathbb{R}^{(n_x N_\mathrm{p}) \times n_x}$, $\hat{\mathbf{B}} \in \mathbb{R}^{(n_x N_\mathrm{p}) \times (n_u N_\mathrm{p})}$, and $\hat{\mathbf{E}} \in \mathbb{R}^{(n_x N_\mathrm{p}) \times (n_d N_\mathrm{p})}$ are the prediction matrices, which can be calculated recursively. The diagonal of $\hat{\mathbf{B}}$ consists of the input matrices $\mathbf{B}_n$ to $\mathbf{B}_{n+N_\mathrm{p}-1}$. Entries below the diagonal are calculated recursively, with $k \in \{1, \dots, N_\mathrm{p}\}$. Starting at row k, the entries left of the diagonal on the row $(k+1)$ are obtained by left-multiplying the entries of the k-th row by $\mathbf{A}_{n+k}$. Structure of $\hat{\mathbf{E}}$ is formed identically using $\mathbf{E}_n$ to $\mathbf{E}_{n+N_\mathrm{p}-1}$. Propagation of the initial condition in $\hat{\mathbf{A}}$ is obtained by recursive left-multiplication, starting with $\mathbf{A}_n$ on the first row, and left-multiplying the row above by $\mathbf{A}_{n+k-1}$ at row k.

Substituting Eq. (5.18) for the states into the objective function Eq. (5.10) gives

$$J(\bar{\mathbf{u}}) = \frac{1}{2}\left(\hat{\mathbf{B}}\bar{\mathbf{u}} + \hat{\mathbf{A}}\mathbf{x}_n + \hat{\mathbf{E}}\bar{\mathbf{d}}\right)^\top \bar{\mathbf{Q}}\left(\hat{\mathbf{B}}\bar{\mathbf{u}} + \hat{\mathbf{A}}\mathbf{x}_n + \hat{\mathbf{E}}\bar{\mathbf{d}}\right) + \frac{1}{2}\bar{\mathbf{u}}^\top \bar{\mathbf{R}}\bar{\mathbf{u}} \tag{5.19}$$

Expanding the expression and collecting the terms with respect to the reduced optimization variable $\bar{\mathbf{u}}$, the quadratic cost becomes

$$J(\bar{\mathbf{u}}) = \frac{1}{2}\bar{\mathbf{u}}^\top \underbrace{\left(\hat{\mathbf{B}}^\top \bar{\mathbf{Q}}\hat{\mathbf{B}} + \bar{\mathbf{R}}\right)}_{\hat{\mathbf{H}}} \bar{\mathbf{u}} + \underbrace{\left(\mathbf{x}_n^\top \hat{\mathbf{A}}^\top + \bar{\mathbf{d}}^\top \hat{\mathbf{E}}^\top\right)\bar{\mathbf{Q}}\hat{\mathbf{B}}}_{\hat{\mathbf{g}}^\top} \bar{\mathbf{u}} + \text{const.} \tag{5.20}$$

Constant terms not dependent on $\bar{\mathbf{u}}$ do not influence the optimizer $\bar{\mathbf{u}}^*$, but only shift the function value at the minimum. As we are interested only in the optimizer, constant terms in Eq. (5.20) are omitted. Further, the vector $\hat{\mathbf{g}}$ can be factored to a part of the initial state $\mathbf{x}_n$ and the disturbance preview $\bar{\mathbf{d}}$, and a part of the linear parameter-varying system dynamics, denoted $\mathbf{G}$,

$$\hat{\mathbf{g}} = \underbrace{\hat{\mathbf{B}}^\top \bar{\mathbf{Q}}\begin{bmatrix}\hat{\mathbf{A}} & \hat{\mathbf{E}}\end{bmatrix}}_{\mathbf{G}} \begin{bmatrix}\mathbf{x}_n\\ \bar{\mathbf{d}}\end{bmatrix}, \tag{5.21}$$

which becomes constant in the case of a Linear Time-Invariant (LTI) system.

Primal dense formulation incorporates the state transitions by injecting the prediction matrices from Eq. (5.18) into the cost function. Compared to the sparse formulation, the problem size is reduced from $N_\mathrm{p}(n_x + n_u)$ to $N_\mathrm{p} n_u$ and the problem

is not equality-constrained. However, the originally sparse structure is lost and even simple box constraints of the state variables

$$\bar{\mathbf{x}}_{\text{lb}} \leq \bar{\mathbf{x}} \leq \bar{\mathbf{x}}_{\text{ub}} \tag{5.22}$$

have to be expressed using prediction matrices of Eq. (5.18) as more complicated polytopic constraints of the control input sequence $\bar{\mathbf{u}}$

$$\underbrace{\bar{\mathbf{x}}_{\text{lb}} - \hat{\mathbf{A}}\mathbf{x}_n - \hat{\mathbf{E}}\bar{\mathbf{d}}}_{\mathbf{b}_{\text{lb}}} \leq \hat{\mathbf{B}}\bar{\mathbf{u}} \leq \underbrace{\bar{\mathbf{x}}_{\text{ub}} - \hat{\mathbf{A}}\mathbf{x}_n - \hat{\mathbf{E}}\bar{\mathbf{d}}}_{\mathbf{b}_{\text{ub}}}. \tag{5.23}$$

A polytope is an intersection of a finite number of half-spaces. To reduce the number of inequality constraints, Eq. (5.18) can be left-multiplied by an output matrix before the insertion into Eq. (5.23), selecting only the states to be constrained. Similarly, Eq. (5.18) can be left-multiplied by an output matrix to formulate any polytopic state constraints of the original sparse problem. The resulting quadratic programming problem in the *primal dense* formulation is

$$\min_{\bar{\mathbf{u}}} \quad \frac{1}{2}\bar{\mathbf{u}}^\top \hat{\mathbf{H}}\bar{\mathbf{u}} + \hat{\mathbf{g}}^\top \bar{\mathbf{u}} \tag{5.24a}$$

$$\text{s.t.} \quad \mathbf{b}_{\text{lb}} \leq \hat{\mathbf{B}}\bar{\mathbf{u}} \leq \mathbf{b}_{\text{ub}}, \tag{5.24b}$$

$$\bar{\mathbf{u}}_{\text{lb}} \leq \bar{\mathbf{u}} \leq \bar{\mathbf{u}}_{\text{ub}}, \tag{5.24c}$$

with the cost criterion of Eq. (5.20), optional box constraints on the states Eq. (5.23) expressed as Eq. (5.24b), and control input box constraints Eq. (5.24c).

5.2 Solution of Unconstrained MPC

In this section, an analytic solution of the unconstrained version of the MPC problem Eq. (5.24a) is described, comparing it with the state feedback structure obtained using LQR in Section 4.3. Unconstrained control performance is studied as a necessary prerequisite to achieve good performance once constraints are applied.

Assuming minimization of the quadratic cost function Eq. (5.20), the analytic solution is obtained using the first-order necessary condition of optimality, by setting the first derivative of the cost function equal to the zero vector: $\hat{\mathbf{H}}\bar{\mathbf{u}} + \hat{\mathbf{g}} = \mathbf{0}$, from which the stationary point is obtained as

$$\bar{\mathbf{u}}^* = -\hat{\mathbf{H}}^{-1}\hat{\mathbf{g}} = -\hat{\mathbf{H}}^{-1}\mathbf{G}\begin{bmatrix}\mathbf{x}_n \\ \bar{\mathbf{d}}\end{bmatrix}. \tag{5.25}$$

To guarantee that $\bar{\mathbf{u}}^*$ is the minimum, the Hessian $\hat{\mathbf{H}}$ of the cost must be positive definite. Since $\bar{\mathbf{R}} \succ 0$ was chosen as positive definite and $\bar{\mathbf{Q}}$ as positive semi-definite, $\hat{\mathbf{H}}$ is positive definite, which also guarantees that its inverse exists.

However, Eq. (5.25) serves for illustration and the minimizer shall not be obtained numerically by calculating the inverse. Especially in the case of unstable systems, the predicted response is growing in time. Therefore, $\hat{\mathbf{B}}$ and $\hat{\mathbf{H}}$ are ill-conditioned. The solution is more appropriately obtained using Cholesky factorization of Eq. (5.25), or even better as a *least-squares* problem solution, as shown in [56, Chapter 3] and repeated here.

First introduce "square-roots" $\mathbf{S}_{\bar{\mathbf{Q}}}$, $\mathbf{S}_{\bar{\mathbf{R}}}$ of the weighting matrices $\bar{\mathbf{Q}}$, $\bar{\mathbf{R}}$ as

$$\mathbf{S}_{\bar{\mathbf{Q}}}^\top \mathbf{S}_{\bar{\mathbf{Q}}} = \bar{\mathbf{Q}}, \qquad \mathbf{S}_{\bar{\mathbf{R}}}^\top \mathbf{S}_{\bar{\mathbf{R}}} = \bar{\mathbf{R}}, \tag{5.26}$$

which can be obtained by applying square root to the individual elements for diagonal matrices, or using Singular Value Decomposition (SVD) for non-diagonal symmetric positive semidefinite matrices. For example $\bar{\mathbf{Q}}$ is diagonalized as

$$\bar{\mathbf{Q}} = \mathbf{U}\mathbf{S}\mathbf{U}^\top, \quad \rightarrow \quad \mathbf{S}_{\bar{\mathbf{Q}}} = \mathbf{U}\mathbf{S}^{\frac{1}{2}}\,\mathbf{U}^\top, \tag{5.27}$$

where $\mathbf{U}$ is an orthogonal diagonalizing matrix and $\mathbf{S}$ is diagonal. Next, assume a vector

$$\begin{bmatrix} \mathbf{S}_{\bar{\mathbf{Q}}}\left(\hat{\mathbf{B}}\bar{\mathbf{u}} + \hat{\mathbf{A}}\mathbf{x}_n + \hat{\mathbf{E}}\bar{\mathbf{d}}\right) \\ \mathbf{S}_{\bar{\mathbf{R}}}\bar{\mathbf{u}}, \end{bmatrix} \tag{5.28}$$

of the weighted tracking error and control effort. Squared l^2 norm of Eq. (5.28), is the quadratic cost Eq. (5.19), up to the scalar multiplication by $\frac{1}{2}$. The optimal control sequence is obtained by setting Eq. (5.28) equal to the zero vector and solving the (almost always in MPC) overdetermined system

$$\underbrace{\begin{bmatrix} \mathbf{S}_{\bar{\mathbf{Q}}}\hat{\mathbf{B}} \\ \mathbf{S}_{\bar{\mathbf{R}}} \end{bmatrix}}_{\mathbf{A}_{\text{lsq}}} \bar{\mathbf{u}} = -\underbrace{\begin{bmatrix} \mathbf{S}_{\bar{\mathbf{Q}}}\hat{\mathbf{A}} & \mathbf{S}_{\bar{\mathbf{Q}}}\hat{\mathbf{E}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{B}_{\text{lsq}}} \begin{bmatrix} \mathbf{x}_n \\ \bar{\mathbf{d}} \end{bmatrix}, \tag{5.29}$$

for $\bar{\mathbf{u}}$ in the least-squares sense, minimizing l^2 norm of Eq. (5.28). The dimension of the zero block in $\mathbf{B}_{\text{lsq}}$ is $(n_u N_\text{p}) \times (n_x + n_d N_\text{p})$.

As discussed in [56], the analytic solution Eq. (5.29) has a better conditioning over Eq. (5.25), the prediction matrices are not squared as in the cost Eq. (5.20). Solution of Eq. (5.29) is obtained numerically using the QR algorithm, in the Matlab notation $\mathbf{M} = \mathbf{A}_{\text{lsq}} \setminus \mathbf{B}_{\text{lsq}}$. Only the first part of the solution, corresponding to n_u inputs, is used as the input in the current control step

$$\mathbf{u} = -\underbrace{\mathbf{N}\mathbf{M}}_{\mathbf{K}} \begin{bmatrix} \mathbf{x}_n \\ \bar{\mathbf{d}} \end{bmatrix} = -\mathbf{K}_1\mathbf{x}_1 - \mathbf{K}_2\mathbf{x}_2 + \mathbf{F}\bar{\mathbf{d}}, \tag{5.30}$$

where $\mathbf{N} = [\mathbf{I}, \mathbf{0}]$, with identity matrix $\mathbf{I} \in \mathbb{R}^{n_u \times n_u}$, zero matrix $\mathbf{0} \in \mathbb{R}^{n_u \times n_u(N_\text{p}-1)}$, selects the first n_u rows of $\mathbf{M}$. In the LTI system case, the matrix $\mathbf{K}$ is constant and can be split into three parts, the state feedback gain $\mathbf{K}_1$ from the controlled system, the state feedback gain $\mathbf{K}_2$ from the reference generator, and a feedforward gain $\mathbf{F}$ from the disturbance preview $\bar{\mathbf{d}}$.

When the MPC cost function Eq. (5.10) is formulated with constant weighting matrices $\mathbf{Q}$, $\mathbf{R}$ over the whole prediction horizon and the horizon length is increased, gains $\mathbf{K}_1$ and $\mathbf{K}_2$ converge to the solution of the infinite horizon LQR problem, as in Section 4.3. While LQR problem optimizes the state feedback gain, MPC optimizes directly the control sequence. However, in the unconstrained case and without the preview ($\bar{\mathbf{d}} = \mathbf{0}$), the resulting structure is identical: a state feedback law. With the disturbance preview, the state feedback structure becomes augmented. The gain $\mathbf{F}$ represents coefficients of a non-causal Finite Impulse Response (FIR) filter acting upon the preview of the known disturbance signal $\mathbf{d}$.

5.3 Solution of Constrained MPC

Solving the constrained version of MPC results in an instance of Quadratic Programming Eq. (5.3) Only a brief description of methods for solving QPs falls within the scope of this thesis. Three main methods are described from the application perspective, as some of the properties are important for the solver selection in Section 5.4.

An introductory overview of the basic methods is summarized in a bachelor's [57], a master's [58], and a doctoral [59] thesis, all available online. An introduction is given in books dedicated to MPC, in [60, Chapter 12] and in [56, Chapter 3.3]. A further level of detail is found in the books [61, 62, 63] dedicated to numerical optimization.

5.3.1 Active Set Method

The most common [60] method for solving a QP is the Active Set Method (ASM). The method is based on the observation that possibly only some of the inequality constraints are *active* at the constrained optimum. The solution lies at the boundary of the active constraints which constitute the *active set*. An equality constraint is imposed by each of the active inequality constraints. Inactive inequality constraints do not restrict the solution.

The active set method iteratively adds and removes potentially active constraints into the working set and solves a QP subject to the equality constraints at each iteration. The iterations are repeated until a feasible solution of the original problem is found. A potential shortcoming of the method is that the number of iterations may be large when many constraints are active since a single constraint is added or removed per iteration. A computational speed-up is achieved by reusing the active set from the previous control step as an initial guess for warm-starting.

5.3.2 Barrier Interior Point Method

In the barrier interior point method, inequality constraints are removed, and the optimization criterion is modified by an additive barrier function to prevent constraint violation. The barrier function shall be differentiable to use Newton's method and shall approach infinity as the optimization variable approaches the constraint boundary. Often a logarithmic barrier function is used.

The constrained quadratic problem is replaced by an unconstrained nonlinear problem at each iteration. Newton's method is applied and the unconstrained nonlinear problem is approximated by a quadratic function. As the solution is approached, the barrier function is incrementally lowered so that the optimizer is unbiased by the barrier function. The method requires starting in the interior of the feasible region. Interior methods are becoming more favored for real-time applications, because of better constant run-time guarantees.

5.3.3 Gradient Projection Method

The Gradient projection method utilizes the steepest descent, a step in the negative gradient direction at each iteration, projected onto the feasible set. While projection onto the feasible set is computationally expensive for the general type of constraints, it can be obtained without difficulty for box constraints using element-wise median with the bounds of the box. Since only the gradient of the problem is used to determine the step direction, the method converges slowly for *ill-conditioned* problems due to an uneven scaling of the optimization criterion.

5.4 Solvers for Constrained MPC

This section provides an overview of the available state-of-the-art QP solvers and justifies the solver selection for the controller implementation. Decision criteria are not only the computational performance, but also licensing, supported interfaces to

Matlab and Simulink, interface for time-varying systems, and quality of the documentation. Only solvers specialized for MPC are considered. Utilizing the particular problem structure, these outperform the general QP solvers in the MPC application. A recent overview of solvers for *embedded optimization* (running on hardware with limited resources and with real-time computational time) is given in [64]. For a detailed performance comparison of the solvers, refer to [65, 66]. As licensing of the software is an important aspect for the possible industrial use, a license with free modification, proprietary distribution, and commercial use is preferred for prototyping, such as the GNU Lesser General Public License (GNU LGPL) [67].

5.4.1 FiOrdOs

FiOrdOs, First Order Optimization Software [68], is a Matlab toolbox for generation of C implementations of the gradient projection methods. While general inequality constraints are supported, only box constraints on the optimization variables are handled efficiently. The computation time of gradient methods depends on the Hessian condition number. The servomechanism problem formulation would require prescaling to improve the Hessian conditioning due to the unstable dynamics. The package is released under GNU GPL 3 license.

5.4.2 CVXGEN

CVXGEN [69, 70] by Mattingley and Boyd offers automated generation of optimized C code from a high-level description of the optimization problem, utilizing *primal-dual interior point* method to find the solution. Generated code is best suited for smaller QPs, approximately with less than 50 decision variables. The size of the generated branching-free and library-free code may be restrictive for embedded application with larger problem size. The code must be regenerated when, for example, problem size is changed, which is less convenient for fast prototyping. A free license is granted for academic use only.

5.4.3 FORCES

Similar to CVXGEN, FORCES (Fast Optimal Real-time Control on Embedded Systems) package [71, 72] produces automatically generated code, featuring interior point methods with Newton step, specialized to the MPC problem structure. Moreover, it offers interfaces to Matlab and Simulink, and LTV systems are supported. The package was utilized for the MPC-based motion control of the AMZ Driverless Formula [73, 74] and for 1:43 scale RC cars racing in [43]. The package offers some of the best computational performance in the recent benchmarks [65], and, compared to CVXGEN, scales better with problem size [66]. However, a free license is granted for non-commercial academic research only.

5.4.4 qpDUNES

Specialized quadratic programming solver qpDUNES [75, 76] utilizes Dual Newton Strategy, which combines the benefits in terms of structure exploitation of the interior point methods and warm-starting capabilities of the active set methods. The package is written in plain C with an interface to Matlab and support for LTV problems. In [75], a superior performance over FORCES was reported. The package is published under the terms of GNU LGPL, free for both academic and industrial use.

5.4.5 qpOASES

Solver qpOASES [77] extends the online active set strategy [78], and utilizes warm starting based on the assumption that the active set does not change significantly between the consecutive control steps. When the assumption is valid, it provides very fast computational times [65], especially for small and medium-sized problems, below 100 decision variables. The package is written in C++ and extensively documented, including the interfaces to both Matlab and Simulink. LTV systems are supported. The solver is optimized for the dense formulation, with an interface as in Eq. (5.24). The package is distributed under the industry-friendly GNU LGPL license.

5.4.6 HPMPC

HPMPC, the library for High-Performance implementation of solvers for MPC [79] utilizes an interior point method with Riccati-based solver [80]. Based on [81], HPMPC is the highest-performing solver for QPs with the structure arising from MPC. It supports LTV system description in the sparse formulation, offers code optimizations for embedded target architectures, and is published under the GNU LGPL license. However, a detailed documentation and a Simulink interface is missing.

5.4.7 Solver Package Selection

Based on the above survey, qpOASES was selected for implementing the controller as it offers fast computational times, interfaces to Matlab and Simulink, extensive documentation, and is published under the GNU LGPL license. HPMPC or qpDUNES are viable second options, as they utilize the sparse formulation, which is easier to implement. However, the documentation is not as detailed, compared to qpOASES, and the two packages are developed and supported only under Linux.

5.5 Constraint Handling

While MPC supports a systematic account of constraints, these shall be imposed with extra care as the problem complexity is increased and special measures must be taken to ensure the problem feasibility. An MPC controller is only defined when the set of possible solutions is non-empty. Otherwise, the problem is *infeasible* and the resulting control may be arbitrary.

The two main causes of constraint violation are unrealistic control objectives and model-plant mismatch [60]. In the first case, the constraints are mutually inconsistent. An example is demanding a quick change in position using position constraints together with an overly restrictive velocity constraint. In the second case, the future behavior of the system differs from the prediction. When the system is operating at the boundary of a constraint, the actual state response may end up violating the constraint as a result of model mismatch, including the effects of unmodeled disturbances, or measurement noise. The controller may be unable to find a feasible control input that would bring the state inside the constraints in a single time step, and the problem becomes infeasible.

5.5.1 Constraint Types

Two types of constraints are distinguished: *hard* and *soft*. Some level of violation is allowed only for soft constraints. The constraint boundary may be shifted if the feasible set of the original hard-constrained problem was empty.

Input constraints such as actuator limitations are usually *hard*. Variables directly manipulated by the optimization (the input sequence) can have hard constraints imposed as these can always be satisfied. If both input rate and input level constraints are imposed, the bounds must be selected consistently over the horizon.

State constraints arise from restrictions of the allowed or preferred operating range. The predicted state response depends on the initial state which is not manipulable in the optimization. Therefore, hard state constraints cause infeasibility problems and only soft state constraints shall be imposed.

