Technical University of Munich
TUM School of Engineering and Design

Robust and data-driven control for autonomous racing

Alexander Wischnewski

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

genehmigten Dissertation.

Vorsitz:	Prof. Dr. phil. Klaus Bengler
Prüfer*innen der Dissertation:	1. Prof. Dr.-Ing. habil. Boris Lohmann
	2. Prof. Dr.-Ing. Ulrich Konigorski
	3. Priv.-Doz. Dr.-Ing. habil. Marion Leibold

Die Dissertation wurde am 22.12.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 02.05.2023 angenommen.

Contents

Acknowledgments

This dissertation was written during my work as a research assistant at the *Chair of Automatic Control* at the *Technical University of Munich* in cooperation with the *Institute of Automotive Technology* from December 2017 to February 2022.

I would like to thank my supervisor, Prof. Dr.-Ing. Boris Lohmann, for the outstanding supervision, the challenging discussions on a variety of topics, ranging from fundamentals of control theory to its practical application in engineering as well as the chance to work within an excellent research group. In addition, I would like to thank Prof. Dr.-Ing. Ulrich Konigorski, for serving as a second supervisor, Priv.-Doz. Dr.-Ing Marion Leibold, for serving as a third supervisor, and Prof. Dr. phil. Klaus Bengler for serving as a chairman of the examination committee.

This thesis and the various experiments within realistic real-world conditions have been made possible via the founding of the TUM Autonomous Motorsport team by Prof. Dr-Ing. Markus Lienkamp and Dr.-Ing. Johannes Betz. You made it possible to grew this team from initially a handful of people up to a large interdisciplinary group of great researchers and automotive enthusiasts. I will be forever grateful for the trust and chance to lead this group starting from Summer 2020 up to winning the first Indy Autonomous Challenge event in October 2021. Even though its a long list, I am going to name everyone at this point to highlight their outstanding achievements: Tobias Betz, Felix Fent, Maximilian Geisslinger, Alexander Heilmeier, Leonhard Hermansdorfer, Thomas Herrmann, Sebastian Huch, Phillip Karle, Felix Nobis, Levent Ögretmen, Matthias Rowold, Florian Sauerbeck, Tim Stahl, Rainer Trauth, and Frederik Werner. We couldn't have done it with a single person missing!

All of the experiments done within this thesis have been conducted on two different autonomous racing vehicles: The first was provided by the team at Roborace, the second was provided by the team from the Indy Autonomous Challenge. Special thanks go to Ollie Walsh, Tim Phillips, Dan Willians, Danny King, David Sale, Timo Völkl, and the many others on the Roborace team which allowed us to achieve so much via the continuous support. The same has to be said to Lauren and Craig Brooks, Matt Peak, Kent Anderson, Kris Kozak, Paul Mitchell and the many others on the Indy Autonomous Challenge team. It has been an amazing experience to work with all of you.

I want to thank the team at the Chair of Automatic Control for all the fruitful discussions on control engineering as well as the great atmosphere within the team. This has been complemented by all the students who contributed with their research to the TUM Autonomous Motorsport project and I was happy to work with all of you. In addition, I want to thank my mentor Thomas Specker, who has supervised my master thesis which was my entrypoint into the field of motion control for autonomous vehicles at the handling limits and has been a constant source of valuable feedback on my work.

Finally, I want to thank the team which has shared my path since the very beginning: my parents, my sister, my grandparents, and my friends. You all supported and encouraged me in pursing this large venture and have therefore been an invaluable part to finish it.

Freising, December 2022

Alexander Wischnewski

Abstract

Highly automated vehicle motion control is a well-matured part of driver assistance systems. The progress towards autonomous driving and the rise of autonomous racing competitions inspired researchers to investigate motion control algorithms at the handling limit. The nonlinear and complex dynamics in this operating region are well-suited for model predictive control algorithms, however, their performance depends strongly on the model accuracy and reliability of numerical optimization algorithms.