5.5.2 Softening the Constraints

Soft constraints are formulated by introducing an additional optimization variable, called the *slack variable*, which represents the amount of constraint violation and is simultaneously minimized. We reformulate the *primal dense* formulation of MPC from Eq. (5.24) with soft constraints as

$$\min_{\bar{\mathbf{u}},\,\boldsymbol{\epsilon}} \quad \frac{1}{2}\,\bar{\mathbf{u}}^\top \hat{\mathbf{H}}\bar{\mathbf{u}} + \hat{\mathbf{g}}^\top \bar{\mathbf{u}} + w\|\boldsymbol{\epsilon}\|_2 + v\|\boldsymbol{\epsilon}\|_1 \tag{5.31a}$$

$$\text{s.t.} \quad \mathbf{b}_{\text{lb}} - \mathbf{M}\boldsymbol{\epsilon} \leq \hat{\mathbf{B}}\bar{\mathbf{u}} \leq \mathbf{b}_{\text{ub}} + \mathbf{M}\boldsymbol{\epsilon}, \tag{5.31b}$$

$$\bar{\mathbf{u}}_{\text{lb}} \leq \bar{\mathbf{u}} \leq \bar{\mathbf{u}}_{\text{ub}}, \tag{5.31c}$$

$$\boldsymbol{\epsilon} \geq \mathbf{0}, \tag{5.31d}$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^{n_\epsilon}$ is the vector of slack variables and $\mathbf{M} \in \mathbb{R}^{n_{\text{iq}} \times n_\epsilon}$ is a matrix mapping the slack variables to the individual constraints, often an identity. Parameters w, v determine constraint weights, with $w \geq 0$ and $v > 0$.

Some small w shall be used to maintain a smooth cost function. Value of v is more critical as it determines whether the constraints are *exact*. An exact soft constraint is violated only if no other option exists. The cost of constraint violation cannot be outweighed by the control performance cost. An *inexact* soft constraint can be violated unnecessarily, in a tradeoff for a lower value of the combined cost criterion. To obtain an exact constraint, slack variable penalization cannot be purely quadratic with $v = 0$, as such would always result in some violation when active [56, pg. 98]. A linear cost must be introduced. To obtain exact constraints, the value of v must be selected to be higher than v^*, the largest of the Lagrange multipliers of the original hard-constrained problem at the optimum [56, pg. 99]. However, v^* is not precisely known apriori, especially in time-varying problems. Usually, a sufficiently large value is selected based on simulations.

Considering the above arguments, it may seem that all state constraints shall be softened. However, that would significantly increase the number of optimization variables and the computational time. To limit the number of variables, constraints are often reformulated using l^∞ norm. A single slack variable is used, representing the worst case violation for a given state or output over the whole prediction horizon. Matrix $\mathbf{M}$ maps the same slack variable to multiple scalar constraints, reducing the required n_ϵ. However, an equal level of violation is permitted along the whole prediction horizon once the slack variable becomes nonzero. A combination of both approaches may be used to provide more granularity, splitting the prediction horizon into subintervals, each covered by one slack variable. Particular applications of softened state constraints are discussed in Section 6.5, 7.5, and 8.5.

5.6 Move and Preview Blocking

A common strategy to reduce the computational complexity is *move blocking* [82, 83]. It reduces the number of input changes (moves) over the prediction horizon, for example by repeating an input value multiple times in few consecutive time steps of the horizon. The size of the optimization variable and the required computational time is reduced. A move blocking strategy can be described by a mapping

$$\bar{\mathbf{u}} = \bar{\mathbf{U}}\bar{\mathbf{u}}_\mathrm{r}, \tag{5.32}$$

where $\bar{\mathbf{U}} \in \mathbb{R}^{n_u N_\mathrm{p} \times n_u N_\mathrm{c}}$ projects the reduced input moves $\bar{\mathbf{u}}_\mathrm{r} \in \mathbb{R}^{n_u N_\mathrm{c}}$ to the original input sequence covering the whole prediction horizon $\bar{\mathbf{u}} \in \mathbb{R}^{n_u N_\mathrm{p}}$. The number of input degrees of freedom is denoted N_c.

A simple move blocking strategy for the tracking formulation is based on using input rates in the optimized control input sequence. Assuming that the system is approaching a steady state towards the end of the prediction horizon, the input level shall also approach its steady-state value and no further change shall be required. Therefore, the input rates towards the end of the horizon can be fixed to zero by using a move blocking matrix

$$\bar{\mathbf{U}} = \begin{bmatrix} \mathbf{I}_{N_\mathrm{c} \times N_\mathrm{c}} \\ \mathbf{0}_{\bar{N}_\mathrm{c} \times N_\mathrm{c}} \end{bmatrix} \otimes \mathbf{1} = \left[\begin{array}{cccc} \mathbf{1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{1} \\ \hdashline \mathbf{0} & \dots & \dots & \mathbf{0} \\ \vdots & & & \vdots \\ \mathbf{0} & \dots & \dots & \mathbf{0} \end{array}\right], \tag{5.33}$$

where $\otimes$ denotes the Kronecker product, N_c is the control horizon, $\bar{N}_\mathrm{c} = N_\mathrm{p} - N_\mathrm{c}$ denotes the number of steps during which $\mathbf{u} = \mathbf{0}$, and $\mathbf{1} \in \mathbb{R}^{n_u}$ is a vector of ones, sized according to the number of system inputs n_u.

To satisfy the steady-state assumption, the disturbance preview $\bar{\mathbf{d}}$ shall be clipped to zero after $N_\mathrm{w} \leq N_\mathrm{c}$ steps so that no change of the reference is commanded where the controller no longer has a control authority. A preview blocking matrix can be defined similarly as

$$\bar{\mathbf{D}} = \begin{bmatrix} \mathbf{I}_{N_\mathrm{w} \times N_\mathrm{w}} & \mathbf{0}_{N_\mathrm{w} \times \bar{N}_\mathrm{w}} \\ \mathbf{0}_{\bar{N}_\mathrm{w} \times N_\mathrm{w}} & \mathbf{0}_{\bar{N}_\mathrm{w} \times \bar{N}_\mathrm{w}} \end{bmatrix} \otimes \mathbf{1} \tag{5.34}$$

where $\mathbf{I}$ is an identity and $\mathbf{0}$ is a zero matrix of indicated sizes, with $\bar{N}_\mathrm{w} = N_\mathrm{p} - N_\mathrm{w}$, and $\mathbf{1} \in \mathbb{R}^{n_d}$. The primal-dense problem matrices are changed to

$$\hat{\mathbf{B}}_\mathrm{blk} \leftarrow \hat{\mathbf{B}}\bar{\mathbf{U}}, \qquad \hat{\mathbf{E}}_\mathrm{blk} \leftarrow \hat{\mathbf{E}}\bar{\mathbf{D}}, \qquad \bar{\mathbf{R}}_\mathrm{blk} \leftarrow \bar{\mathbf{U}}^\top \bar{\mathbf{R}}\bar{\mathbf{U}}. \tag{5.35}$$

Other possible move blocking strategies spread the control inputs over the prediction horizon in blocks of increasing length, offering some control authority even at the horizon end. However, move blocking is a heuristic technique. Stability guarantees developed for MPC are lost. By blocking control moves at some of the predicted steps, a further discrepancy arises between the receding horizon formulation and the infinite horizon optimal control problem, because an input change is eventually possible at every step of the receding horizon control. Possible negative impacts of move blocking on the performance have to be determined experimentally.

6 | Longitudinal Controller

This chapter introduces a controller of the vehicle longitudinal velocity. The controller is based on the modeling of Section 2.8 and formulated using the servomechanism problem structure described in Section 4.1. The goal of the longitudinal controller is to track a velocity trajectory that was designed to decouple the longitudinal and lateral dynamics. The model used in the independent lateral motion controller is parametrized with the trajectory of the desired longitudinal velocity, and therefore a low tracking error in the longitudinal velocity is necessary for the validity of the approximation. Moreover, accurate velocity tracking helps to ensure that the maximum allowed level of lateral acceleration is not exceeded during cornering, by braking to the required cornering speed in advance.

6.1 Controller Structure

For the controller design, the desired velocity trajectory is approximated as a piecewise ramp signal. The objective is to track the trajectory while satisfying the constraints of the longitudinal acceleration and of its time derivative, the longitudinal jerk. Constraint satisfaction is addressed separately in Section 6.5.

The reference generator is modeled as a double integrator, the model of ramp reference signals. Reference velocity and acceleration values are set using the initial condition. Input u_2 of the reference generator system is the time derivative of the reference acceleration, approximated using forward difference of the reference acceleration trajectory. The model of the controlled longitudinal dynamics is summarized in Eq. (2.27). Control input u_1 used in the optimization is the time derivative of the commanded longitudinal acceleration. The controller incorporates a model of the reference, resulting in zero steady-state tracking error of ramp reference signals in velocity with zero steady-state value of the control input variable u_1. Continuous-time model of the system is

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} v_{\mathrm{act}} \\ a_{\mathrm{act}} \\ a_{\mathrm{cmd}} \\ v_{\mathrm{ref}} \\ a_{\mathrm{ref}} \end{bmatrix} = \underbrace{\left[\begin{array}{ccc:cc} 0 & 1 & 0 & 0 & 0 \\ 0 & -\tau_{\mathrm{L}}^{-1} & \tau_{\mathrm{L}}^{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right]}_{\mathbf{A}} \underbrace{\begin{bmatrix} v_{\mathrm{act}} \\ a_{\mathrm{act}} \\ a_{\mathrm{cmd}} \\ v_{\mathrm{ref}} \\ a_{\mathrm{ref}} \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{B}} u_1 + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{\mathbf{E}} u_2, \quad (6.1)$$

where subscripts are used to distinguish between actual, commanded and reference values of the longitudinal velocity and acceleration. Since all variables are related to the longitudinal direction, x index is omitted. Velocity tracking error is given as

$$e_v = \underbrace{\left[\begin{array}{ccc:cc} 1 & 0 & 0 & -1 & 0 \end{array}\right]}_{\mathbf{C}} \mathbf{x}. \quad (6.2)$$

The continuous-time LTI model Eq. (6.1) is discretized at $f_\mathrm{s} = 20\,\mathrm{Hz}$ using exact discretization with input zero-order hold, assuming constant jerk between the time steps. Prediction time is given as $T_\mathrm{p} = N_\mathrm{p}/f_\mathrm{s}$, where N_p is the horizon length.

6.2 Performance Criterion

Using the discretized version of the state-space model Eq. (6.1), discrete-time controllers are designed using LQ servomechanism and MPC techniques, both with a quadratic performance criterion. As the controller is tuned for a race track application, the vehicle is expected to undergo high levels of acceleration to track the prescribed velocity trajectory. Therefore, the acceleration level is not penalized, but acceleration level constraints are imposed later.

The cost function at each step is

$$L_k(\mathbf{x}_k, \mathbf{u}_k) = q e_v^2(k) + r \dot{a}_\mathrm{cmd}^2(k) = \mathbf{x}_k^\top \underbrace{q \mathbf{C}^\top \mathbf{C}}_{\mathbf{Q}} \mathbf{x}_k + \dot{a}_\mathrm{cmd}^2(k). \tag{6.3}$$

In the single input case, parameter r penalizing the control effort can be fixed $r = 1$, as only the ration of $\mathbf{Q}$ to r determines the performance. Only a single parameter is used to tune the performance: q determining the importance of accurate velocity tracking. Increasing q places more emphasis on close tracking of the velocity trajectory, making the controller more aggressive. Tuning the controller for a different application, such as for a passenger vehicle, reducing the weight q would increase the ride comfort, smoothing the velocity trajectory and reducing the rate of transition from acceleration to braking and conversely.

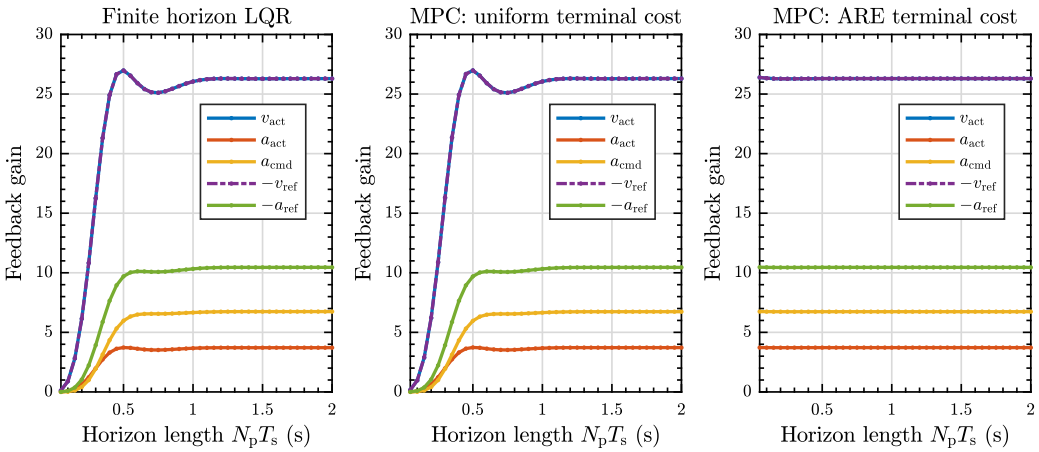
6.3 Unconstrained Controller

As described in Section 4.3 and Section 5.2, both the LQ servomechanism and the unconstrained MPC formulation consist of a state feedback. Receding horizon control applies only the first input of the optimal sequence and shifts the end of the horizon after each step. In the finite horizon LQR, the end of the horizon is assumed fixed and time-varying gains are precalculated for each time step. However, when the same weighting matrices are used in the quadratic costs Eq. (4.4) and Eq. (5.10) and the horizon length is increased, the feedback gains at the beginning of the horizon converge to identical values, as shown in Fig. 6.1.

Gains obtained from iterating Eq. (4.6) are shown in Fig. 6.1a. When used for time-varying state feedback, LQR gains would be applied in the reversed order of the horizon time indicated by the horizontal axis and would change only in the last few time steps. Solutions of the difference Riccati equation converge to the solution of the algebraic Riccati equation of the infinite horizon problem.

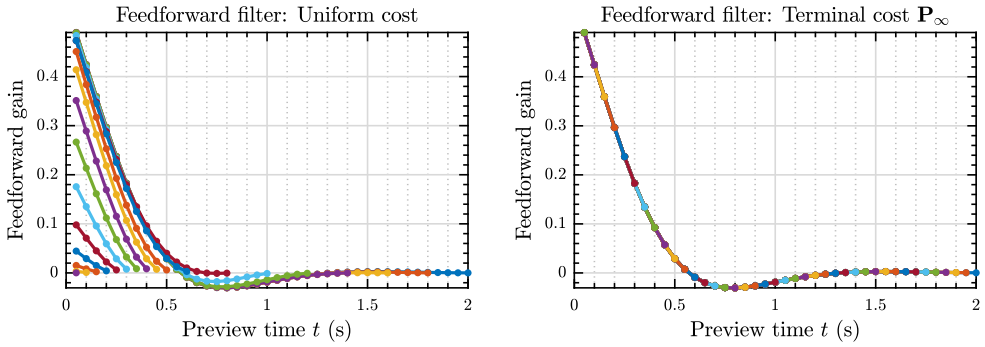
Convergence of the unconstrained MPC can be ensured either by a sufficiently long prediction horizon as in Fig. 6.1b, or by an appropriate setting of the terminal cost weight. Using the steady-state solution $\mathbf{P}_\infty$ of the Riccati Eq. (4.6) as the terminal cost $\mathbf{Q}_{n+N_\mathrm{p}}$ in the unconstrained MPC problem ensures convergence of the feedback gains regardless of the prediction horizon length N_p, as shown in Fig. 6.1c.

Fig. 6.2 analyzes the feedforward part $\mathbf{F}$ of the unconstrained MPC, calculated using Eq. (5.30). When weighting matrices of the cost function are uniform over the prediction horizon, the shape of the filter coefficients evolves with increasing horizon length as in Fig. 6.2a. For $T_\mathrm{p} \geq 0.5\,\mathrm{s}$, the filter coefficients start to converge,



(a) LQR (b) MPC: $\mathbf{Q}_{n+N_p} = \mathbf{Q}$ (c) MPC: $\mathbf{Q}_{n+N_p} = \mathbf{P}_\infty$

Figure 6.1: Convergence of state feedback gains with increasing horizon length for longitudinal velocity controller Eq. (6.1) discretized at $f_s = 20$ Hz.



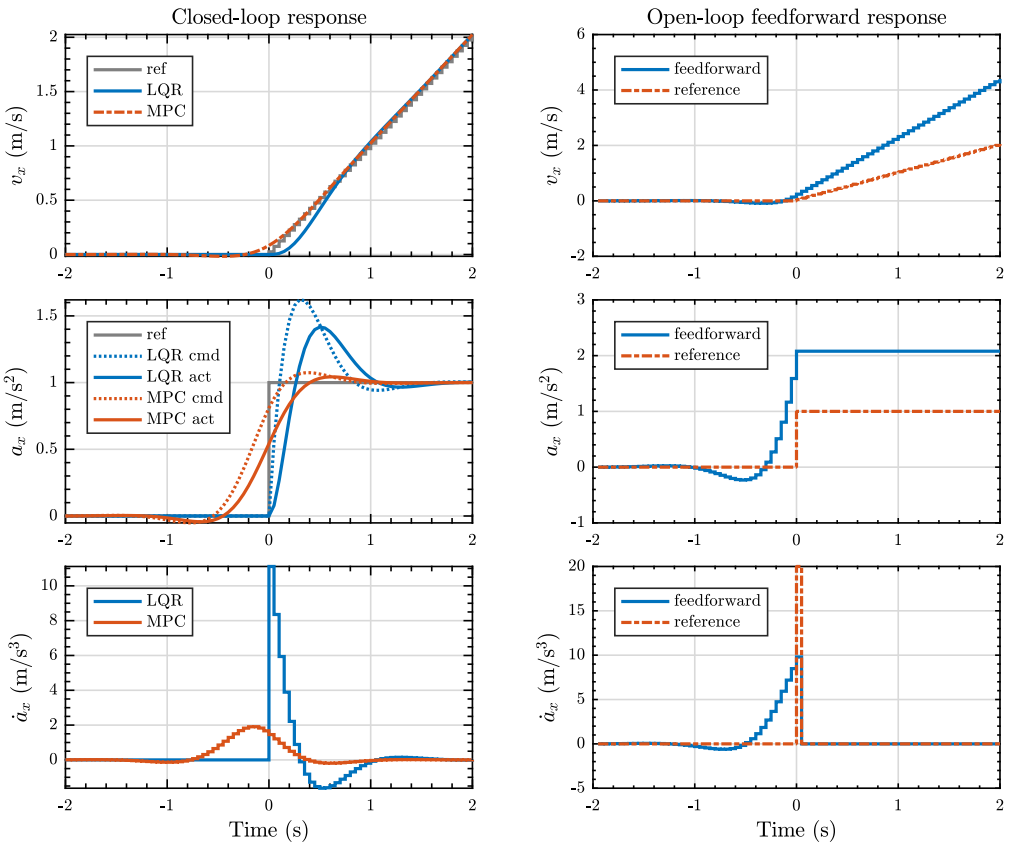
(a) with $\mathbf{Q}_{n+N_p} = \mathbf{Q}$ (b) with $\mathbf{Q}_{n+N_p} = \mathbf{P}_\infty$

Figure 6.2: Shape of feedforward coefficients $\mathbf{F}$, the gains from disturbance $\mathbf{d}$ previewed at time t in the future, obtained for $q = 10^3$, $f_s = 20$ Hz.

at a similar horizon length as the feedback part in Fig. 6.1b. For $T_p \geq 1.5$ s, the shape does no longer change and any further coefficients are negligible. In the unconstrained case, it makes no performance improvement to preview the reference further into the future. For $T_p \geq 0.8$ s, some of the coefficients become negative, meaning that the controller would initially command acceleration in the opposite direction to achieve closer velocity tracking. This behavior is a result of penalizing only the velocity error using $q = 10^3$ and no penalization of the acceleration error.

Fig. 6.2b shows the feedforward part $\mathbf{F}$ obtained with the steady-state solution $\mathbf{P}_\infty$ of Eq. (4.6) as the terminal weight. The shape of the filter does not change from the result obtained for an infinitely long prediction horizon. Horizon length determines only the number of nonzero coefficients, clipping the filter at $t = T_p$.

Obtaining the same solution in the unconstrained MPC as in LQR is essential because both controllers share the same stability guarantees. $\mathbf{P}_\infty$ in the terminal cost is the sum of the remaining cost-to-go of the infinite horizon problem. Therefore, the controller behaves as if there was LQ-optimal state feedback applied past the prediction horizon. Setting the terminal cost improves stability and reduces the discrepancy between the predicted input sequence and its receding horizon im-



(a) closed-loop responses

(b) MPC feedforward response

Figure 6.3: Comparison of LQR and MPC longitudinal velocity controllers in response to a unit step of reference acceleration.

plementation, making the calculated input sequence more useful for state response predictions. When constraints are imposed, further conditions must be satisfied to guarantee stability: the recursive feasibility as described in [53].

6.4 Importance of Preview

This section illustrates the importance of preview and benefits of the predictive control strategy. In particular, the impact of the preview-based feedforward $\mathbf{F}$ is analyzed. The controllers are implemented in discrete time at $f_\mathrm{s} = 20$ Hz and system response is simulated using the continuous-time model. Prediction time of $T_\mathrm{p} = 2$ s, resp. prediction horizon length of $N_\mathrm{p} = 40$ samples, was used.

Comparison of closed-loop responses to a unit step of the reference acceleration is shown in Fig. 6.3a. Both configurations achieve zero steady-state tracking error of ramp reference velocity. Utilizing the preview, MPC achieves a lower velocity error during the change of reference, with a lower peak in the longitudinal acceleration, and significantly lower and smoother longitudinal jerk. The lag due to the first-order response from the acceleration command to the actual acceleration is accounted for in the predictive control and acceleration is commanded further in advance.

The response of the feedforward filter to the same reference acceleration step is shown in Fig. 6.3b. In the lowermost plot, the unit impulse response of the feedforward input longitudinal jerk is shown. The shape of the feedforward input is

a mirror image of the coefficients plotted in Fig. 6.2b. Upon integration, the resulting acceleration and velocity without feedback are higher, as shown in the two upper plots of Fig. 6.3b. A larger input value has to be generated in the feedforward part as the feedback part is opposing the change before the reference changes for $t < 0$. The feedforward gain is higher than unity and depends on the feedback gains. Using MPC, the feedforward part is designed consistently with the particular values of LQR feedback gains. The feedforward from the previewed disturbance results in optimal transitions on reference changes.

6.5 Longitudinal Constraints

Imposing constraints is critical for the decoupled operation of the longitudinal and lateral controller. The total acceleration of the vehicle is limited, and the allocation of acceleration between the longitudinal and lateral directions is handled in the trajectory generation problem. Lateral motion has a higher priority to maintain the vehicle on the track. Therefore, the longitudinal controller must respect this allocation in the form of constraints to avoid coupling.

Constraining longitudinal acceleration during cornering helps to prevent *power oversteer*: sliding of the rear tires when too much throttle is applied in turn. Cornering forces of the rear tires are reduced due to the combined generation of tire force, increasing the vehicle sideslip angle. A similar effect also occurs during simultaneous braking and cornering. The front tires apply most of the braking force as a result of the forward weight transfer. Cornering forces of the front tires are reduced due to the combined force generation, producing a lower yaw rate and causing understeer: the vehicle executes an arc of larger radius.

Exceeding the maximum velocity may lead to an excessive lateral acceleration in corners and loss of tire grip. Depending on the setting of q, a velocity constraint may be necessary to slow down to the speed required in the corner and to prevent an early acceleration at the corner exit. Without the upper bound on velocity, the controller may relax its braking effort in a tradeoff for a lower penalization of the control effort as acceleration from the corner is anticipated.

To formulate constraints of the velocity error and the acceleration level, we first define the outputs to be constrained. The velocity error is defined in Eq. (6.2). The acceleration command is obtained as

$$a_{\mathrm{cmd}} = \underbrace{\left[\begin{array}{ccc:cc} 0 & 0 & 1 & 0 & 0 \end{array}\right]}_{\mathbf{C}_a} \mathbf{x}. \tag{6.4}$$

Formally, the output matrices are collected using the Kronecker product $\otimes$ into

$$\bar{\mathbf{C}} = \begin{bmatrix} \mathbf{I}_{N_\mathrm{c} \times N_\mathrm{c}} \otimes \mathbf{C} \\ \mathbf{I}_{N_\mathrm{c} \times N_\mathrm{c}} \otimes \mathbf{C}_a \end{bmatrix}, \tag{6.5}$$

where $\mathbf{I}$ is an identity matrix of indicated dimensions. Matrix $\bar{\mathbf{C}}$ projects the predicted states into a vector of predicted velocity errors and predicted acceleration commands over the first N_c steps, the part of the horizon with possible control action where we select to impose the constraints.

Longitudinal jerk constraints are implemented as simple box constraints of the optimization variable in the form of Eq. (5.24c). Together with the constraints of the acceleration command, both input rate and input level are constrained. Acceleration command bounds change along the trajectory and are precalculated during

the trajectory generation based on the available tire grip using track curvature and reference velocity. The bounds assume the form of

$$\begin{aligned} \mathbf{a}_{\mathrm{lb}} &= \Big[a_{\min}(t_n),\, a_{\min}(t_n + T_{\mathrm{s}}),\, \ldots,\, a_{\min}(t_n + T_{\mathrm{c}})\Big]^{\top}, \\ \mathbf{a}_{\mathrm{ub}} &= \Big[a_{\max}(t_n),\, a_{\max}(t_n + T_{\mathrm{s}}),\, \ldots,\, a_{\max}(t_n + T_{\mathrm{c}})\Big]^{\top}. \end{aligned} \tag{6.6}$$

Bounds of the acceleration command a_{cmd} can be formulated as hard, provided that the bounds on $\dot{a}_{\mathrm{cmd}}$ are sufficiently tolerant, and that the bounds on a_{cmd} from the previous step are only shifted in time with no abrupt change. For example with the setting of $|\dot{a}_{\mathrm{cmd}}| \leq 20\,\mathrm{m/s^3}$ and the sample time $T_{\mathrm{s}} = 0.05\,\mathrm{s}$, the command can change $|\Delta a_{\mathrm{cmd}}| \leq 1\,\mathrm{m/s^2}$ per control step, typically enough to follow the bounds. For larger changes, the controller can use multiple steps as the horizon is sufficiently long $T_{\mathrm{c}} = 2\,\mathrm{s}$. We did not encounter any infeasibility problems with the above settings in simulations.

Only an upper bound is imposed on the velocity error to prevent exceeding the reference velocity by more than v_ϵ. The bounds are

$$\mathbf{v}_{\mathrm{lb}} = -\infty\,\mathbf{1}, \quad \mathbf{v}_{\mathrm{ub}} \;\; = v_\epsilon\,\mathbf{1}, \tag{6.7}$$

where $\mathbf{1} \in \mathbb{R}^{N_{\mathrm{c}}}$ is a vector of ones. However, the velocity error constraint must be softened. The controller has to cope with model mismatch, such as reduced braking efficiency due to reduced friction coefficient, or unmodeled disturbances, such as the road grade. It is not possible to guarantee reduction of the velocity below the upper bound in a single step. Therefore, the parameter v_ϵ is used as a slack variable. For low complexity, a single slack variable is used, representing the worst case error over the whole horizon. The linear penalization weight v of the soft constraint was set to $v = 10^6$ so that the soft constraint is exact.

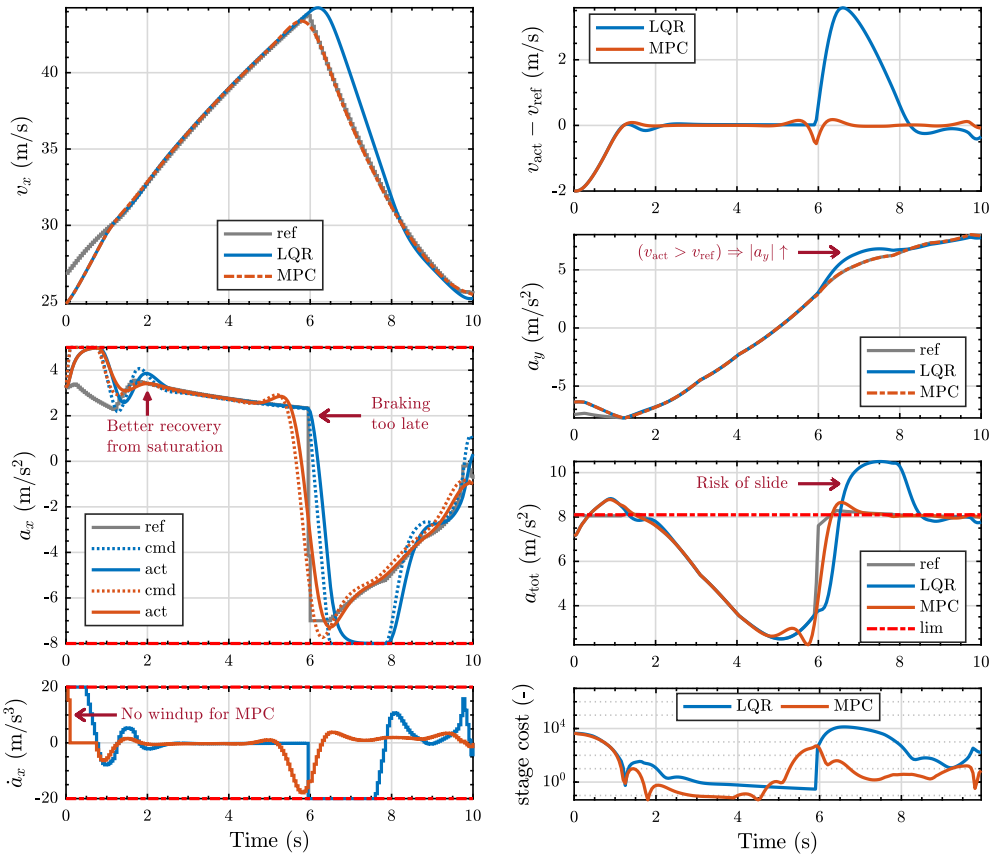
Finally, the constraints are collected into the form of Eq. (5.24b) as

$$\underbrace{\begin{bmatrix}\mathbf{v}_{\mathrm{lb}}\\ \mathbf{a}_{\mathrm{lb}}\end{bmatrix} - \bar{\mathbf{C}}\hat{\mathbf{A}}\mathbf{x}_n - \bar{\mathbf{C}}\hat{\mathbf{E}}\bar{\mathbf{d}}}_{\mathbf{b}_{\mathrm{lb}}} \leq \underbrace{\bar{\mathbf{C}}\hat{\mathbf{B}}}_{\mathbf{A}_{\mathrm{iq}}}\bar{\mathbf{u}} \leq \underbrace{\begin{bmatrix}\mathbf{v}_{\mathrm{ub}}\\ \mathbf{a}_{\mathrm{ub}}\end{bmatrix} - \bar{\mathbf{C}}\hat{\mathbf{A}}\mathbf{x}_n - \bar{\mathbf{C}}\hat{\mathbf{E}}\bar{\mathbf{d}}}_{\mathbf{b}_{\mathrm{ub}}}. \tag{6.8}$$

6.6 Performance on Time-Optimal Velocity Profile

This section compares the tracking performance of the infinite-horizon discrete-time LQR and constrained MPC using a realistic reference, a minimum-time velocity and acceleration trajectory, generated as described in Chapter 3. The selected portion represents a transition between two turns, with acceleration followed by braking and cornering. Controller parameters are set as in Subsection 6.4, with $f_{\mathrm{s}} = 20\,\mathrm{Hz}$ and $N_{\mathrm{p}} = 40$. For both controllers, longitudinal jerk was constrained to $|\dot{a}_{\mathrm{cmd}}| \leq 20\,\mathrm{m/s^3}$ and acceleration command to $a_{\mathrm{cmd}} \in \langle -8,\, 5\rangle\,\mathrm{m/s^2}$. While MPC handles the input limits explicitly as optimization constraints, saturation had to be applied in the LQR case. Typically, the performance of MPC in saturation is better. Applying saturation with LQR can induce command oscillations, as rate constraints are not considered, or even lead to instability.

Velocity tracking is shown in Fig. 6.4a, initialized with $v_{\mathrm{act}} = (v_{\mathrm{ref}} - 2)\,\mathrm{m/s}$. The response to the initial condition is identical. However, the control input generated by MPC accounts for constraints and does not windup during acceleration saturation. The recovery from saturation is less oscillatory in the case of MPC, both because of handling the constraints systematically and previewing the future reference.



(a) tracking (b) performance indicators

Figure 6.4: Comparison of LQR (reactive) and MPC (predictive) longitudinal controllers tracking a time-optimal velocity profile.

The major difference becomes apparent when the model transitions from acceleration to braking at time $t = 6\,\mathrm{s}$. The change is anticipated in MPC, braking is initiated earlier, resulting in a superior tracking performance. Moreover, the effect of the first-order response is accounted for and braking is commanded further in advance. Anticipating the change in reference reduces the tracking error. Moreover, the required control input is lower and smoother, thus less demanding on the vehicle.

The advantages of the predictive control are further highlighted in Fig. 6.4b. A lower velocity error translates into a lower deviation from the planned lateral acceleration in the turn. When the reference velocity is exceeded, the resulting centrifugal force is higher, demanding a higher tire cornering force to maintain the turning radius. Similarly, braking only in reaction to the velocity error requires a higher longitudinal acceleration to track the trajectory. Both these effects contribute to the total acceleration. The limit of the total acceleration, for which the trajectory was designed, was significantly exceeded in the LQR case, as indicated in the third subplot of Fig. 6.4b. Such controller reaction would likely result in saturation of tire forces and a slide. Therefore, the predictive control helps to maintain the total acceleration in the designed limits and prevents forcing the vehicle into the region of strongly nonlinear operation. In lateral motion control, anticipating corners helps similarly to reduce the lateral jerk and lateral acceleration. Therefore, utilizing a preview-based feedforward control is necessary to execute a trajectory at the limits of the vehicle handling with additional constraints not considered in the trajectory

optimization, such as the rate limits of the acceleration command.

Lastly, the performance of both approaches can be compared by evaluating the quadratic cost at each step. Although the same criterion is optimized, utilizing future information further helps to reduce the criterion value, as shown in the lowermost subplot of Fig. 6.4b. Anticipating the step change of acceleration, the predictive controller deviates slightly at first to obtain a lower overall penalization later during the braking part of the trajectory. The predictive controller performs better in terms of the quadratic cost, the tracking error, and satisfaction of the constraints.

7 | Lateral Controller

Lateral motion controller uses steering to track the reference path. The problem can be posed as model matching using the servomechanism problem structure described in Chapter 4, allowing to combine different models of the controlled system and the reference generator from Chapter 2 for particular tracking tasks.

7.1 Controller Formulation

To follow a reference path closely, the control system must keep the velocity vector of the vehicle aligned with the tangent of the reference curve. A sufficiently smooth curved path can be locally approximated using an osculating circle, which is tangent to the curve and of the same curvature at the given point of approximation. The distance of the vehicle from the curve in the radial direction of the osculating circle defines the crosstrack error and change in the relative heading of the vehicle is used to suppress it. Small crosstrack deviations are permitted, for example, to achieve reduced lateral acceleration and jerk, for a smoother ride experience in the situations where the risk of collision with an obstacle is not increased.

Consider situation depicted in Fig. 7.1 showing a bicycle model following a curved path in the xy plane. At the point closest to the vehicle, a tangential curve-fixed reference frame is attached, and the curve is approximated using the osculating circle. The crosstrack error e_d is governed by a differential equation, obtained as the projection of the vehicle velocity vector $\mathbf{v}$ from the body-fixed coordinate frame into the radial direction y_c, perpendicular to the tangential heading of the curve ψ_ref. The projection, representing the crosstrack error rate, is given as

$$\dot{e}_d = \dot{x}\sin(\psi - \psi_\mathrm{ref}) + \dot{y}\cos(\psi - \psi_\mathrm{ref}), \tag{7.1}$$

similar as assuming Eq. (2.6b) in the reference frame of the curve, called the *curvilinear abscissa reference frame* in the literature [84, 85].

As the vehicle is supposed to move along the reference curve closely, the difference in orientation $|\psi - \psi_\mathrm{ref}|$ is typically smaller than a 15° angle and e_d is small with respect to the reference radius $R = \kappa_\mathrm{ref}^{-1}$ at higher speeds. Therefore, crosstrack error rate $\dot{e}_d$ can be linearized for small heading errors as

$$\dot{e}_d \approx v_x(\psi - \psi_\mathrm{ref}) + \dot{y} \approx v_x(\psi + \beta - \psi_\mathrm{ref}) \approx v_x e_\psi. \tag{7.2}$$

For the kinematic model, lateral velocity component is often neglected $\dot{y} \approx 0$, as in [17], because no sideways slipping is assumed in the kinematic model of Section 2.2. The crosstrack error e_d is the integral of the heading error e_ψ. The purpose of the controller is to regulate e_d to zero and to keep headings of the controlled and the reference system aligned.

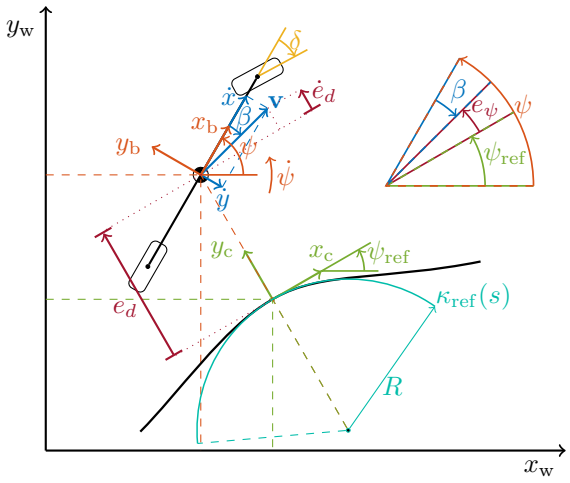


Figure 7.1: Coordinate frame convention used in the lateral motion controller. All angles are oriented positively in the counter-clockwise direction.

7.1.1 Kinematic Models

Several configurations of the servomechanism problem are possible, selecting different models to represent the heading response. The reference generator can either be formulated using Eq. (2.2) when a reference steering trajectory is known, for example from a higher-level planner utilizing the simpler kinematic model, or using Eq. (2.4) when the reference path is a geometric curve represented by curvature. The reference generator model of a C^2-continuous curve in the state-space form is

$$\begin{bmatrix} \dot{\psi}_\mathrm{ref} \\ \dot{\kappa}_\mathrm{ref} \end{bmatrix} = \begin{bmatrix} 0 & v_x \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \psi_\mathrm{ref} \\ \kappa_\mathrm{ref} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_2 = \mathbf{A}_{2,\psi}\mathbf{z}_2 + \mathbf{E}_2 u_2, \tag{7.3}$$

where $\mathbf{z}_2$ denotes the reference state and the input u_2 represents the rate of curvature change of the reference curve.

For control of maneuvers at low levels of lateral acceleration, the kinematic model of Eq. (2.2) can be used as the model of the controlled system, with an additional state introducing the continuity of the commanded steering angle δ

$$\begin{bmatrix} \dot{\psi} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} 0 & v_x/L \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \delta \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_1 = \mathbf{A}_{1,\psi}\mathbf{z}_1 + \mathbf{B}_1 u_1, \tag{7.4}$$

where $\mathbf{z}_1$ denotes the controlled state and the input u_1 is the rate of steering angle change. This setup provides tracking feedback. However, the ideal open-loop steering u_1 is already known from the input u_2 of the generator.

7.1.2 Dynamic Model

On the contrary, utilizing the dynamic model Eq. (2.20) leads to a model matching task. The controller provides not only a crosstrack error feedback but also determines the appropriate feedforward steering such that the reference curve heading is followed, utilizing the knowledge of the vehicle's velocity-dependent yaw rate and sideslip transient responses. The task is no longer kinematic.

State-transition Eq. (2.20) of the controlled system is rewritten as

$$\begin{bmatrix} \dot{\psi} \\ \ddot{y} \\ \dot{\psi} \\ \dot{\delta} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -\frac{C_{\alpha_\mathrm{r}}+C_{\alpha_\mathrm{f}}}{mv_x} & \frac{l_\mathrm{r}C_{\alpha_\mathrm{r}}-l_\mathrm{f}C_{\alpha_\mathrm{f}}}{mv_x} - v_x & \frac{C_{\alpha_\mathrm{f}}}{m} \\ 0 & \frac{l_\mathrm{r}C_{\alpha_\mathrm{r}}-l_\mathrm{f}C_{\alpha_\mathrm{f}}}{v_x} & -\frac{l_\mathrm{r}^2C_{\alpha_\mathrm{r}}+l_\mathrm{f}^2C_{\alpha_\mathrm{f}}}{I_z v_x} & \frac{l_\mathrm{f}C_{\alpha_\mathrm{f}}}{I_z} \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}_{1,\psi}} \underbrace{\begin{bmatrix} \psi \\ \dot{y} \\ \dot{\psi} \\ \delta \end{bmatrix}}_{\mathbf{z}_1} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{\mathbf{B}_1} u_1, \tag{7.5}$$

with the state $\mathbf{z}_1$ augmented by yaw rate $\dot{\psi}$ and lateral velocity $\dot{y} \approx v_x\beta$ of the dynamic bicycle model in contrast to the kinematic model of Eq. (7.4). The pair $(\mathbf{A}_{1,\psi}, \mathbf{B}_1)$ represents a model of the vehicle heading response and the dynamic bicycle of Eq. (7.5) is used in this work.

7.1.3 Velocity Projection

The third ingredient is how the state of each subsystem affects the resulting crosstrack error. Although Eq. (7.1) shall be integrated to obtain the crosstrack error, we first integrate the contribution of each subsystem separately and then subtract the integrals to comply formally with Eq. (4.12), where the errors are formulated using a linear combination of subsystem states. From Eq. (7.1), output matrix is introduced for the curve model as

$$\mathbf{C}_{d_2}(v_x) = \begin{bmatrix} v_x & 0 \end{bmatrix}, \tag{7.6}$$

for the kinematic bicycle model as

$$\mathbf{C}_{d_1}(v_x) = \begin{bmatrix} v_x & v_x\frac{l_\mathrm{r}}{L} \end{bmatrix}, \tag{7.7}$$

and for the dynamic bicycle model as

$$\mathbf{C}_{d_1}(v_x) = \begin{bmatrix} v_x & 1 & 0 & 0 \end{bmatrix}, \tag{7.8}$$

Defining additional system states d_1 and d_2 representing the contribution of each subsystem to the crosstrack error $e_d = d_1 - d_2$, the two subsystems are joined according to the structure of Eq. (4.12)

$$\left[\begin{array}{c} \dot{d}_1 \\ \dot{\mathbf{z}}_1 \\ \dot{d}_2 \\ \dot{\mathbf{z}}_2 \end{array}\right] = \left[\begin{array}{cc:cc} 0 & \mathbf{C}_{d_1} & 0 & \mathbf{0} \\ 0 & \mathbf{A}_{1,\psi} & 0 & \mathbf{0} \\ \hdashline 0 & \mathbf{0} & 0 & \mathbf{C}_{d_2} \\ 0 & \mathbf{0} & 0 & \mathbf{A}_{2,\psi} \end{array}\right] \left[\begin{array}{c} d_1 \\ \mathbf{z}_1 \\ \hdashline d_2 \\ \mathbf{z}_2 \end{array}\right] + \left[\begin{array}{c} 0 \\ \mathbf{B}_1 \\ \hdashline 0 \\ \mathbf{0} \end{array}\right] u_1 + \left[\begin{array}{c} 0 \\ \mathbf{0} \\ \hdashline 0 \\ \mathbf{E}_2 \end{array}\right] u_2. \tag{7.9}$$

The above block-diagonal structure is required for the LQ tracker design as described in Section 4.3. However, for MPC formulation the structure can also be written as

$$\begin{bmatrix} \dot{e}_d \\ \dot{\mathbf{z}}_1 \\ \dot{\mathbf{z}}_2 \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{C}_{d_1} & -\mathbf{C}_{d_2} \\ 0 & \mathbf{A}_{1,\psi} & \mathbf{0} \\ 0 & \mathbf{0} & \mathbf{A}_{2,\psi} \end{bmatrix} \begin{bmatrix} e_d \\ \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{B}_1 \\ \mathbf{0} \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ \mathbf{0} \\ \mathbf{E}_2 \end{bmatrix} u_2, \tag{7.10}$$

for model compatibility with the related work [17]. Further in this work, only the structure of Eq. (7.9) is used for the controller design.