This work investigates how these shortcomings could be overcome with the application of robust and data-driven control techniques. Within the first stream of work, it introduces a Tube-MPC concept based on a friction-limited point-mass model, fast low-level control loops for the lateral and longitudinal acceleration, and a cost tuning allowing the controller to re-optimize coarse target trajectories for minimum jerk within an admissible driving corridor. This concept is combined with a state estimator based on a similar point-mass model. Both do not utilize information about the tire behavior, which makes their predictive performance independent from detailed nonlinear models in the tire saturation region. The algorithms were applied during the Indy Autonomous Challenge, where they reached $265\,\mathrm{km\,h^{-1}}$ and accelerations up to $21\,\mathrm{m\,s^{-2}}$ in a two-vehicle race at the Las Vegas Motor Speedway. In the second stream of work, a safe learning algorithm is introduced to optimize trajectory planning parameters for minimum lap-time while monitoring certain safety metrics such as the lateral control error. Finally, a recursive uncertainty estimation algorithm consisting of a least-squares estimator and a quantile estimator which is capable of improving the feedforward control law of the acceleration controllers and adjusting the uncertainty assumptions of the Tube-MPC simultaneously is presented. This algorithm is then integrated with the presented Tube-MPC and demonstrates to reduce the uncertainty within the control system and therefore decrease the required safety margins.

The results show that the explicit consideration of uncertainties within the control design can lead to more cautious behavior when needed while allowing aggressive driving behavior when possible, which leads to safer driving behavior overall. In addition, data-driven strategies can improve the overall performance, however, it has to be kept in mind that they will never perfectly match the model to the real-world behavior. This indicates that its a promising direction to merge data-driven strategies with thorough handling of the remaining uncertainties within a robust control framework.

Acronyms

ACC Adaptive Cruise Control

EKF Extended Kalman Filter

GNSS Global Navigation Satellite System

GPR Gaussian Process Regression

IAC Indy Autonomous Challenge

ILC Iterative Learning Control

IMS Indianapolis Motor Speedway

IMU Inertial Measurement Unit

LKA Lane Keeping Assist

LVMS Las Vegas Motor Speedway

MPC Model Predictive Control

NLP nonlinear program

ODD Operational Design Domain

PTP Precision Time Protocol

QP quadratic program

SAE Society of Automotive Engineers

SQP sequential quadratic program

1 Introduction

1.1 Motivation

In recent years, the automotive industry has been subject to two significant transformations: First, the shift from internal combustion engines to electric powertrains is already underway for in-production vehicles. Their market share is increasing, and optimistic scenarios predict electric vehicle sales to rise to more than two-thirds of the market by 2030 within Europe [19]. Second, the transition from human-driven vehicles to autonomously operated vehicles is a focus of research and early-stage product development teams worldwide. In contrast to electric mobility, this area is less mature and has many challenges that have yet to be solved.