7.1.4 Error Outputs

Several error outputs can be penalized. The most important one to ensure path tracking is the crosstrack error, the only output from which all controllable states are observable. Secondly, heading error can be penalized as it is the source of crosstrack error and low heading error ensures the validity of Eq. (7.1) linear approximation. Yaw rate and lateral acceleration errors allow tuning the aggressivity of the response, reducing the deviation from the planned trajectory in terms of lateral acceleration and demand of lateral tire force. In this order, outputs are introduced as follows

$$\underbrace{\begin{bmatrix} d_1 \\ \psi + \beta \\ \dot{\psi} \\ a_y \end{bmatrix}}_{\mathbf{y}_1} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & {v_x}^{-1} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & a_{21} & a_{22} & a_{23} + v_x & a_{24} \end{bmatrix}}_{\mathbf{C}_1(v_x)} \underbrace{\begin{bmatrix} d_1 \\ \psi \\ \dot{y} \\ \dot{\psi} \\ \delta \end{bmatrix}}_{\mathbf{x}_1}, \tag{7.11}$$

for the controlled system based on the dynamic bicycle model, with a_{2j} indicating the j-th column of the second row of $\mathbf{A}_{1,\psi}$ from Eq. (7.5), forming a_y as in Eq. (2.9). Outputs of the generator based on the model of a curve are

$$\underbrace{\begin{bmatrix} d_2 \\ \psi_{\mathrm{ref}} \\ \dot{\psi}_{\mathrm{ref}} \\ a_{y,\mathrm{ref}} \end{bmatrix}}_{\mathbf{y}_2} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & v_x \\ 0 & 0 & v_x^2 \end{bmatrix}}_{\mathbf{C}_2(v_x)} \underbrace{\begin{bmatrix} d_2 \\ \psi_{\mathrm{ref}} \\ \kappa_{\mathrm{ref}} \end{bmatrix}}_{\mathbf{x}_2}. \tag{7.12}$$

The error output is formed as $\mathbf{e} = \mathbf{y}_1 - \mathbf{y}_2$ with an output matrix $\mathbf{C} = [\mathbf{C}_1,\ -\mathbf{C}_2]$.

7.2 Controller Tuning

With five states of the controlled system, three states of the reference generator and linear parameter-varying dynamics, implementation and tuning of the lateral controller is more a complex task than was the case of the longitudinal controller. Both the state-transition matrix $\mathbf{A}(v_x)$ and the output matrix $\mathbf{C}(v_x)$ of the servomechanism structure depend on the longitudinal velocity. Therefore, a steady-state solution of the difference Riccati equation exists only for v_x fixed. In contrast, the MPC formulation can be parametrized with a velocity trajectory over the prediction horizon to provide optimal control of the parameter-varying system.

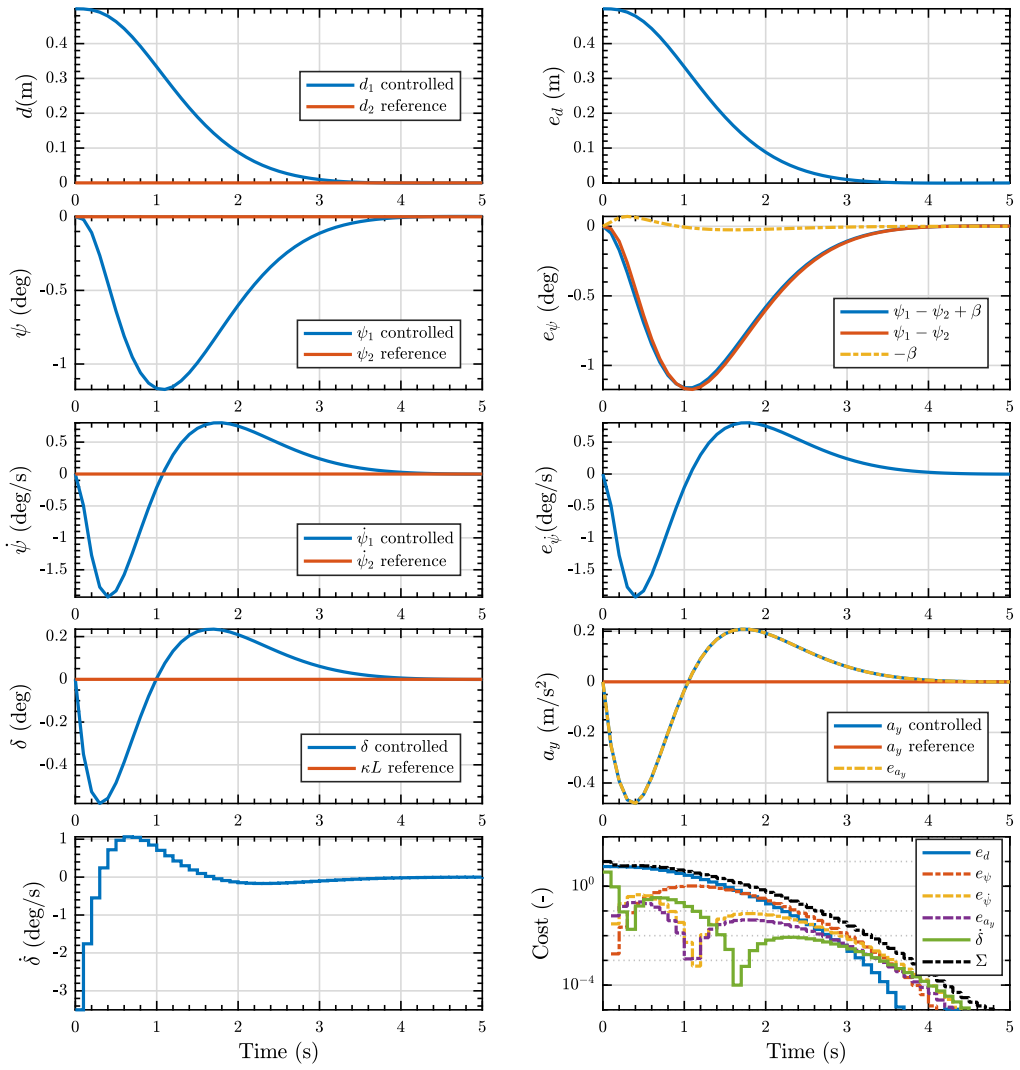
For the the MPC formulation, the criterion weights can vary with the velocity similarly as the system dynamics as long as $\mathbf{Q}_\mathbf{x} \succeq 0, \mathbf{R} \succ 0$. However, this option was not investigated for the sake of controller tuning simplicity. A fixed, velocity-independent diagonal matrix of tracking error weights $\mathbf{Q}_\mathbf{e}$ was used

$$\mathbf{Q}_\mathbf{e} = \mathrm{diag}\left(\begin{bmatrix} 0.25, & 25, & 4, & 0.01 \end{bmatrix}\right), \qquad \mathbf{R} = 10. \tag{7.13}$$

with the errors ordered as $\mathbf{e} = [e_d,\ e_\psi,\ e_{\dot{\psi}},\ e_{a_y}]^\top$. The choice of $\mathbf{Q}_\mathbf{e}$ and $\mathbf{C}(v_x)$ defines the matrix $\mathbf{Q}_\mathbf{x}$ of weights in terms of the system states

$$\mathbf{Q}_\mathbf{x} = \mathbf{C}^\top(v_x)\, \mathbf{Q}_\mathbf{e}\, \mathbf{C}(v_x), \tag{7.14}$$

which is velocity-dependent, non-diagonal due to the squared differences between states, but positive semidefinite. The symbolic structure of $\mathbf{Q}_\mathbf{x}$ is rather complicated and thus is kept in the unexpanded form.



(a) State response

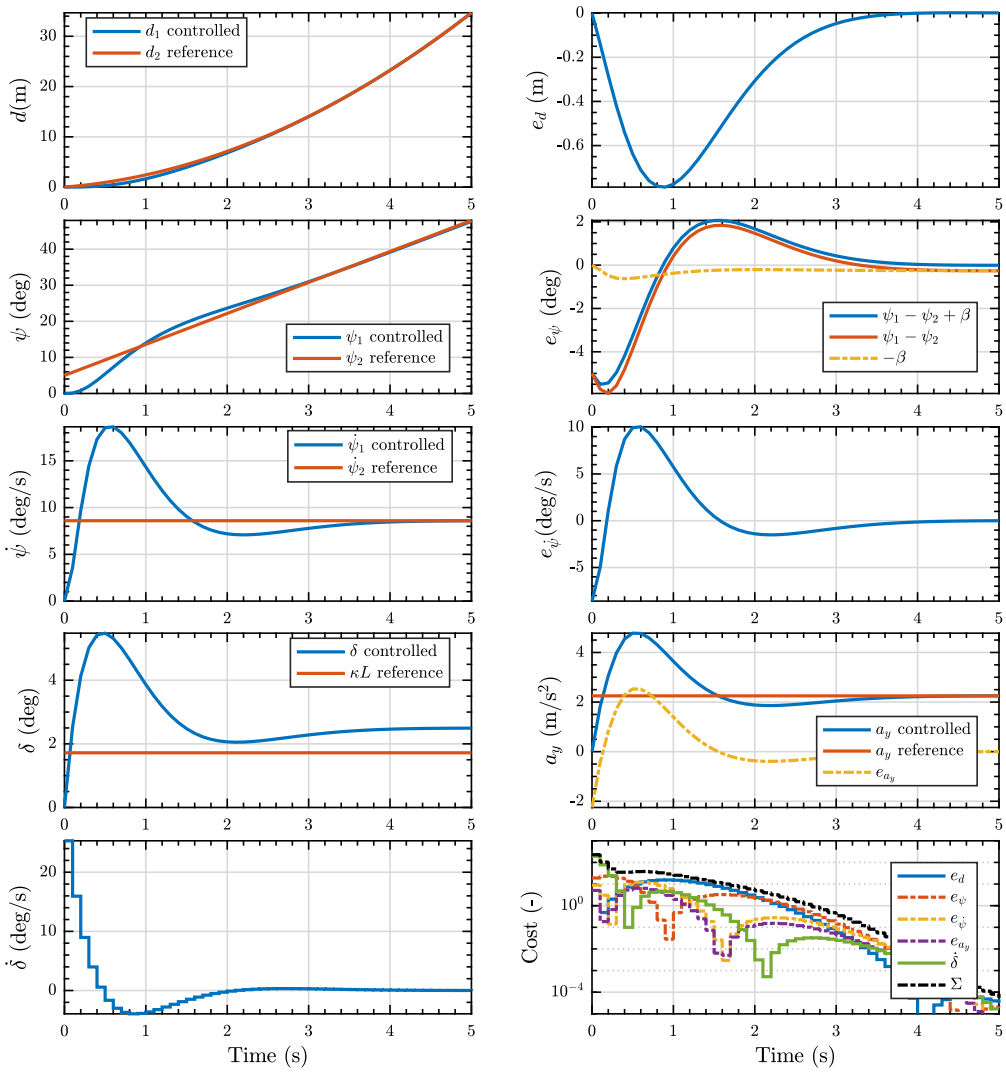
(b) Error response

Figure 7.2: Response of the unconstrained lateral state-feedback controller to an initial crosstrack error $e_d = 0.5$ m at velocity $v_x = 15$ m/s.

7.2.1 Response to Initial Conditions

First, the response of the structure to initial conditions is analyzed in the unconstrained case and at fixed $v_x = 15$ m/s, where the controller acts as a linear state feedback. The feedback gains can be equivalently obtained as the steady-state solution of the difference Riccati Eq. (4.6) or as the solution of the unconstrained MPC from Eq. (5.30). Response to an initial crosstrack error $e_d = d_1 - d_2 = 0.5$ m is shown in Fig. 7.2. State responses are shown in Fig. 7.2a, while errors introduced in Eq. (7.11) are shown in Fig. 7.2b, together with the weighted quadratic costs.

At each step, the controller reduces the cost of the current state as the output error is regulated to zero. Relative weights of the errors are essential for tuning the controller behavior. The controller has to develop an error in all of the output variables to steer the system to zero and the criterion weights determine the tradeoff. To illustrate the relation, the heading error e_ψ grows until $t = 1.0$ s, the controller is pointing the vehicle towards the path and the crosstrack error e_d starts decreasing.



(a) State response (b) Error response

Figure 7.3: Response of the unconstrained lateral state-feedback controller to initial heading error $e_\psi = -5°$ at velocity $v_x = 15\,\mathrm{m/s}$ and reference radius of curvature $\kappa^{-1} = 100\,\mathrm{m}$.

At $t = 1.2\,\mathrm{s}$, the two weighted quadratic errors become of the same order and the controller starts to realign the vehicle with the path heading.

The servomechanism structure of the lateral controller was designed to achieve a zero steady-state crosstrack error on circular paths, which translates into offset-free tracking of ramp reference in ψ and of parabolic reference in d_1. Tracking a circular path with $\kappa_{\mathrm{ref}}^{-1} = R = 100\,\mathrm{m}$ at velocity $v_x = 15\,\mathrm{m/s}$ is shown in Fig. 7.3. The response is started with an additional heading error, which increases the crosstrack error peak, before the required curvature is reached. The weights of $e_{\dot{\psi}}$ and e_{a_y} limit aggressivity of the response at the beginning of the transient, respectively the deviation from the reference yaw rate and lateral acceleration. The absolute levels of $\dot{\psi}$ and a_y are not penalized as achieving non-zero reference values is necessary for tracking, but exceeding the preplanned reference may lead to a loss of tire adhesion.

In the steady state, zero tracking errors are achieved in Fig. 7.3b, which justifies the choice of the tracking error outputs. Similarly, the control input and the criterion

value reaches zero in steady state. In the second row of Fig. 7.3b, vehicle yaw angle ψ does not align with the path heading $\psi_2 \neq \psi_1$, the vehicle is moving along the circular path at a sideslip angle $\beta \neq 0$, which the heading error e_ψ has to reflect in order to be zero in the steady state.

As in the first case, the response has initial peaks of $\dot{\delta}$, δ, $\dot{\psi}$ in Fig. 7.3a, amplified by the growing reference. A relatively large input is developed in the first time steps to avoid the state penalization in the future. The controlled system can be viewed as a cascade of an integrator, a second-order underdamped system, and a double integrator. From the control theory perspective, the single overshoot in δ and ψ responses indicates a zero at the origin in the transfer function from the reference generator states to the states of the controlled system once the state feedback is closed. To prevent an abrupt reaction, input constraints will be imposed to resolve the initial peak of the control input.

7.2.2 Lateral Tracking with Preview

This section presents the performance of lateral tracking when preview of the disturbance $\dot{\kappa}_{\text{ref}}$ is utilized. The disturbance represents the change of reference curvature. Previewing becomes essential when the controller is operating close to the system limits, either in terms of the control input or state responses. With the preview, the transition to a new reference can be spread over a time interval, respecting the system limitations. In this respect, the effect of the preview is similar to a reference governor, which smoothens the reference and ensures that an abrupt change that could destabilize the system is not commanded. In the MPC formulation, the reference change is predicted based on the apriori known disturbance entering the reference generator and the controller selects an input to match the anticipated response optimally with respect to the quadratic cost. The preview scheme is possible only because the disturbance $\dot{\kappa}_{\text{ref}}$ is known in advance based on the desired trajectory and the non-causal filtration can be realized.

The behavior is analyzed using a step in the reference curvature to a value at which the lateral acceleration is $a_y = 4\,\text{m/s}^2$, as in the steepsteer identification of Section 2.5. The experiment is simulated at two different speeds, $v_x = 15\,\text{m/s}$ and $v_x = 50\,\text{m/s}$, to compare how the performance changes in the presence of reduced damping and non-minimum phase β response at higher speeds with the velocity-independent weighting $\mathbf{Q}_\text{e}$ of Eq. (7.13). The variation of the model transient response with speed was analyzed in Fig. 2.7, showing that damping of β and $\dot{\psi}$ is reduced and the period of oscillations is increased with increasing speed, possibly prolonging the horizon length required for the system settling.

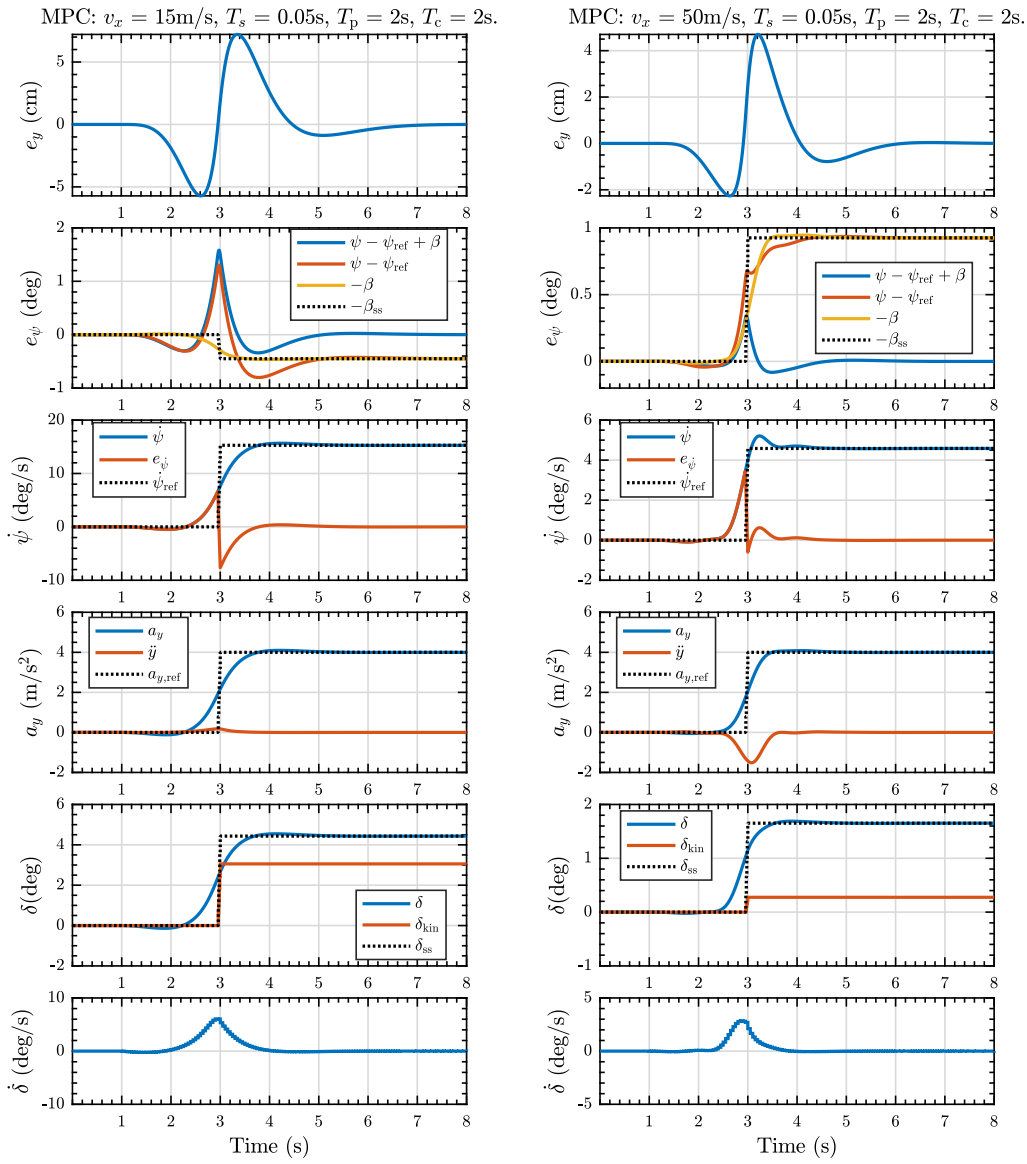
Prior to analyzing the response, let us recall the steady-state cornering values of the linear dynamic bicycle model from [27]. The steady-state steering angle required to track a circular trajectory of curvature κ_{ref} is given as

$$\delta_{\text{ss}}(v_x, \kappa_{\text{ref}}) = \left(L + K_V v_x^2\right) \kappa_{\text{ref}} \tag{7.15}$$

and consists of the kinematic steering angle $\delta_{\text{kin}} = L\kappa_{\text{ref}}$ increased by the lateral acceleration in proportion to the vehicle understeer gradient K_V. The steady state sideslip angle β_{ss} is

$$\beta_{\text{ss}}(v_x, \kappa_{\text{ref}}) = \left(l_\text{r} - \frac{l_\text{f}}{2C_{\alpha_\text{r}} L}\, m v_x^2\right) \kappa_{\text{ref}}, \tag{7.16}$$

and becomes negative for the second case with $v_x = 50\,\text{m/s}$.



(a) for $v_x = 15$ m/s (b) for $v_x = 50$ m/s

Figure 7.4: Response to step change in curvature at different velocities.

The controller was implemented at $f_\mathrm{s} = 20$ Hz and the continuous-time model was used for simulation of the response using a variable-step solver. The response simulated at $v_x = 15$ m/s is shown in Fig. 7.4a and at $v_x = 50$ m/s in Fig. 7.4b. In both cases, offset-free tracking of the reference curve in e_d is achieved, because the controlled system contains the model of the reference curve. Heading error with the sideslip angle β included reaches zero in steady state and is the correct indicator of the vehicle heading. The preview of the reference change is utilized to execute the transition from a straight segment into the turn smoothly, with lower tracking errors, lower peak values of lateral acceleration and yaw rate, and lower steering rate, compared to the reactive control in Fig. 7.3.

While δ_ss of Eq. (7.15) was derived in [27] to be used as a feedforward input in a steering controller, presented MPC formulation determines the steady-state steering angle required for tracking the reference curvature implicitly. Moreover, the controller accounts for the transient behavior and adjusts the control input to

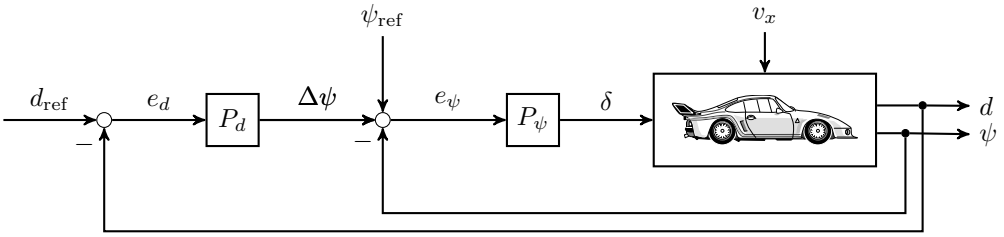


Figure 7.5: Reference architecture of a lateral motion controller based on a cascade structure and classical control design.

dampen the oscillations at higher speeds, improving the performance compared to a static feedforward gain.

7.3 Structure Comparison

In the recent related work [10, 11, 86, 17, 54, 43], both linear and nonlinear MPC was applied for the control of lateral vehicle motion, but, to our knowledge, the resulting controller structure was never compared with the previous works [5, 15] based on the classical control design methods. This section compares the controller structures using a unified notation, where P_x indicates a proportional gain from variable x and K_x indicates a proportional state feedback gain from x.

Fig. 7.5 shows the reference controller structure implemented for a scale RC car in [87]. For simplicity of the comparison, we consider only proportional controllers in place of the PIDs, as typically only P gains were tuned nonzero. Output saturation is neglected to compare the nominal linear control law. The steering control law is

$$\delta = P_\psi(\psi_{\text{ref}} - \psi + \Delta\psi) = P_\psi(\psi_{\text{ref}} - \psi) + P_\psi P_d(d_{\text{ref}} - d). \tag{7.17}$$

Considering Eq. (7.1), the structure is equivalent to PD control of e_d. When the controller is tracking a circular trajectory, the steady-state error e_d cannot reach zero as nonzero δ is required for tracking. No feedforward based on path curvature is utilized. High controller gains are required for low steady-state errors.

In Stanley [5], the winning car of the 2005 DARPA challenge, a nonlinear feedback law was used. Neglecting the δ saturation, Stanley's steering control law can be rewritten using the same notation as Eq. (7.17) into

$$\delta = P_\psi\Big(\psi_{\text{ref}} - \psi - \beta_{\text{ss}}(\kappa_{\text{ref}}, v_x)\Big) + \arctan\left(\frac{P_d}{1+v_x}\, e_d\right) + P_{\dot\psi}\Big(\dot\psi - \dot\psi_{\text{ss}}(\kappa_{\text{ref}}, v_x)\Big). \tag{7.18}$$

The first two terms were designed based on the kinematic model and later augmented by the yaw rate damping term, since yaw dynamics become more oscillatory at higher speeds as the damping due to the tire forces decreases, as we analyzed in Fig. 2.7b. The nonlinearity of arctan was used to smoothly reduce the steering command for larger crosstrack errors e_d. A gain scheduling of P_d was done dividing it by $(1 + v_x)$ to account for the LPV dynamics dependent on vehicle longitudinal velocity.

In 2012, Shelly, Stanford's Audi TTS [15], completed the first autonomous run up Pikes Peak in Colorado. The lateral motion controller for the task was designed using the classical control methods and consisted of three components

$$\delta = \underbrace{\delta_{\text{ss}}(\kappa_{\text{ref}}, v_x)}_{\delta_{\text{feedforward}}} + \underbrace{P_d\Big(e_d + x_{\text{la}} e_\psi\Big)}_{\delta_{\text{control}}} + \underbrace{P_{\dot\psi}\Big(\dot\psi - \dot\psi_{\text{ss}}(\kappa_{\text{ref}}, v_x)\Big)}_{\delta_{\text{damping}}}. \tag{7.19}$$

The feedforward part was based on track curvature and the dynamic bicycle model as expressed in Eq. (7.15). A lookahead distance x_{la} was used to project the crosstrack error into the curve-fixed reference frame, creating a prediction

$$\hat{e}_d = e_d + x_{\mathrm{la}} \sin(e_\psi) \approx e_d + x_{\mathrm{la}} e_\psi \tag{7.20}$$

of the crosstrack error e_d based on the heading error e_ψ. The result is equivalent as a proportional control of e_ψ with gain $P_{e_\psi} \leftarrow P_d x_{\mathrm{la}}$. Yaw damping was introduced to suppress yaw oscillations at higher speeds, as in the case of Stanley in Eq. (7.18).

Finally, let us analyze the state-feedback structure obtained using the presented lateral controller (LQR or MPC) in the unconstrained case and at a fixed speed. In contrast to the above approaches, steering rate $\dot{\delta}$ is controlled. The controller commands the steering rate, changing the steering angle until a steady state is reached. In the steady state with $\dot{\delta} = 0$, the feedback structure can be grouped into

$$\dot{\delta} = -K_d \underbrace{(d - d_{\mathrm{ref}})}_{e_d} - K_\psi \underbrace{(\psi - \psi_{\mathrm{ref}} + \beta)}_{e_\psi} - \dots \tag{7.21}$$

$$- K_{\dot{\psi}} (\dot{\psi} - \dot{\psi}_{\mathrm{ss}}(\kappa_{\mathrm{ref}}, v_x)) - K_\delta (\delta - \delta_{\mathrm{ss}}(\kappa_{\mathrm{ref}}, v_x)) - K_\beta (\beta - \beta_{\mathrm{ss}}(\kappa_{\mathrm{ref}}, v_x)). \tag{7.22}$$

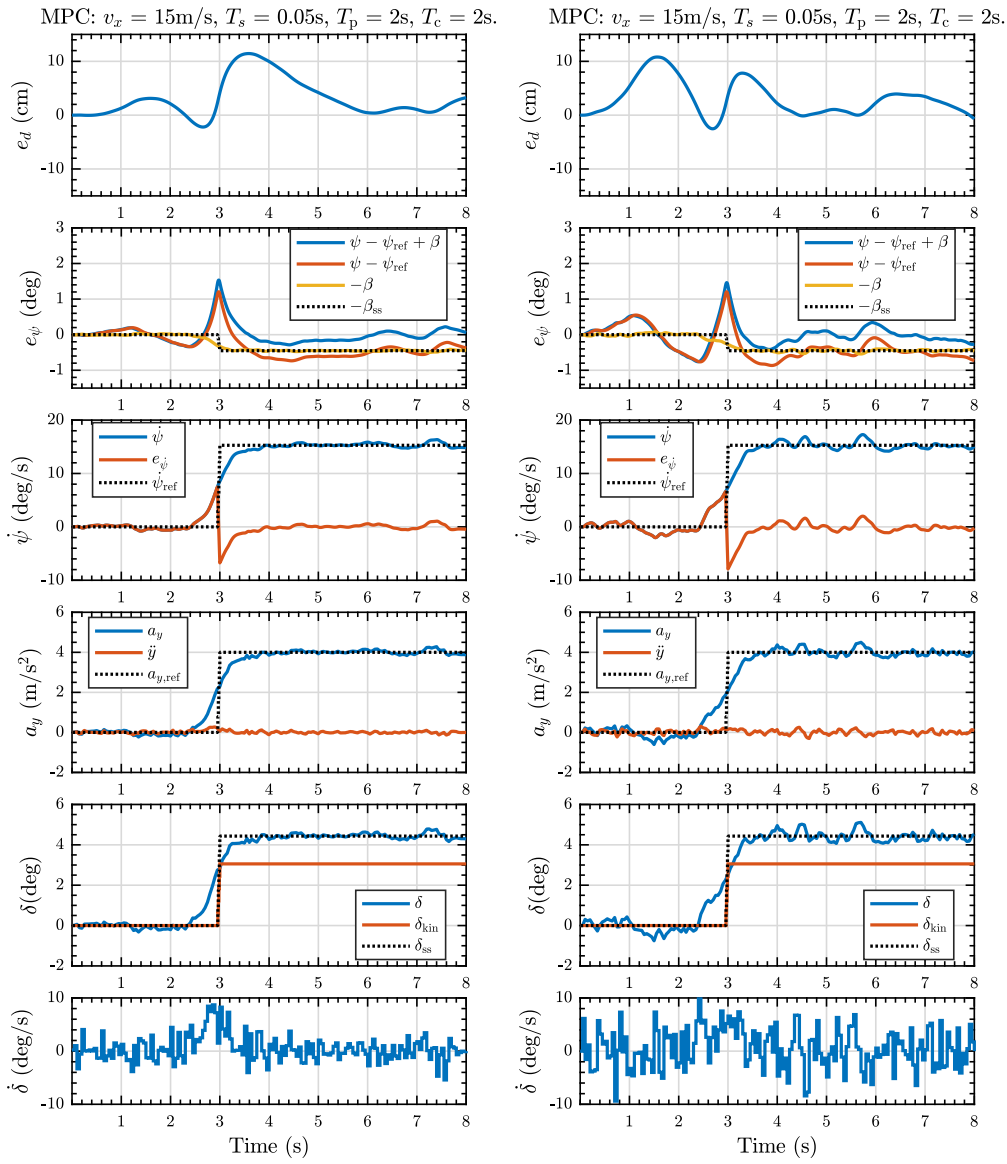
The controller contains all the typical components: a proportional feedback from the crosstrack error e_d, from the heading error e_ψ, a yaw rate damper, a δ feedforward based on Eq. (7.15), and an additional sideslip β compensation. Moreover, the MPC version features the preview-based feedforward $\mathbf{F}\bar{\mathbf{d}}$, creating an additional input to achieve optimal transient behavior when the reference is changing.

In contrast to controllers of Eq. (7.17) and Eq. (7.18), zero crosstrack error on circular trajectories is possible. Compared to Eq. (7.18), the state feedback law is designed all at once, instead of the sequential design of multiple SISO control laws. A similar approach as we present is adopted in the modern flight control systems [88], achieving typically a superior performance using the MIMO design compared to successive designs of dampers and autopilots as SISO loops.

Including reference curvature κ_{ref} as a state of the reference generator model is essential, allowing to model the steady-state cornering and to obtain the feedforward and compensation terms of Eq. (7.22) in the state feedback structure, thanks to the servomechanism problem formulation. MPC can be viewed as an online gain-scheduling scheme along an arbitrary velocity trajectory, taking into account the LPV dynamics, with the additional benefit of constraint handling. However, it can also be used for offline gain scheduling for relaxed constraints and v_x given. Controlling the steering rate $\dot{\delta}$ with constraints results in a smoother response of the steering angle δ. In Shelly [15], δ was the control input variable, causing non-smooth abrupt changes in presence of noise and disturbances.

7.4 Performance with Noise and Disturbance

To analyze the controller performance under noise, the typical accuracy of Oxford Technical Solution RT3002 [89] was used to create an estimator noise model. RT3002 is a high-grade sensing system which features dual-antenna GPS with an IMU to measure the position and the orientation of the vehicle in the inertial frame and the vehicle velocities in the vehicle body frame. RT3002 is used in vehicle dynamics testing and as a ground-truth reference for autonomous driving tests. In 2005, two of the five teams that successfully completed the DARPA Grand Challenge used



(a) $5 \times \sigma$ of RT3002 (b) $10 \times \sigma$ of RT3002

Figure 7.6: Performance of the lateral controller with simulated additive white Gaussian noise in the estimated state of the controlled system.

RT3002 for state estimation. The sensing system was also used for MPC-based autonomous steering in the experiments of [10, 86].

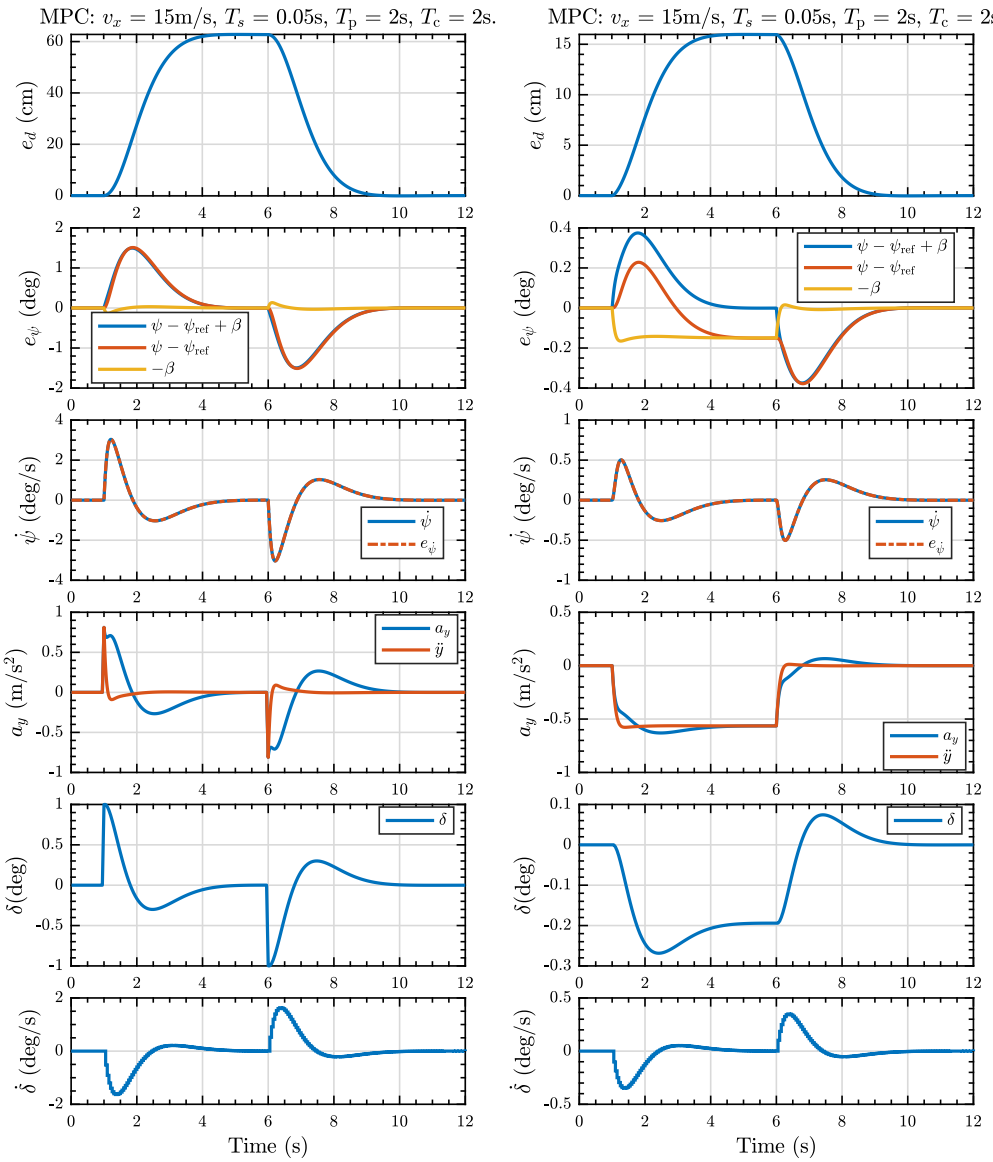
With differential corrections from a base station, the accuracy of RT3002 [89] is

$$\sigma_x = \sigma_y = 0.02\,\mathrm{m}, \quad \sigma_\psi = 0.1^\circ, \quad n_{v_x,\mathrm{RMS}} = 0.05\,\mathrm{km/h}, \quad \sigma_{\dot{y}} = 0.2\% v_x. \tag{7.23}$$

The noise model accuracy in acceleration and yaw rate measurements was based on the Bosch SMI710 inertial sensor for vehicle dynamics control [90]

$$n_{\dot{\psi},\mathrm{RMS}} = 0.15^\circ/\mathrm{s}, \qquad \sigma_{a_x} = \sigma_{a_y} = 0.06\,\mathrm{m/s^2}. \tag{7.24}$$

While a different noise models would be more appropriate in some of the state variables, such as a random walk in heading ψ, a zero-mean additive white Gaussian noise was used with standard deviations of Eq. (7.23) and Eq. (7.24), and zero



(a) unmeas. $+1°$ pulse in δ for $t \in \langle 1, 6 \rangle$ (b) side gust-wind, $v_\mathrm{wind} = 24$ m/s

Figure 7.7: Performance of the lateral controller under disturbances

cross-covariance. The purpose of the simple noise model is to validate the controller with degraded state information to avoid overtuning the controller with the perfect knowledge of all states.

Fig. 7.6 repeats the experiment of a step increase in curvature from Fig. 7.4a for two noise settings, five and ten times the nominal standard deviations of the noise model. The controller response is not tuned too aggressively, but with an increasing noise level, the steering rate and lateral jerk increases. The crosstrack error exhibits a random walk behavior and the precision of tracking with noise becomes limited, in Fig. 7.6b case to $|e_d| \leq 10$ cm with $\sigma_x = 20$ cm.

The response of the lateral motion controller to unmeasured disturbances was analyzed on a straight path segment using two experiments shown in Fig. 7.7. In both cases the open-loop response would result in an unbounded crosstrack error. The first case in Fig. 7.7a models a defect of the steering mechanism, a step increase of the steering angle by $1°$. The controller adjusts the steering angle back to zero

so that the crosstrack error remains bounded, but the crosstrack error is not fully suppressed. When the δ disturbance is removed, the controller readjusts the steering angle to zero and suppresses the crosstrack error completely.

The second case in Fig. 7.7b models a side-wind gust. The wind is modelled as a step side force calculated using the aerodynamic parameters of the vehicle model and corresponds to a side wind speed of $v_{\text{wind}} = 24\,\text{m/s}$. The vehicle is pushed sideways in the wind, causing a sideslip angle. The controller counter steers, stabilizing the crosstrack error.

In both cases, the disturbance is unknown to the controller. The output of the system is stabilized, but the persistent disturbance is not fully suppressed as the controller does not contain its model. To achieve a zero crosstrack error under step disturbance, the model would have to be augmented with an additional integrator, representing the step disturbance itself or the integral of crosstrack error.

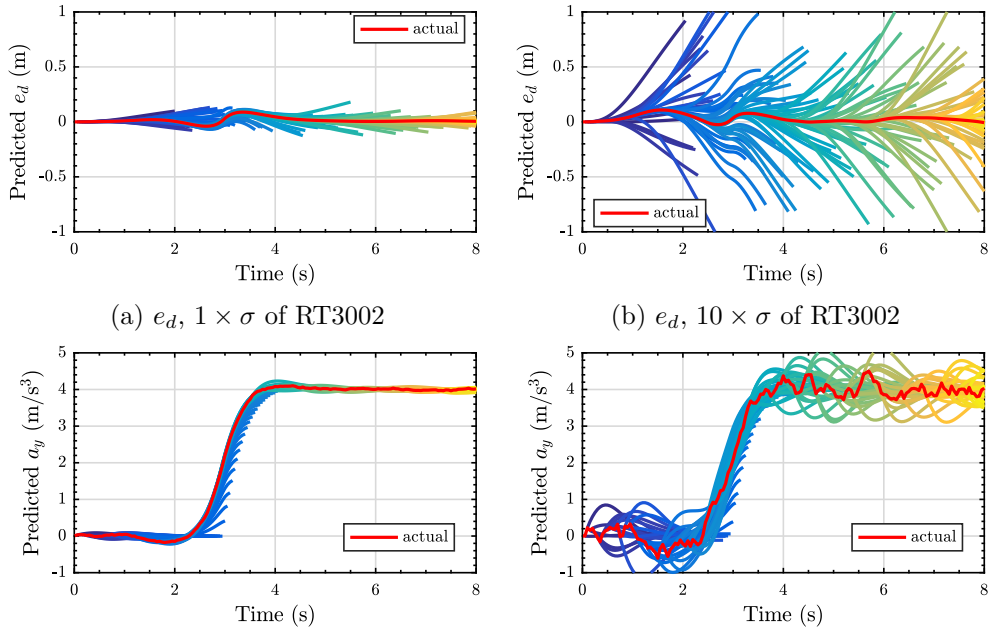
7.5 Lateral Controller Constraints

This section discusses the constraints imposed in the lateral motion controller. In Fig. 7.3 a large initial peak occurred in response to the initial conditions. Steering rate is directly related to the lateral jerk, which shall be limited for ride comfort and safety. Moreover, rate constraints ensure that the control action is feasible given the slew rate of the steering servomotor. Input box constraints are simple to include in both of the presented MPC problem formulations and most QP solvers support box constraints with no noticeable increase of the computational time.