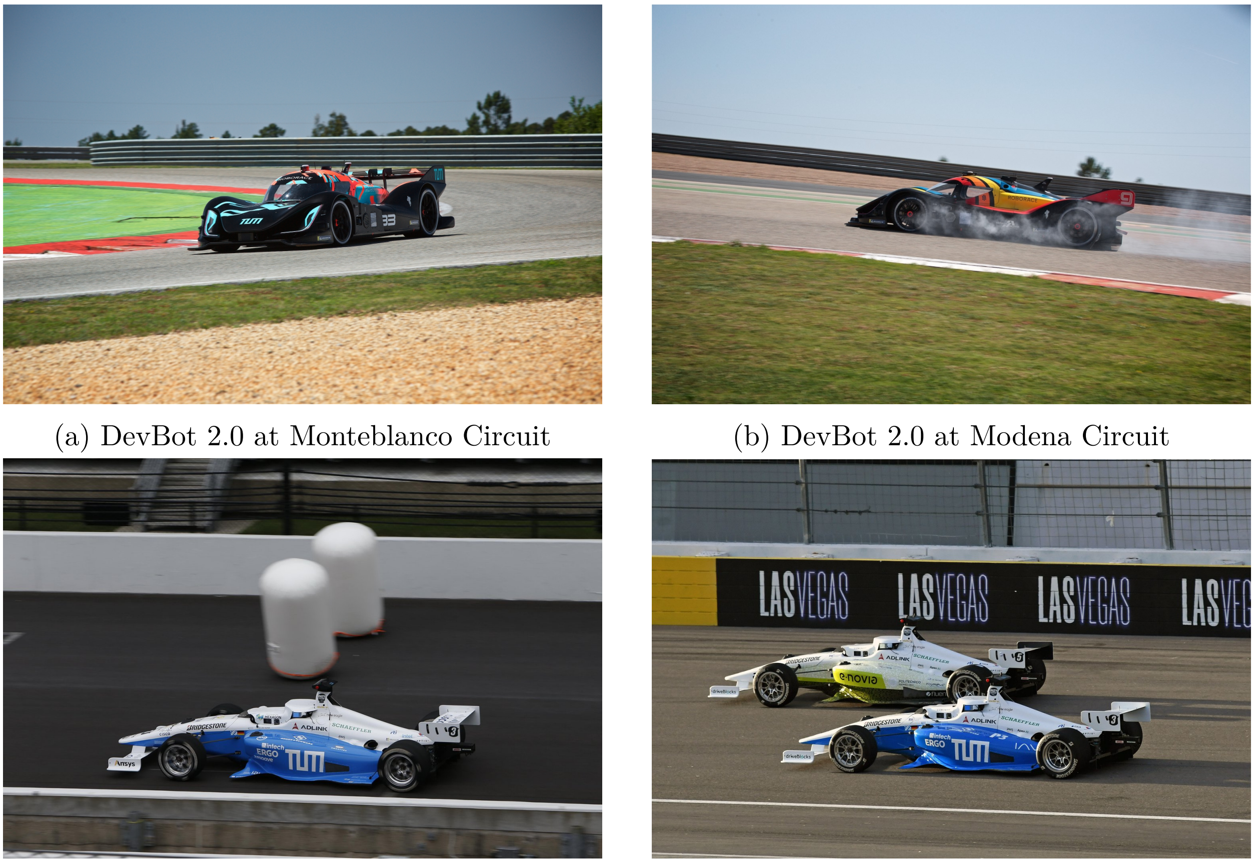
The capability levels of autonomous vehicles are commonly grouped into six autonomy levels according to the Society of Automotive Engineers (SAE) [20]. Current in-production vehicles show wide adaption of SAE level 2 systems, which accounts for joint automation of the lateral and longitudinal control of the vehicle motion under consideration of the behavior of other traffic participants in the driving scene. These are usually realized via systems like Lane Keeping Assist (LKA) and Adaptive Cruise Control (ACC). The primary characteristic of these systems is that they require constant supervision of the human driver, making them essentially an assistant to the human rather than self-driving systems. In contrast, level 4 and 5 systems are (by definition) non-supervised systems that come with significantly increased capability and safety requirements. While level 4 still allows the system to be restricted to certain driving situations (e.g. highway operation), the system has to handle all potential driving situations to be classified as a level 5 system. Level 3 specifies an intermediate capability level by allowing the system to ask for human assistance as long as this is done with sufficient time to react for the driver.

From a software perspective, the research community divides the driving task into three main areas: Perception, planning, and control (sometimes also referred to as sense, plan, act) [21]. While this notion is simplifying, it gives an overview of the tasks that have to be conducted by the autonomous system: First, the system has to perceive what is happening around the vehicle and how the current driving scenario looks. Second, it has to understand the potential future behaviors of other traffic participants and plan its strategy according to a long-term goal (e.g. reaching the destination). Third, this strategy has to be executed by applying appropriate steering, throttle, and brake commands. The work in this thesis is embedded in this last part, namely the vehicle motion control.