State constraints in the primal dense formulation are polytopic in the control input and the increase in computational time with the active set solver qpOASES is noticeable, especially when the active set changes significantly. Moreover, infeasibility may result from too restrictive state constraints. Therefore, this section presents the possible uses of state constraint and discusses their applicability to achieve the desired behavior. The steps of implementing the constraints are formally identical as in Section 6.5, using outputs Eq. (7.11) and Eq. (7.12) with the model of Eq. (7.9).

Steering angle δ is mostly determined by curvature of the reference trajectory. While a small steering angle may cause a large lateral acceleration at high speeds, a larger steering angle is required to negotiate sharp turns at low speeds, making the steering angle bounds velocity dependent. Along the trajectory, only small corrections from the steady-state steering angle are required to compensate the system dynamics and disturbances. Violation of the physical range of the steering angle is unlikely as only small angles are typically commanded at high speeds.

The ultimate goal is to command the steering angle such that the tires do not lose adhesion, respectively the lateral acceleration limit must not be exceeded. Lateral acceleration constraints indirectly limit the steering angle amplitude and the bounds can be formulated independent of the velocity. Constraints of a_y cannot be formulated as hard, because the response of a_y cannot be always controlled into the feasible space in a single control step. Therefore, a_y constraints are softened over the first n_s steps of the prediction horizon, limiting the additional complexity and the number of slack variables. The purpose of a_y constraints is to prevent steering too sharply and sliding in response to disturbances. However, when the velocity trajectory is not planned consistently, a larger turning radius is executed, which may lead to exiting the track. Therefore, the bounds on a_y shall be higher in the lateral motion controller than in the trajectory generation to allow a safety margin. On the contrary, a_y predictions based on the linear model tend to overestimate the actual



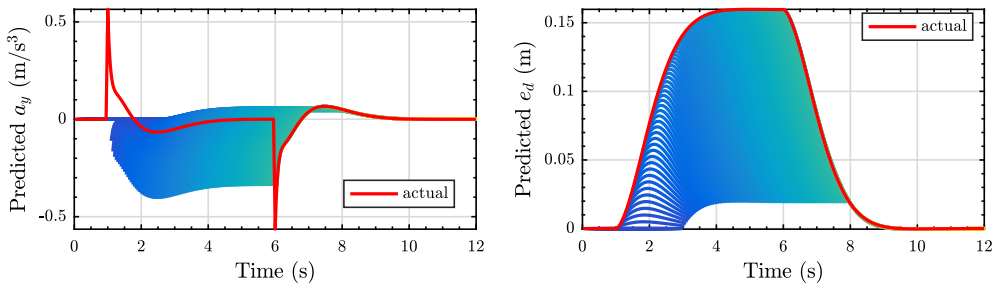
(a) e_d, $1 \times \sigma$ of RT3002

(b) e_d, $10 \times \sigma$ of RT3002

(c) a_y, $1 \times \sigma$ of RT3002

(d) a_y, $10 \times \sigma$ of RT3002

Figure 7.8: Impact of noise in the estimated state on the controller predictions during the step increase of reference curvature



(a) lateral acceleration a_y

(b) crosstrack error e_d

Figure 7.9: Impact of an unknown side force disturbance on the predictions

lateral acceleration, as was analyzed in Fig. 2.9a, the bounds may be too restrictive, and therefore, must be set experimentally.

The crosstrack error e_d can be constrained to prevent large deviations from the nominal path to stay within the road boundaries or to avoid obstacles as described in [17, 86]. Since e_d cannot be controlled instantaneously, its constraints must be softened to ensure problem feasibility. Increased state estimator noise causes an overestimated prediction of e_d as shown in Fig. 7.8b, making the predicted constraint violations too pessimistic. Comparing Fig. 7.8c and Fig. 7.8d, the predictions of a_y are not affected as much by the measurement noise. At the limits of vehicle handling, predicted e_d is inaccurate as the model uncertainty is integrated. The actual crosstrack error is higher than the prediction when sliding occurs. Experimental results at the handling limits in Chapter 8 are similar to those of an unknown side force disturbance in Fig. 7.9. Predicted e_d is below the actual and predicted a_y becomes offset from zero. Therefore, constraints of e_d were not implemented and only the criterion weight is used to keep e_d low.

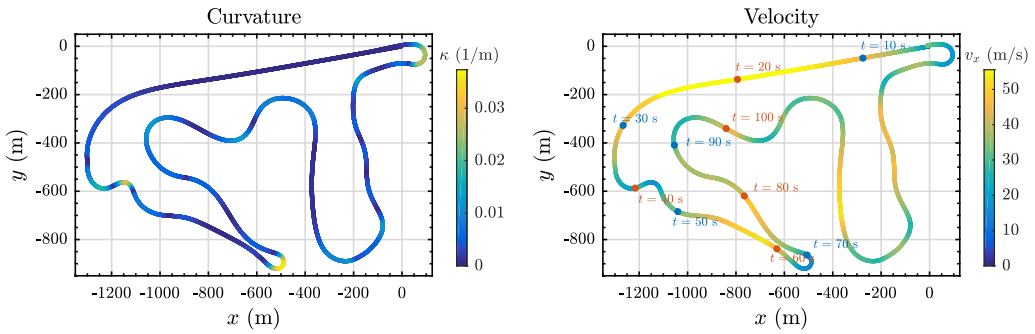
8 | Experiments

This chapter presents closed-loop test results of the lateral and longitudinal motion controllers implemented in Simulink and integrated with the vehicle model of IPG Carmaker. The test consists of simulated driving on the Nardo handling track.

The experiment was motivated by the possible application of the controllers in automated vehicle dynamics testing in collaboration with Porsche Engineering Services, Prague. Other applications of the experiment are to provide a vehicle motion control for the eForce Prague Formula [91] for possible participation in the emerging Formula Student Driverless competition [1], and for the downscaled F1/10 competition [2]. The goal of the experiment is to track the minimum-time trajectory, pushing the vehicle to the handling limits to determine controller limitations when the simplifying assumptions become violated. The experiment was simulated with four times the typical noise power of RT3002, described in Section 7.4, to benchmark the controller in a more realistic setup with degraded state information.

8.1 Trajectory Generation

The shape of the Nardo handling track is shown in Fig. 8.1a. The track contains several straight segments, a chicane, and two hairpin turns, indicated by high curvature. The simulated track is not entirely flat, its profile varies both in elevation and road bank angle. For the trajectory generation, the height profile was neglected. The input path was assumed fixed. The reference curve was reconstructed using spline interpolation of positions measured during simulated execution of the track by IPG Driver. The velocity trajectory was generated using steps described in Chapter 3 with the parameters of Appendix B. The track was traversed in the counter-clockwise direction, starting at the origin. The velocity map is shown in Fig. 8.1b, with a checkpoint indicated every 10 s for reference with the time responses.



(a) track curvature

(b) velocity trajectory

Figure 8.1: Nardo handling track: Reference curve reconstructed using spline interpolation and velocity trajectory mapped to the track segments.

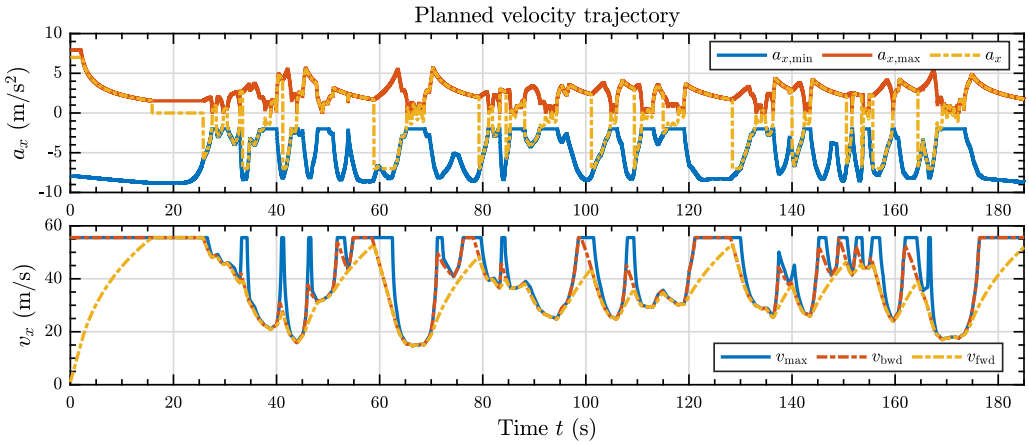


Figure 8.2: Planned acceleration and velocity trajectory

The reference values planned for the longitudinal controller are shown in Fig. 8.2. The upper plot depicts the desired longitudinal accelerations and the limits based on the vehicle powertrain and tire adhesion. The maximum available longitudinal acceleration is applied, limited by either the powertrain torque limit or lateral acceleration in turns. For simplicity, the friction coefficient μ was assumed constant along the track. The resulting velocity trajectory is shown in the lower plot of Fig. 8.2, together with the braking and the maximum speed profile.

8.2 Longitudinal Tracking

This section documents the performance of the longitudinal velocity controller commanding the desired acceleration of the full vehicle model. While the vehicle completed successfully multiple laps, with the first lap time of 178 s, only the first 100 s is depicted for clarity. A video of the full experiment is posted online [92, 93], several pictures of the experiment are attached in Fig. A.1 of Appendix A.

The uppermost plot of Fig. 8.3 shows the longitudinal acceleration commanded to the underlying powertrain controller. The command was constrained by the limits precalculated together with the trajectory in Fig. 8.2. The second plot of Fig. 8.3 captures the velocity tracking performance achieved on the time-varying reference, with the speed difference magnified in the third plot. The desired speed was tracked closely while satisfying the acceleration limits, using limited longitudinal jerk and accounting for the response time of the lower-level powertrain controller. The velocity controller achieved a robust performance in the presence of measurements noise (in v_x and a_x) and disturbances, such as road bumps.

A large error occurred at time $t = 60\,\text{s}$, due to a sudden change of the road grade, resulting in a jump over a track horizon, with all four wheels leaving the road surface temporarily. The relative elevation of the road is shown in Fig. 8.4a. Just before $t = 60\,\text{s}$, a negative error developed as the vehicle was at the limit of commanded acceleration while going uphill. The jump occurred at $t = 60\,\text{s}$, a positive error developed because the vehicle was unable to brake in mid-air and the braking capability was reduced as the vehicle suspension oscillated after landing. The height of the vehicle's center of mass above the road surface is shown in Fig. 8.4b. Nonetheless, the controller stabilized the vehicle and recovered successfully after the

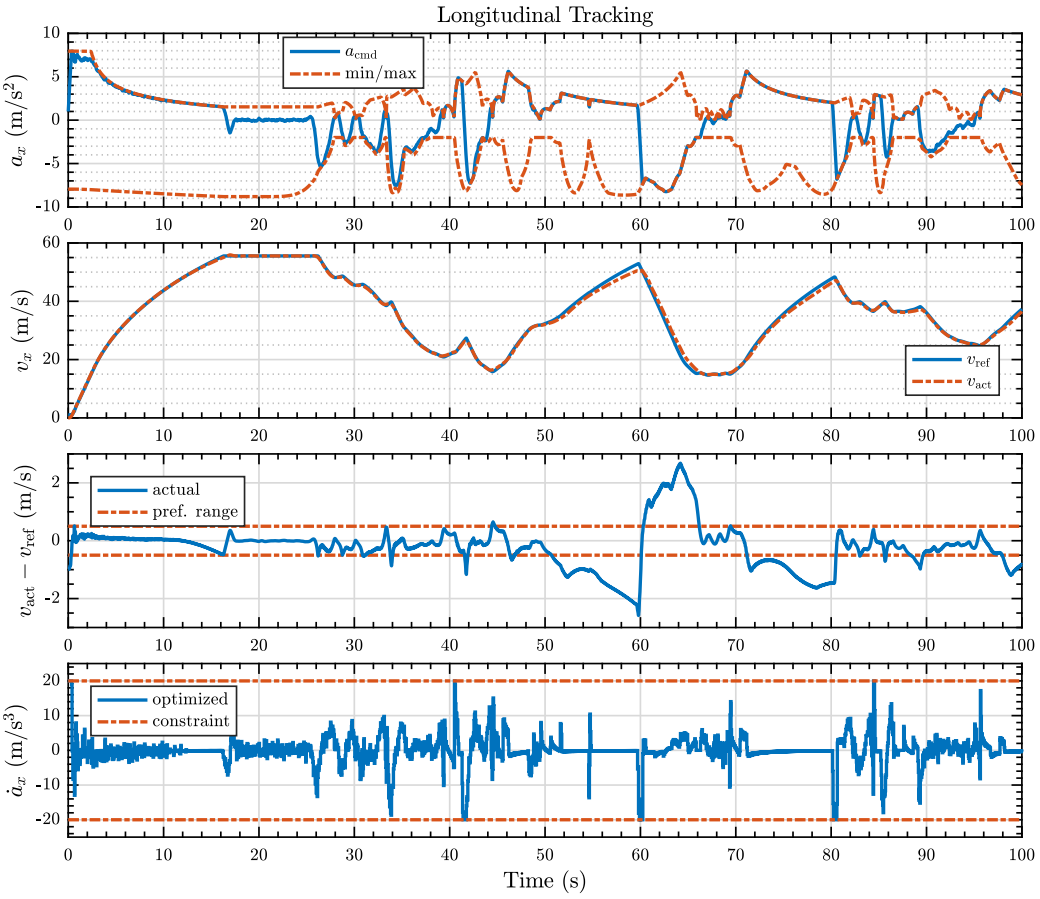
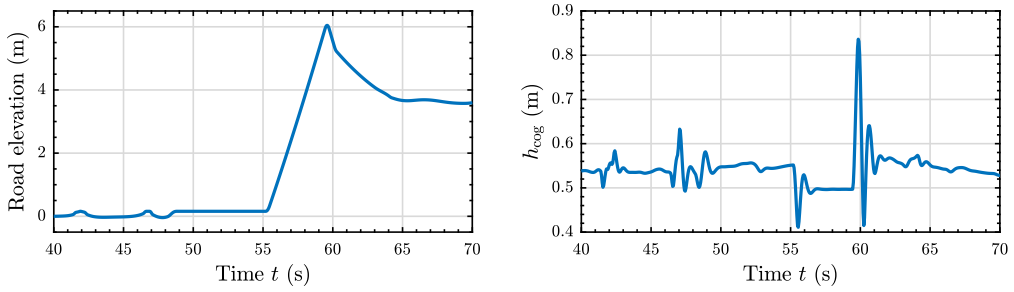


Figure 8.3: Longitudinal velocity tracking performance during simulated driving on the Nardo handling track



(a) relative elevation of the road

(b) height above the road surface

Figure 8.4: A sudden change in the road grade causing a jump

disturbance. In the flat segments of the track, close tracking of the velocity was achieved, with the error within ± 0.5 m/s.

8.3 Lateral Tracking

This section analyzes the lateral controller performance on the selected part of the Nardo handling track. The controller was formulated using the linearized dynamic bicycle model and used from the standstill, with the parameter $v_x \geq 1$ m/s lower-bounded to avoid the model singularity. The results are shown in Fig. 8.5. Along

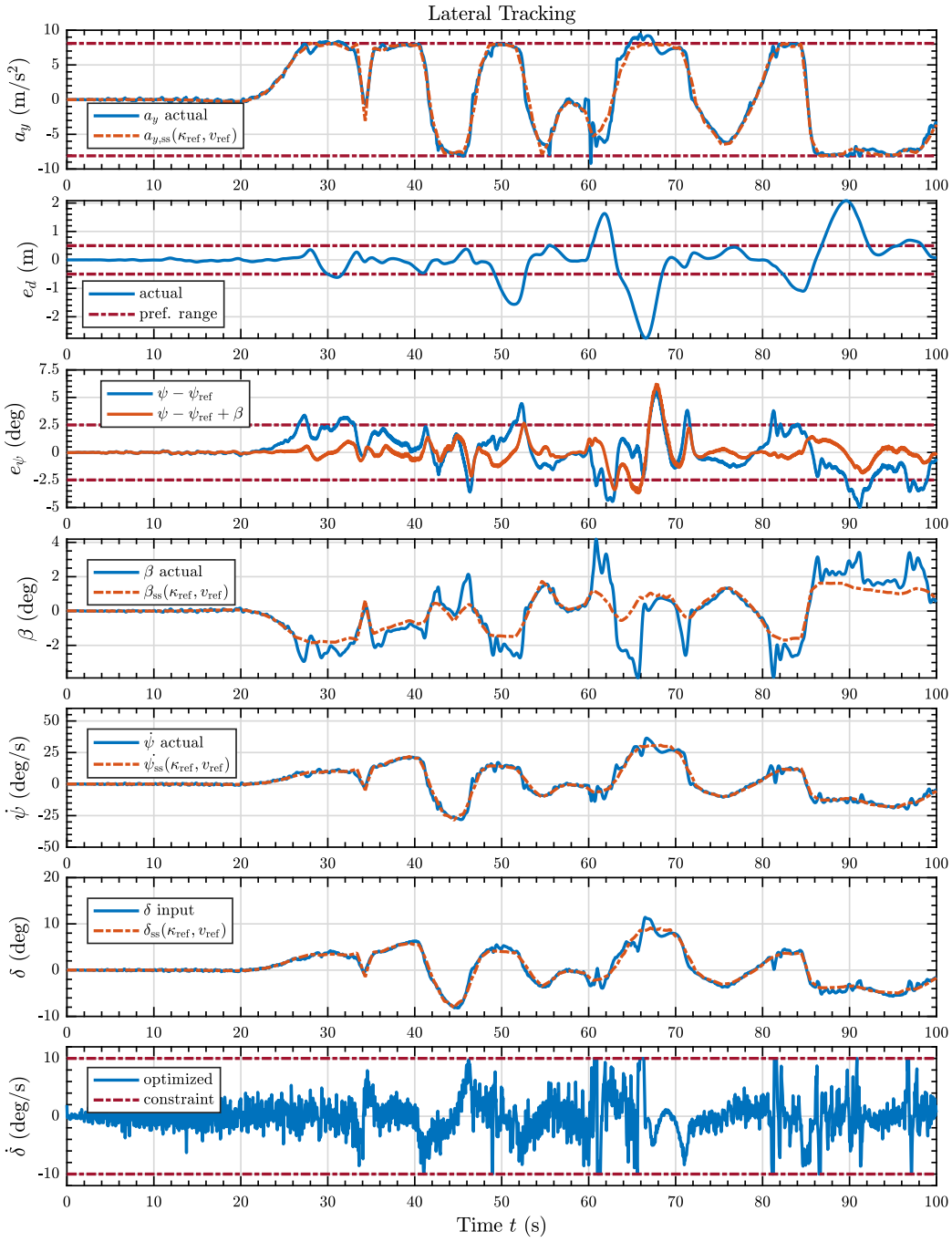
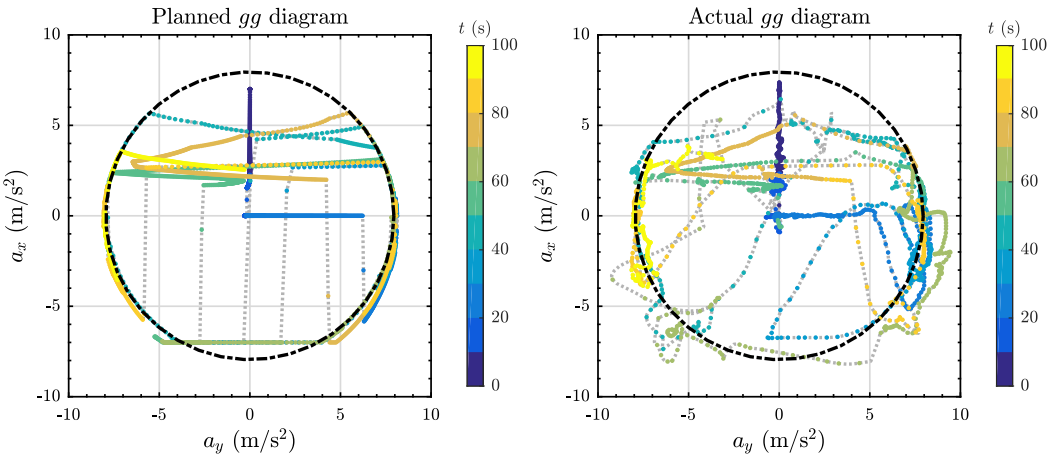


Figure 8.5: Performance of the lateral motion controller in simulated driving on the Nardo handling track

the first straight segment, a small chatter in steering occurred due to measurement noise in the crosstrack and heading errors. The first three turns (peaks in a_y) were executed at the planned level of lateral acceleration, with the crosstrack error below 50 cm. The largest error occurred after the jump at $t = 60$ s, the planned speed was exceeded and the lateral acceleration constraint prevented the controller from increasing the steering angle. Possible sliding from a more aggressive response was prevented, at the expense of a larger crosstrack error. At high levels of lateral acceleration $|a_y| \geq 5\,\mathrm{m/s^2}$, the vehicle exhibited a larger side slip angle β than modeled using the linear dynamic bicycle model, as predicted by the results of Fig. 2.9



(a) planned (b) actual

Figure 8.6: Performance of the controller expressed using the *gg* diagram.

in Chapter 2. The controller attempted to prevent a further increase of β by gentle counter-steering and compensated the β response based on the linear model. The controller achieved a tracking performance with $|e_\psi| \leq 2.5°$ and $|e_d| \leq 0.5\,\mathrm{m}$ during $t \in \langle 0, 50 \rangle$ or $t \in \langle 70, 80 \rangle$ when the motion was not disturbed significantly due to the road unevenness, and when longitudinal velocity was not exceeded.