While control algorithms and structures have been fundamental to enable SAE level 2 automation, the focus of the research community shifted towards the perception and planning parts for SAE level 4 applications. However, as the maturity of those parts increases, the ability to control a vehicle in rare but challenging vehicle dynamics situations gains importance. This motivates researchers to focus on the enlargement of the so-called Operational Design Domain (ODD) of autonomous vehicles from comfort-oriented assistance systems (covering moderate accelerations up to $4\,\mathrm{m\,s^{-2}}$ [22]) to safety-critical motion control systems. These will be required to execute all kinds of evasion maneuvers with the complete vehicle acceleration potential (usually around $8\,\mathrm{m\,s^{-2}}$ to $10\,\mathrm{m\,s^{-2}}$). Even though these will not occur regularly, the wide-spread deployment of autonomous vehicles will require a thorough handling of rare situations.

Research groups have been using different strategies to work towards this goal. One of them is the fully integrated design of motion and vehicle dynamics control systems for drifting and evasion maneuvers [23, 24, 25, 26, 27]. Impressive real-world experiments have been conducted by [23] with an autonomous DeLorean as well as [24] with a BMW 5 series. Another strategy is the emergence of several autonomous racing platforms and competitions, which have fostered innovation in recent years. Probably the smallest, but at the same time the most widespread, are applications on scaled race-cars. The F1Tenth movement hosts regular virtual and real-world events to provide a cost-effective way of starting education and research projects within autonomous racing [28]. Similar platforms find wide use during lab research projects [29, 30]. An initiative with larger cars is the Formula Student Driverless competition [31, 32], held yearly at various racing venues within Europe. While the cars are faster and more sophisticated, this competition is restricted to single-vehicle scenarios, cones as track marks, and speeds of around $80\,\mathrm{km\,h^{-1}}$. These limitations are overcome by the Roborace competition [3], where teams have demonstrated qualifying laps and simplified multi-vehicle scenarios on a variety of different circuits around Europe. The vehicles are fully electric and based on LMP2 race car chassis, with a top speed of approx. $220\,\mathrm{km\,h^{-1}}$. Even faster cars have been retro-fitted by the Indy Autonomous Challenge (IAC) using an Indy Lights chassis and an automated driving kit. The series held races at the Indianapolis Motor Speedway (IMS) and the Las Vegas Motor Speedway (LVMS), two of the fastest oval racetracks of the world. The vehicles reached up to $270\,\mathrm{km\,h^{-1}}$ and the final round included a passing competition with two autonomous racing vehicles using only onboard sensors for opponent detection [33].

The work presented in this thesis has been embedded in the participation of a large research group (*TUM Autonomous Motorsport*) in the Roborace Championship, with DevBot 2.0 (Figure 1a and Figure 1b), as well as in the Indy Autonomous Challenge, with the Dallara-AV21 (Figure 1c and Figure 1d). The regular application in conjunction with a full software stack has led to the development of several well-proven and mature control algorithms for this thesis. The major challenges in these racing applications from a motion



(a) DevBot 2.0 at Monteblanco Circuit

(b) DevBot 2.0 at Modena Circuit

(c) AV-21 at Indianapolis Motor Speedway

(d) AV-21 at Las Vegas Motor Speedway

Figure 1: The TUM Autonomous Motorsport racing software driving on different vehicles and racetracks. *Image credits: Roborace / Indy Autonomous Challenge*

control perspective has been the presence of nonlinear and complex dynamic characteristics of the vehicle combined with the inherent uncertainties in these models. They can be attributed to two major sources: First, the force-generating mechanism of the tire is highly nonlinear, it depends on the current wear of the tire and on the environmental conditions. While there has been a lot of research (see e.g. [34]) into generating appropriate models, they usually overfit to individual situations and fail to accompany reliably for the outside factors such as temperature. Second, the suspension and tire dynamics tend to be too complex to fully include into real-time capable controller designs, which leads to simplifications and therefore uncertainty in the representation of the vehicle dynamics. This thesis investigates how data-driven and robust controller designs can be used to to overcome these challenges with a focus on applications under real-world circumstances. This continuous operation without safety-drivers on various racetracks has been one of the major design decision drivers within this work and outlines the high requirements towards reliability and performance.