8.4 Total Acceleration Analysis

This section analyzes the utilization of the vehicle handling limits in terms of *gg* diagrams, described in Section 3.1, indicating the total acceleration. Fig. 8.6a shows the desired *gg* diagram resulting from the trajectory generation. The interior of the circle contains fewer points as instantaneous transitions between braking and acceleration were assumed. The upper part of the diagram boundary contains no points because of the powertrain limitation. The highest positive longitudinal acceleration a_x is obtained at the start, where the peak motor torque is available. The points extend slightly outside the dashed boundary for negative a_x as the tire adhesion is increased by aerodynamic downforce at high speeds when braking is started. The planned diagram is limited to $|a_x| \leq 7\,\mathrm{m/s^2}$, to provide a safety margin for emergency braking during control.

The ability of the controller in terms of handling the vehicle at the limits is shown in Fig. 8.6b. The diagram interior contains more points due to the limited transition rates. Most of the points are scattered around the boundary as both controllers compensated model uncertainty and disturbances. The largest transgression of the boundary occurred after the jump at $t = 60\,\mathrm{s}$, dotted in light green. To reduce the speed, $a_x \leq 7\,\mathrm{m/s^2}$ was required, utilizing the safety margin. In the emergency situation at $t = 60\,\mathrm{s}$, the boundary had to be exceeded. The controllers stabilized the system, but the response could have been improved if motion coupling outside the approximately linear range of tire operation was modeled. A controller based on the nonlinear single-track model with combined tire force generation would perform better in control of emergency maneuvers in the frictional region of tire operation, such as during sliding or on road surfaces with low friction coefficient. Total acceleration time responses are compared in Fig. 8.7, indicating that the vehicle is kept successfully at the acceleration limits most of the time during the experiment.

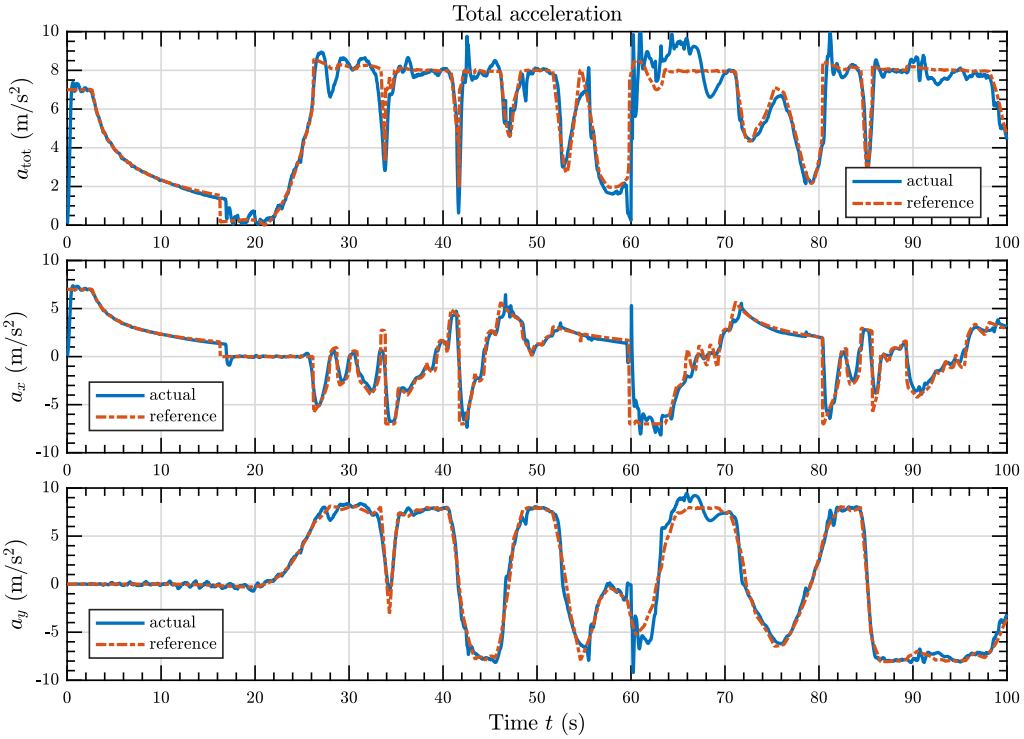


Figure 8.7: Utilization of the vehicle's total acceleration envelope

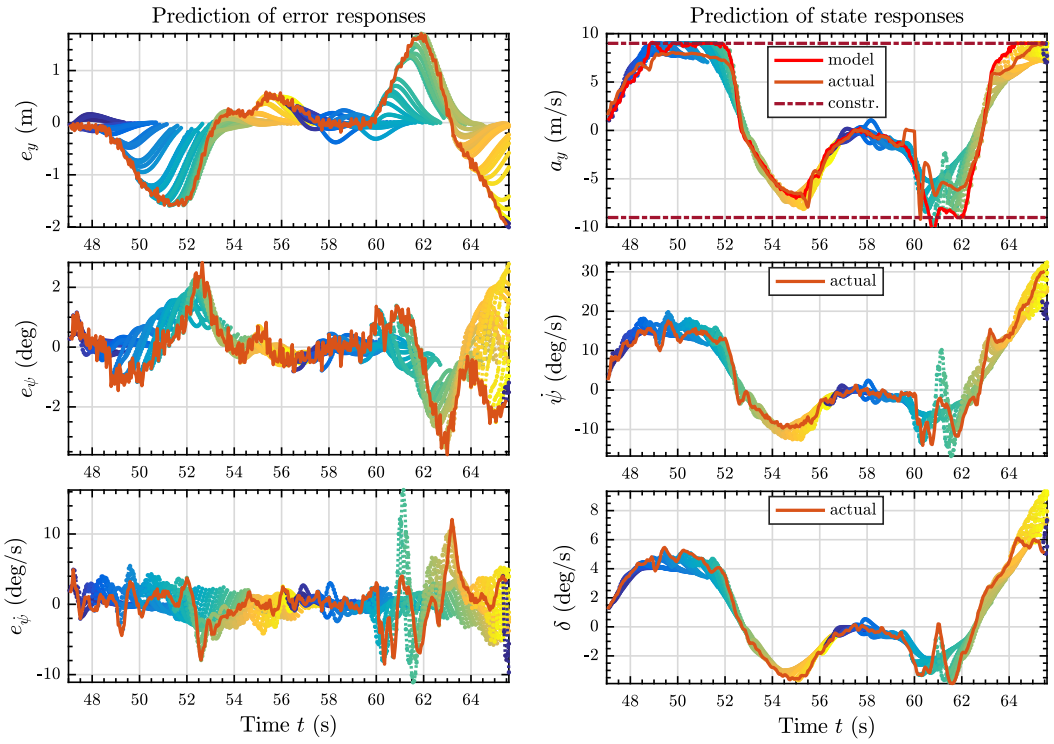
8.5 Predictions and Constraint Satisfaction

This section summarizes the accuracy of state response predictions and the use of state constraints during the validation test with the high-fidelity vehicle model.

The longitudinal controller was implemented with hard constraints of the longitudinal acceleration command. Combining a sufficient length of the prediction horizon and sufficiently large bounds on the longitudinal jerk (optimized variable), infeasibility did not occur in the simulated experiments, and the controller was always able to adjust its command ahead of the constraint. Using the constraint helped to satisfy the total acceleration limit, the boundary of the *gg* diagram, and the assumption of decoupling from the lateral controller. When implementing the constrained controller on a real vehicle, the constraint shall be softened or at least a backup routine has to be included to guarantee the problem infeasibility.

Secondly, an upper velocity limit was implemented as a soft constraint. However, the velocity constraint was not found to be practical. The upper bound was set to $e_{v_x} \leq 0.5\,\text{m/s}$ to keep some tolerance around the reference. The constraint comes into play only occasionally, resulting in short peaks of the commanded jerk in response to disturbances. In case of a large disturbance, the constraint must be temporarily relaxed. Without disturbances, the reference velocity is followed closely in predictions and the constraint is unnecessary. A similar performance is achieved by using a larger velocity error penalization q alone. Contrary to the longitudinal acceleration constraint, constraining the velocity error did not bring any benefits and increased the controller complexity. Thus, it may be removed in the future.

Lateral state predictions are shown in Fig. 8.8. The error accumulates by integration and is most noticeable in e_d of Fig. 8.8a. The predictions underestimate the future error, similar as in the case of an unmeasured side force disturbance in Fig. 7.9b. Soft constraints of e_d were used in [17] to offset the vehicle from



(a) error outputs

(b) steering and lateral acceleration

Figure 8.8: Model-based predictions used in the lateral controller

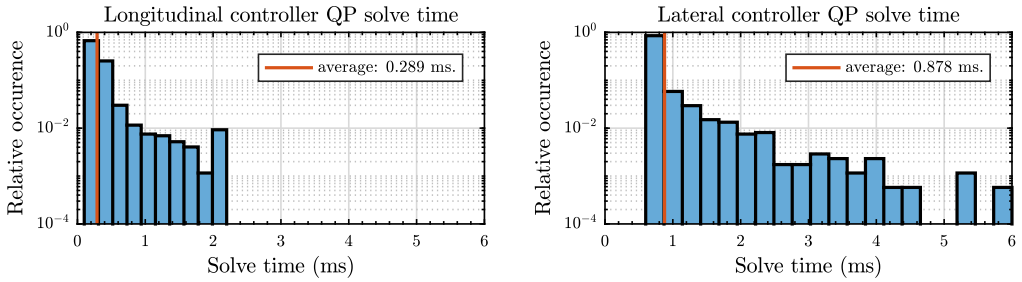
the reference to avoid an obstacle. However, penalization of e_d is sufficient for the track driving application as the predictions underestimate the actual error under the model mismatch. Constraining the error would increase the controller complexity and softening would be necessary. The constraints would not prevent the error as the error is the result of model uncertainty and not of the controller deliberate action.

Predictions of the controller's intended action, the steering angle and yaw rate responses in Fig. 8.8b, are more accurate than the predictions of the control action outcome in Fig. 8.8a. Soft constraints of the lateral acceleration are shown in Fig. 8.8b, preventing the vehicle from steering too sharply and possibly from sliding. The lateral acceleration constraints must be implemented as soft to avoid infeasibility under disturbances. The peak predictions based on the linear model overestimate the actual values, as was analyzed in Fig. 2.9a. Therefore, the upper bound was set experimentally to $9\,\mathrm{m/s^2}$, strictly higher then the limit used in the trajectory generation for consistency.

In conclusion, both of the acceleration constraints help to keep the vehicle at the handling limits and to satisfy the assumptions used for decoupling of the controllers. On the contrary, constraints of the longitudinal velocity and crosstrack error increase the controller complexity and no benefits were found for the presented application over using the error penalization alone.

8.6 Solve Time

Solve times and especially the guaranteed worst-case runtime are an important aspect for the real-time application of the controller. The active-set solver qpOASES



(a) longitudinal controller

(b) lateral controller

Figure 8.9: Time required to solve the quadratic program to obtain the control input

exhibited fast solve times when the working set of constraints was not changing significantly. When some of the constraints had become active, the number of working set recalculations increased. A limit on the number of iterations is used to fulfill the real-time requirements. If exceeded, a backup procedure shall be used. The procedure can be a simpler constant-gain controller or the MPC problems with some of the constraints relaxed. However, no more than 40 iterations out of the limit of 100 was required in our experiments. The times required to solve the QP on a desktop PC with i5-6200 CPU running at 2400 MHz are shown in Fig. 8.9, with an average of 878 µs for the lateral controller with time-varying matrices and problem size $n_z = 60$, and 289 µs for the longitudinal controller with fixed matrices and problem size $n_z = 40$. The solve times alone fall well within the sampling time of 50 ms.

However, a significant amount of time was spent formulating the time-varying longitudinal acceleration bounds (0.4 ms) and reformulating the dense LPV MPC problem (5 ms) at each iteration, taking roughly 25% of the total computational time and limiting the simulation execution to a real-time factor of 2.4. Both routines were implemented as a Level-2 MATLAB S-Function and further optimizations are possible by reimplementing the routines in C/C++. Another option to speed-up the LPV problem formulation is to use a QP solver optimized for the sparse formulation, such as HPMPC, to avoid the multiplications associated with the formulation of the prediction matrices and the condensation of the Hessian.

While the move-blocking of Section 5.6 was implemented in the controllers and tested, we did not invest the time into the heuristic tuning of the control horizon as the possible reduction of the QP solve times was negligible compared to the time required for the LPV MPC problem formulation. Therefore, no move-blocking was utilized in the final experiments and optimizing the problem formulation is the primary focus for possible computational performance improvement.

9 | Conclusions

The three main goals of the thesis were to develop a vehicle dynamics model, to design a trajectory tracking controller and to verify the closed-loop tracking performance. All the three goals were fulfilled and the achieved results uncovered possible future research directions for further improvements.

9.1 Achieved Results

In Chapter 2, we developed models for description of vehicle motion dynamics and performed experiments for model identification and validation using standardized maneuvers and a vehicle dynamics simulator. Variation of lateral dynamics with speed was analyzed and pitfalls of model discretization were highlighted. Mismatch of the linear model under high lateral acceleration was analyzed, indicating the limited range of linear operation.

Chapter 3 presented a trajectory generation algorithm, based on representing the reference path using a spline curve and applying a forward-backward iterative integration scheme for minimum-time trajectory generation. Other implementation aspects, such as an approximate calculation of path tracking errors and a look-up scheme to preview the trajectory, were covered in preparation of the closed-loop control experiment.

In Chapter 4, we prepared a state-space formulation of the tracking problem using the servomechanism problem structure and in Chapter 5 we formulated the model-predictive control problem. Selected formulation allowed to analyze the feedback structure resulting from the unconstrained MPC, making the results of optimization-based control more accessible to practitioners.

The two major contributions of the thesis are found in the chapters on controller design. In Chapter 6, the longitudinal velocity controller was designed and used to illustrate the benefits of predictive control in contrast to reactive state feedback. In Chapter 7, the lateral motion control was formulated as a model-matching problem in the servomechanism structure. The state-space model of [14] was formalized and extended for the dynamic bicycle model. Feedback structure resulting from an unconstrained MPC controller was compared with the previous works based on classical control theory [5, 15], creating an important link with the recent work [14]. Controller robustness in the presence of noise and disturbances was analyzed. In both chapters, the use of constraints was discussed to enable control of the vehicle at the limits of handling using linear models and decoupled controllers.

Chapter 8 is the culmination of the development, integrating the trajectory generation, longitudinal and lateral controllers, and the vehicle dynamics simulator in a closed-loop experiment. Simulated driving on a handling track was demonstrated, with an acceleration envelope of $8\,\mathrm{m/s^2}$, close to the handling limits of the vehicle model used in the simulation.

9.2 Future Work

9.2.1 Nonlinear motion control

The linear dynamic bicycle model becomes inaccurate for manuevers with lateral acceleration $|a_y| \geq 4\,\mathrm{m/s^2}$ due to the nonlinearity of the cornering force characteristic. Moreover, the coupling of the cornering and tractive tire force in not covered in the linear models. For improved tracking performance at $|a_y| \geq 4\,\mathrm{m/s^2}$ and systematic handling of the nonlinear operation at $|a_y| \geq 8\,\mathrm{m/s^2}$, controller based on a model with nonlinear combined tire force generation shall be used. Further resources about advanced tire models are found in [26]. In electric vehicles with individual-wheel drive, the tracking problem may include traction control and torque vectoring to achieve even higher performance. Twin-track models for the task are developed in [20, 19]. However, model identification will become more challenging and the control design would call for the use of nonlinear methods, such as Nonlinear MPC.

9.2.2 Hardware Implementation

Another step is the implementation of the controllers on a hardware platform. On top of porting the controller code, implementing a vehicle state estimator and repeating model identification is necessary. The timing performance of the controller may be improved by using a solver such as HPMPC, optimized for embedded platforms and the sparse formulation of the MPC problem.

9.2.3 Vehicle State Estimation

As stated in the introductory outline, vehicle state estimation was not covered in the thesis scope, although full state information is necessary for hardware implementation. While commercial off-the-shelf solutions for vehicle state estimation exist, such as the RT3002 [89], these may be unavailable in prototype projects with a restricted budget or for use in series production.

The conditions imposed by racing call for the use of a nonlinear state estimator. Preliminary experiments were carried out during the thesis. An unscented Kalman filter was implemented using the nonlinear bicycle model of Eq. (2.5) with an option to select either the linear or Fiala tire model. As measurements, a single-antenna GPS and an IMU were modeled. The nonlinear tire model improved estimator performance, especially in terms of sideslip and lateral acceleration. However, the solution was incomplete for use in the closed-loop experiment, in particular the initialization of the estimator during drive-off from stand-still.

An adequate treatment of the topic was outside the available time scope. Design of a nonlinear state estimator for racing applications is a prospective thesis topic. The estimator shall include a nonlinear tire model and the utilized sensory information shall be extended from the GPS/IMU combination to include wheel odometry and possibly visual sensors. A systematic treatment of sensor faults, such as wheel spin, GPS dropout, and GPS multipath error detection, is necessary.

9.2.4 Automated Path Planning

Another area for improvement is the integration of the controllers with an automated path planner, either with the presented trajectory generation algorithm or with simultaneous optimization of both the path and the velocity profile.

Bibliography

[1] The Tech That Won the First Formula Student Driverless Race. https://spectrum.ieee.org/cars-that-think/transportation/self-driving/the-tech-that-won-the-first-formula-student-driverless-race. Accessed: 2018-05-15.

[2] F1/10 Autonomous Racing Competition. http://f1tenth.org/competition. Accessed: 2018-05-17.

[3] Roborace, Press Releases. https://roborace.com/media/. Accessed: 2018-05-09.

[4] Roborace, Autonomous Racing Competition. https://www.audi.com/en/innovation/piloteddriving/roborace.html. Accessed: 2018-05-10.

[5] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun. Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing. In *2007 American Control Conference*, pages 2296–2301, July 2007.

[6] Ch. Urmson, J. Anhalt, D Bartz, et al. A Robust Approach to High-Speed Navigation for Unrehearsed Desert Terrain. *Journal of Field Robotics*, 2006.

[7] R. Craig Coulter. Implementation of the Pure Pursuit Path Tracking Algorithm. Technical report, Carnegie Mellon University, Robotics Institute, 1992.

[8] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat. MPC-based approach to active steering for autonomous vehicle systems. *International Journal of Vehicle Autonomous Systems*, 3(2-4):265–291, 2005.

[9] F. Borrelli, A. Bemporad, M. Fodor, and D. Hrovat. An MPC/hybrid system approach to traction control. *IEEE Transactions on Control Systems Technology*, 14(3):541–552, May 2006.

[10] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat. Predictive Active Steering Control for Autonomous Vehicle Systems. *IEEE Transactions on Control Systems Technology*, 15(3):566–580, May 2007.

[11] P. Falcone, H. E. Tseng, F. Borrelli, J. Asgari, and D. Hrovat. MPC-based yaw and lateral stabilisation via active front steering and braking. *Vehicle System Dynamics*, 46(sup1):611–628, 2008.

[12] A. Gray, M. Ali, Y. Gao, J. K. Hedrick, and F. Borrelli. A Unified Approach to Threat Assessment and Control for Automotive Active Safety. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1490–1499, Sept 2013.

[13] P. Waldmann and D. Niehues. Der BMW Track-Trainer, Automatisiertes fahren im Grenzbereich auf der Nürburgring Nordschleife. *4. Tagung Sicherheit durch Fahrerassistenz*, April 4. Tagung Sicherheit durch Fahrerassistenz.

[14] M. Aeberhard, S. Rauch, M. Bahram, G. Tanzmeister, J. Thomas, Y. Pilat, F. Homm, W. Huber, and N. Kaempchen. Experience, Results and Lessons Learned from Automated Driving on Germany's Highways. *IEEE Intelligent Transportation Systems Magazine*, 7(1):42–57, Spring 2015.

[15] K. Kritayakirana. *Autonomous Vehicle Control at the Limits of Handling*. PhD thesis, Stanford University, June 2012.

[16] P. F. Lima. *Predictive Control for Autonomous Driving*. PhD thesis, KTH Royal Institute of Technology, Stockholm, May 2016.

[17] B. Gutjahr, L. Gröll, and M. Werling. Lateral Vehicle Trajectory Optimization Using Constrained Linear Time-Varying MPC. *IEEE Transactions on Intelligent Transportation Systems*, 18(6):1586–1595, June 2017.

[18] D. M. Bevly, J. Ryu, and J. C. Gerdes. Integrating INS Sensors With GPS Measurements for Continuous Estimation of Vehicle Sideslip, Roll, and Tire Cornering Stiffness. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):483–493, Dec 2006.

[19] M. Guiggiani. *The Science of Vehicle Dynamics: Handling, Braking, and Ride of Road and Race Cars*. Springer, Netherlands, 2014.

[20] D. Schramm, M. Hiller, and R. Bardini. *Vehicle Dynamics: Modeling and Simulation*. Springer, UK, 2014.

[21] Vehicle Dynamics Terminology. Recommended practice, Society of Automotive Engineers, Inc, Warrendale, USA, 1976.

[22] Road vehicles – Three-dimensional reference system and fiducial marks – Definitions . Standard, International Organization for Standardization, Geneva, CH, 1978.

[23] Road vehicles – Vehicle dynamics and road-holding ability – Vocabulary. Standard, International Organization for Standardization, Geneva, CH, 2011.

[24] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer, 2009.

[25] B. Siciliano and O. Khabit, editors. *Springer Handbook of Robotics*. Springer, 2008.

[26] H.B. Pacejka and I. Besselink. *Tire and Vehicle Dynamics*. Butterworth-Heinemann. Butterworth-Heinemann, 2012.

[27] R. Rajamani. *Vehicle Dynamics and Control*. Mechanical Engineering Series. Springer US, 2012.

[28] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli. Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1094–1099, June 2015.

[29] U. Kiencke and L. Nielsen. *Automotive Control Systems: For Engine, Driveline and Vehicle*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 2000.

[30] W.F. Milliken and D.L. Milliken. *Race Car Vehicle Dynamics*. Premiere Series. SAE International, 1995.

[31] S.K. Clark and United States. National Bureau of Standards. Office of Vehicle Systems Research. *Mechanics of Pneumatic Tires*. Number 122 in Mechanics of Pneumatic Tires. U.S. Government Printing Office, 1971.

[32] E. Fiala. Seitenkrafte am rollenden luftreifen. *VDI-Zeitschrift*, 1954.

[33] H. B. Pacejka and E. Bakker. The Magic Formula Tyre Model. *Vehicle System Dynamics*, 21(sup001):1–18, 1992.

[34] R.Y. Hindiyeh and J. Gerdes. Equilibrium Analysis of Drifting Vehicles for Control Design. In *ASME 2009 Dynamic Systems and Control Conference*, volume 1, pages 181–188, 2009.

[35] Road vehicles – Lateral transient response test methods – Open-loop test methods. Standard, International Organization for Standardization, Geneva, CH, 2011.

[36] Mondek M. Active Torque Vectoring Systems for Electric Drive Vehicles. Master's thesis, Czech Technical University in Prague, 2018.

[37] Passenger cars – Test track for a severe lane-change manoeuvre – Part 1: Double lane-change. Standard, International Organization for Standardization, Geneva, CH, 1999.

[38] S. Li, K. Li, R. Rajamani, and J. Wang. Model Predictive Multi-Objective Vehicular Adaptive Cruise Control. *IEEE Transactions on Control Systems Technology*, 19(3):556–566, May 2011.

[39] V. L. Bageshwar, W. L. Garrard, and R. Rajamani. Model predictive control of transitional maneuvers for adaptive cruise control vehicles. *IEEE Transactions on Vehicular Technology*, 53(5):1573–1585, 2004.

[40] A. Rucco, G. Notarstefano, and J. Hauser. Computing minimum lap-time trajectories for a single-track car with load transfer. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 6321–6326, Dec 2012.

[41] Giacomo Perantoni and David J.N. Limebeer. Optimal control for a Formula One car with variable parameters. *Vehicle System Dynamics*, 52(5):653–678, 2014.

[42] R. Lot and F. Biral. A Curvilinear Abscissa Approach for the Lap Time Optimization of Racing Vehicles. In *19th IFAC World Congress*, volume 47, pages 7559 – 7565, 2014.

[43] A. Liniger, A. Domahidi, and M. Morari. Optimization-based Autonomous Racing of 1:43 Scale RC Cars. *Optimal Control Applications and Methods*, 36(5):628–647, November 2017.

[44] Thomas Lipp and Stephen Boyd. Minimum-time speed optimisation over a fixed path. *International Journal of Control*, 87(6):1297–1311, 2014.

[45] J. Subosits and J. C. Gerdes. Autonomous vehicle control for emergency maneuvers: The effect of topography. In *2015 American Control Conference (ACC)*, pages 1405–1410, July 2015.

[46] N.R. Kapania, J. Subosits, and Ch. J. Gerdes. A Sequential Two-Step Algorithm for Fast Generation of Vehicle Racing Trajectories. *ASME Journal of Dynamic Systems, Measurement, and Control*, 2016.

[47] D. González, J. Pérez, V. Milanés, and F. Nashashibi. A Review of Motion Planning Techniques for Automated Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1135–1145, April 2016.

[48] B. A. Francis and W. M. Wonham. The Internal Model Principle of Control Theory. *Automatica 12*, pages 457–465, 1976.

[49] V. Havlena and J. Štěcha. *Moderní teorie řízení*. Ediční středisko ČVUT, Zikova 4, Praha 6, 1999.

[50] V. Kučera. The discrete Riccati equation of optimal control. *Kybernetika*, 1972.

[51] F.L. Lewis, D. L. Vrabie, and V.L. Syrmos. *Optimal Control*. A Wiley-interscience publication. John Wiley & sons, Inc., 3rd edition, 2012.

[52] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733 – 764, 2003.

[53] David Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967 – 2986, 2014.

[54] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes. Collision Avoidance and Stabilization for Autonomous Vehicles in Emergency Scenarios. *IEEE Transactions on Control Systems Technology*, 25(4):1204–1216, July 2017.

[55] P. F. Lima, M. Trincavelli, J. Mårtensson, and B. Wahlberg. Clothoid-based model predictive control for autonomous driving. In *2015 European Control Conference (ECC)*, pages 2983–2990, July 2015.

[56] J. M. Maciejowski. *Predictive Control with Constraints*. Pearson Education. Prentice Hall, 2002.

[57] O. Mikuláš. Quadratic Programming Algorithms for Fast Model-Based Predictive Control. Bachelor's thesis, Czech Technical University in Prague, 2013.

[58] P. Otta. Numerical Algorithms for Quadratic Programming for Approximated Predictive Control. Master's thesis, Czech Technical University in Prague, 2013.

[59] O. Šantin. *Numerical Algorithms of Quadratic Programming for Model Predictive Control*. PhD thesis, Czech Technical University in Prague, August 2016.

[60] J. A. Rossiter. *Model-Based Predictive Control: A Practical Approach*. Control Series. CRC Press, 2003.

[61] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[62] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Verlag, 1999.

[63] M. Diehl and S. Gros. *Numerical Optimal Control*. Unpublished manuscript, 2017. Online: `http://www.syscop.de/numericaloptimalcontrol`.

[64] H. J. Ferreau, S. Almér, R. Verschueren, M. Diehl, D. Frick, A. Domahidi, J. L. Jerez, G. Stathopoulos, and C. Jones. Embedded Optimization Methods for Industrial Automatic Control. In *IFAC World Congress*, volume 50, pages 13194 – 13209, Toulouse, France, July 2017.

[65] D. Kouzoupis, A. Zanelli, H. Peyrl, and J. Ferreau. Towards Proper Assessment of QP Algorithms for Embedded Model Predictive Control. In *European Control Conference*, Linz, Austria, July 2015.

[66] C. N. Jones, A. Domahidi, M. Morari, S. Richter, F. Ullmann, and M. Zeilinger. Fast Predictive Control: Real-time Computation and Certification. In *4th IFAC Conference on Nonlinear Model Predictive Control*, volume 45, pages 94 – 98, 2012.

[67] GNU Lesser General Public License (LGPL) version 2.1, February 1999. Online: `https://www.gnu.org/licenses/license-list.html#LGPL`.

[68] F. Ullmann. FiOrdOs: A matlab toolbox for c-code generation for first order methods, 2011. Available at: `hhttp://fiordos.ethz.ch/dokuwiki/doku.php`.

[69] J. Mattingley, Y. Wang, and S. Boyd. Receding Horizon Control: Automatic Generation of High-speed Solvers. *IEEE Control Systems*, 31(3):52–65, 2011.

[70] J. Mattingley and S. Boyd. CVXGEN: A Code Generator for Embedded Convex Optimization. In *Optimization and Engineering*, November 2012.

[71] A. Domahidi. Forces pro: Fast optimization for real-time control on embedded systems, October 2012. Online: `https://www.embotech.com/FORCES-Pro`.

[72] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones. Efficient interior point methods for multistage problems arising in receding horizon control. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 668–674, Dec 2012.

[73] AMZ Driverless Racing Powered by embotech Motion Planning Technology. `https://www.embotech.com/Automotive`. Accessed: 2018-03-22.

[74] M. Valls, H. Hendrikx, V. Reijgwart, F. V. Meier, I. Sa, R. Dubé, A. R. Gawel, M. Bürki, and R. Siegwart. Design of an Autonomous Racecar: Perception, State Estimation and System Integration. In *International Conference on Robotics and Automation*, Brisbane, May 2018.

[75] J. V. Frasch, S. Sager, and M. Diehl. A Parallel Quadratic Programming Method for Dynamic Optimization Problems. *Mathematical Programming Computation*, 7:289–329, September 2015.

[76] J. V. Frasch. qpDUNES, a dual newton strategy for convex quadratic programming, 2013. Available at: `http://mathopt.de/qpDUNES`.

[77] H.J. Ferreau, C. Kirches, A. Potschka, H.G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014.

[78] H.J. Ferreau, H.G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

[79] G. Frison, H. Sørensen, B. Dammann, and J. B. Jørgensen. High-Performance Small-Scale Solvers for Linear Model Predictive Control. In *13th European Control Conference*, Strasbourg, France, 2014.

[80] G. Frison and J. B. Jørgensen. Efficient implementation of the Riccati recursion for solving linear-quadratic control problems. In *2013 IEEE International Conference on Control Applications (CCA)*, pages 1117–1122, August 2013.

[81] M. Diehl. Embedded optimization for model predictive control of mechatronic systems. Slides online: `https://bit.ly/2H6zROv`.

[82] R. Cagienard, P. Grieder, E. C. Kerrigan, and M. Morari. Move blocking strategies in receding horizon control. *Journal of Process Control*, 17(6):563–570, July 2007.

[83] R. Gondhalekar and J. Imura. Least-restrictive move-blocking model predictive control. *Automatica*, 46(7):1234 – 1240, 2010.

[84] M. Brunner, U. Rosolia, J. Gonzales, and F. Borrelli. Repetitive learning model predictive control: An autonomous racing example. In *IEEE Conference on Decision and Control*, pages 2545–2550, 12 2017.

[85] A. Micaelli and C. Samson. *Trajectory tracking for unicycle-type and two-steering-wheels mobile robots*. PhD thesis, INRIA, 1993.

[86] Y. Gao. *Model Predictive Control for Autonomous and Semiautonomous Vehicles*. PhD thesis, University of California, Berkeley, 2014.

[87] M. Vajnar. Model car for the F1/10 Autonomous Car Racing Competition. Master's thesis, Czech Technical University in Prague, May 2017.

[88] M.B. Tischler. *Advances In Aircraft Flight Control*. Series in Systems and Control. Taylor & Francis, 1996.

[89] Oxford Technical Solutions Ltd, United Kingdom. *RTv2 GNSS-aided Inertial Measurement Systems, User Manual*, 2015. Rev. 180221.

[90] Robert Bosch GmbH, Stuttgart, Germany. *Active safety systems: SMI710 combined inertial sensor for vehicle dynamics control*, 2015. Rev. 180221.

[91] eForce FEE Prague Formula, Czech Technical University in Prague. `https://eforce.cvut.cz/`. Accessed: 2018-05-15.

[92] Trajectory Tracking for Autonomous Vehicles, Nardo Handling Track, Video. `https://youtu.be/H2dodv4G6qs`. Accessed: 2018-05-15.

[93] Trajectory Tracking for Autonomous Racing, Lateral Motion, Video. `https://youtu.be/jcbOGr7LM2I`. Accessed: 2018-05-25.

Appendix A

Vehicle Dynamics Simulators

A part of the assignment was to perform a survey of vehicle dynamics simulators. Notes on the simulator selection are summarized in this section to keep modeling in Chapter 2 compact and applicable in general. The simulators for autonomous driving can be split into two main categories.

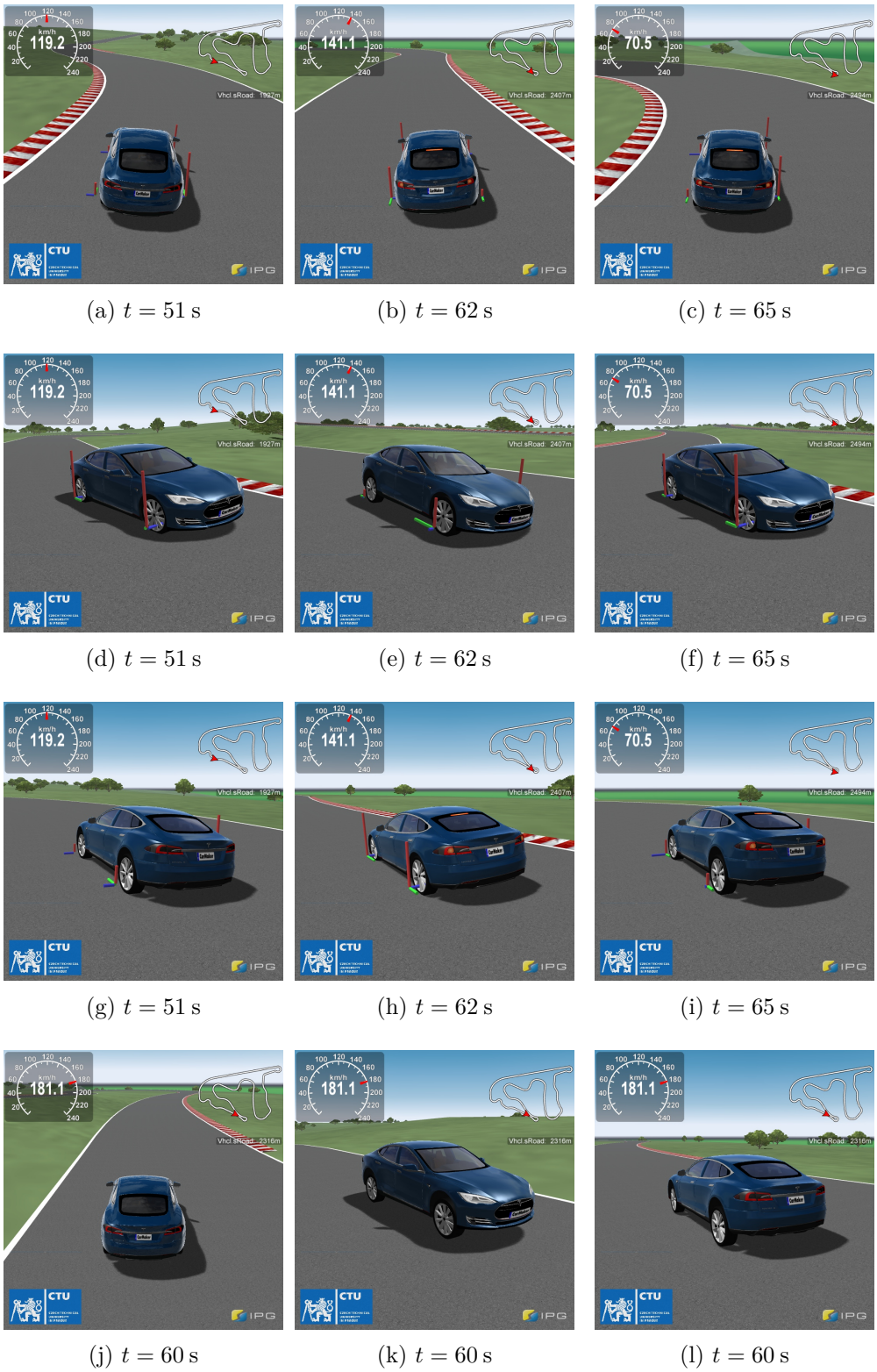
The first category focuses on vehicle dynamics modeling and vehicle handling. With the development of advanced driver assistance system, sensors models are included, but the focus of the simulation remains on the dynamic response of the vehicle, respectively the control action and its results, not the sensing. A wide range of commercial products exists in the first category, some of the most frequently used are CarSim, dSPACE ASM, Tesis Dynaware veDYNA, Elektrobit ADTF, Cruden Panthera, Vires VDT, and IPG Carmaker

The second category emerged recently with the development of deep neural networks, focusing mainly on generating realistic exteroceptive sensory data (such as from lidars, depth and color cameras) for training of classifiers for applications such as pedestrian, road marking, or road signs detection or deep end-to-end learning of automated driving. Examples of the second category are Deepdrive, Microsoft AirSim, and recently even GTA V. Environments adopted from mobile robotics, such as ROS and Gazebo, may be included under this category.

The scope of this work is suited for the first category, to test the model-based controllers with a high-fidelity model of vehicle dynamics, including tires, suspension, aerodynamics, powertrain and other vehicle components. Out of the available vehicle dynamics simulators, we selected IPG Carmaker, as it is widely used[1] in the automotive industry by brands such as Audi, BMW, Ferrari, Porsche, Rimac or Volkswagen. Using a standardized simulator enables the vehicle manufacturers to take over the controllers developed in this work, reparametrize them accordingly and simulate their operation with proprietary validated high-fidelity models. An academic license was granted for the duration of the thesis by IPG Automotive.

We used IPG Carmaker with the Simulink interface, which allowed faster validation and prototyping of the control algorithms. Although every model is only an approximation, using the vehicle dynamics simulator offered a better insight in the dynamics phenomena which are otherwise difficult or expensive measure (tire forces, wheel slip angles, vehicle slip angle). Using model-in-the-loop validation allowed to formulate repeatable tests in a controlled environment, without the need of preparing the vehicle platform and securing the proving ground, and without the risk of property damage or injuries. A use case of the simulator is depicted in Fig. A.1.

[1] Companies using IPG Carmaker: https://ipg-automotive.com/company/references/



(a) $t = 51$ s (b) $t = 62$ s (c) $t = 65$ s

(d) $t = 51$ s (e) $t = 62$ s (f) $t = 65$ s

(g) $t = 51$ s (h) $t = 62$ s (i) $t = 65$ s

(j) $t = 60$ s (k) $t = 60$ s (l) $t = 60$ s

Figure A.1: IPG Carmaker, the vehicle dynamics simulator used for model-in-the-loop validation of the designed controllers. Pictures of the experiment described in Chapter 8, simulated driving on the Nardo handling track with Tesla Model S vehicle model supplied with the software.

Appendix B

Parameters

This section summarizes the parametrization of the vehicle model and algorithms used throughout the thesis. The reference vehicle model is Tesla Model S, as provided in IPG Carmaker for reproducibility of the experiments.

Table B.1: Vehicle model parameters

Description	**Notation**	**Value**
Vehicle mass	m	2108 kg
Vehicle moment of inertia with respect to its vertical principal axis	I_z	3960.8 kgm^2
Distance of the front axle from the table of mass along the longitudinal axis	l_f	1.516 m
Distance of the rear axle from the table of mass along the longitudinal axis	l_r	1.484 m
Front axle effective cornering stiffness	C_{α_f}	98 kN/rad
Rear axle effective cornering stiffness	C_{α_r}	230 kN/rad

Table B.2: Lateral controller parameters

Description	**Notation**	**Value**
Sampling time	T_s	50 ms
Prediction time	T_p	3 s
Preview time	T_w	2 s
Prediction horizon length	N_p	60
Number of control moves	N_c	60
Steering rate box constraints	$\lvert\dot{\delta}_\mathrm{cmd}\rvert$	10°/s
Weight of the crosstrack error squared e_d^2	$\mathbf{Q}_\mathrm{e}(1,1)$	0.025
Weight of the heading error squared e_ψ^2	$\mathbf{Q}_\mathrm{e}(2,2)$	2.5
Weight of the yaw rate error squared $e_{\dot{\psi}}^2$	$\mathbf{Q}_\mathrm{e}(3,3)$	0.4
Weight of the accel. error squared $e_{a_y}^2$	$\mathbf{Q}_\mathrm{e}(4,4)$	0.001
Weight of the steering rate squared	$\mathbf{R}$	1
Maximum lateral acceleration	$a_{y,\max}$	9.0 m/s^2

Table B.3: Trajectory generation parameters

Description	**Notation**	**Value**
Gravitational acceleration	g	$9.8137\,\mathrm{m/s^2}$
Air density	ρ	$1.225\,\mathrm{kg/m^3}$
Friction coefficient	μ	0.80
Maximum speed	v_{lim}	$200\,\mathrm{km/h}$
Sampling step in distance	Δs	$1\,\mathrm{m}$
Tire rolling radius	r	$0.346\,\mathrm{m}$
Transmission ratio	k_{trans}	9.73
Transmission efficiency	k_{eff}	1
Maximum torque	τ_{max}	$600\,\mathrm{Nm}$
Maximum power	R_{max}	$250\,\mathrm{kW}$
Maximum speed	ω_{max}	$16,000$ RPM
Maximum longitudinal acceleration	$a_{\mathrm{max,ub}}$	$7\,\mathrm{m/s^2}$
Minimum longitudinal acceleration	$a_{\mathrm{min,lb}}$	$-7\,\mathrm{m/s^2}$
Vehicle frontal reference area	A	$2.408\,\mathrm{m^2}$
Vehicle drag coefficient	c_{D}	0.280
Vehicle downforce coefficient	c_{L}	0.149
Vehicle rolling resistance coefficient	b	0

Table B.4: Longitudinal controller parameters

Description	**Notation**	**Value**
Sampling time	T_{s}	$50\,\mathrm{ms}$
Prediction time	T_{p}	$2\,\mathrm{s}$
Preview time	T_{w}	$2\,\mathrm{s}$
Prediction horizon length	N_{p}	40
Number of control moves	N_{c}	40
Weight of the speed error squared	q	1000
Weight of the longitudinal jerk squared	r	1
Longitudinal jerk box constraints	$\lvert\dot{a}_{\mathrm{cmd}}\rvert$	$20\,\mathrm{m/s^3}$

Table B.5: State estimator noise model parameters

Description	**Notation**	**Value**
Position	σ_x, σ_y	$4\,\mathrm{cm}$
Heading	σ_ψ	$0.2°$
Velocity	σ_{v_x}, σ_{v_y}	$0.1\,\mathrm{km/h}$
Yaw rate	$\sigma_{\dot{\psi}}$	$0.15°/\mathrm{s}$
Acceleration	σ_{a_x}	$0.1\,\mathrm{m/s^2}$

Appendix C

Content of the attached CD

File	Content description
`dipl_filipj12.pdf`	Text of the thesis in `.pdf` format.
`video_nooverlay.avi`	Full video of the simulated experiment from Ch. 8, no overlay.
`video_lat_trk.avi`	Video of the first 100 s of the experiment from Ch. 8 with an overlay plot of lateral tracking errors.