1.2 Related work

The following section is going to review the state of the art in autonomous vehicle motion control, starting with classical feedforward and feedback design strategies such as PID-control and exact linearization, moving to Model Predictive Control (MPC) approaches covering nominal, robust and stochastic algorithms, and finally giving an overview about data-driven strategies in motion control. The presented work is mostly focussed on autonomous racing or high performance motion control application to keep it relevant and concise. Readers who are interested in learning more about the autonomous racing research community as a whole can use the recent review article [35] covering all areas from perception, motion planning up to control and learning based strategies.

1.2.1 Classical Motion Control

Fundamental work on longitudinal and lateral vehicle motion control has been conducted for more than fifty years [36, 37, 38]. The longitudinal and lateral dynamics have been considered separately within these works, with a strong focus on lateral control systems. The longitudinal dynamics have been handled via straight-forward P- or PI-control laws. The strong influence of the current velocity onto the lateral path tracking dynamics has been taken into account during the design of the feedback gains. The proposed control systems target the tracking of a lateral reference, usually in the middle of the lane, and model the road curvature as a disturbance acting upon the system. If this is known (e.g. via a vision-based system), a feed-forward compensation can be designed. The feedback handles the resulting double integrator dynamics from the path tracking task and the vehicle dynamics with varying levels of detail. The systems proposed in the following years follow a similar philosophy: Separation of longitudinal and lateral control but with advanced additions towards velocity-dependent feedback gains for the lateral dynamics as well as a look-ahead component in the control law to handle the double integrator dynamics [1, 39, 40, 41].

The next generation of motion control systems has aimed to enhance the performance by introducing more complex models for the vehicle dynamics by using nonlinear tire models and nonlinear representations of tire and chassis side-slip angles. They utilize effective input transformations [42] or exact linearization schemes [43, 44] to handle the nonlinearities. While these approaches showed promising results, they do not consider auxiliary control targets such as jerk minimization of the overall vehicle dynamics. In addition, they compensate for the nonlinearities within the dynamics rather than exploiting them which limits the maximum control performance. Furthermore, the implicit model inversion can lead to degraded control performance when uncertainties are present. A related approach is taken by flatness-based controller designs, where the controller is designed based on a flat system output and the corresponding target dynamics. It has been applied with success to autonomous vehicle control by [45, 46]. Another strategy proposed is sliding-mode control [47, 48], where the controller tries to track a pre-defined sliding surface with a control law dominating the remaining uncertainties. While this approach shows good robustness from a theoretical perspective, its often prone to actuator chattering. This phenomenon describes that the high-gain feedback amplifies the noise and requests varying actuator setpoints with high frequency. A comparison of several different control approaches can be found in [49]. In addition to these path-tracking focused approaches, there have been controller designs explicitly targeting unstable open-loop driving conditions in drifting situations [23, 24, 50]. However, these are limited to pre-defined drifting trajectories and have not been integrated into a generalistic motion control system that can handle all kinds of standard cruising scenarios, evasion maneuvers, and drifting situations.

1.2.2 Model Predictive Motion Control

In recent years, MPC gained a lot of attention from the research community and is nowadays considered one of the most important advanced control techniques for practical applications [51]. It promises to overcome the mentioned disadvantages and handle complex nonlinear systems, multi-input/multi-output systems, and state and input constraints seamlessly. These properties make it an ideal candidate for the field of autonomous vehicle control. In addition, it allows incorporating secondary objectives such as ride comfort or energy efficiency into the controller design. The first approaches towards the application of MPC for trajectory tracking of autonomous vehicles have been presented in the early 2000s by [52] and [53]. They use a nonlinear single track model and a Pacjeka tire model. Both suggest using successive online linearization to obtain real-time feasible approximations to the original optimization problem and demonstrate the applicability of their approach in extensive double lane-change simulations and experiments. The results have been extended to joint longitudinal and lateral vehicle control in [54, 55, 56]. With the advancement of available computational power, researchers turned towards applying more complex models and a significant interest in vehicle motion control for autonomous racing as a benchmark application emerged. The authors of [57] propose to use a contouring control approach with a nonlinear single track model to maximize the progress along the racetrack as well as a two-level controller system approach. These ideas are similar to [58] where a lower fidelity model is used to obtain a near time-optimal trajectory and an NMPC trajectory tracking controller computes the required control signals. The former approach has been demonstrated successfully using a real-time SQP scheme on a 1:43 scale test-bed. However, the resulting closed-loop trajectories differ from the time-optimal line in some corners and the optimization requires fine-tuning to operate reliably. An important modification to increase the numerical reliability of MPC problems for autonomous racing is presented in [59]: The authors reformulate the system dynamics using spatial discretization instead of time discretization. This technique has been successfully applied to several minimum lap-time optimal control problems. It allows formulating track bounds as box constraints on the lateral deviation with respect to a given reference path. Similar formulations have been used in [60, 61, 62]. Many more applications of of MPC can be found in the autonomous racing community, which proves the strenghts of the method with handling nonlinearities in the system under control [30, 63, 64].

An alternative towards the utilization of nonlinear tire models is the application of kinematic vehicle models as they show close-to-linear dynamics for the path tracking task. A predictive controller based on this is presented in [65], but the authors conclude that its application is limited to low-speed scenarios and propose to overcome the limitations by a hierarchical control strategy similar as in [55, 58]. However, other authors demonstrate comparable results for the use of kinematic and dynamic vehicle models [61, 66]. In [62] successful trajectory re-planning and obstacle avoidance on the racetrack is carried out using a point-mass model with appropriate joint lateral and longitudinal acceleration constraints. The system dynamics show considerable similarities to the kinematic vehicle model. These findings suggest that these simpler models might be sufficient to achieve good closed-loop control quality. Controllers using nonlinear tire models must have accurate parameter knowledge to outperform these more straightforward approaches, however, this seems challenging to realize when the parameters depend on environmental circumstances. Another strategy to improve the handling stability of an autonomous vehicle is the introduction of a stability envelope (formulated as a joint constraint on yaw rate and side slip angle) based on the vehicle and tire characteristics, which is then used as a constraint within the MPC [67].

The ability to consider constraints and the predictive nature motivates to utilize MPC as an approach to react to suddenly occurring obstacles as well as for trajectory refinement [55, 62, 68]. This extension of responsibilities comes with the challenge to ensure recursive feasibility of the receding horizon optimal control problem. A short foresight horizon could lead to an optimistic driving style neglecting corners or obstacles not yet within the horizon. A common approach is to mitigate this by the introduction of a terminal constraint requiring the end of the optimization horizon to coincide with a global (collision-free) path and that the velocity has to be lower or equal to the global velocity profile [58, 62]. More involved concepts determine appropriate terminal sets depending on a-priori knowledge about the road ahead using game-theory [69, 70].

The real-world application of the approaches presented above faces further difficulties created by deviations between the predicted and the actual vehicle behavior stemming from external disturbances and parameter uncertainties [71]. These become especially important at the limits of handling, where the tire nonlinearities have a significant impact on the vehicle behavior and are hard to identify accurately. It is well-known that this mismatch between predictions and actual behavior can lead to the loss of stability and recursive feasibility in the presence of state and input constraints [72]. This usually leads to a violation of the constraints, which can be safety-critical, e.g. in the case of collision constraints. While there is a significant body of theory towards robust MPC [73], the applications within autonomous racing have been limited due to overly conservative assumptions interfering with the aim to achieve maximum performance. A Tube-MPC approach for vehicle motion control based on the construction of a robust positive invariant set is presented in [25]. It leads to increased robustness of the controller in simulation and experiments compared to nominal MPC. A similar approach can be found in [74]. A different strategy with online calculation of a reachable set based Tube-MPC is presented in [75, 76]. In contrast to robust MPC, stochastic MPC introduces a risk parameter allowing for a certain level of constraint violations [77, 78, 79]. The resulting controller can trade-off safety and performance. A related strategy is the introduction of a randomized MPC formulation to approximate the fully stochastic solution efficiently [80]. Another approach is presented based on Contingency-MPC [81], which solves the receding horizon optimal control problem simultaneously for the nominal problem and a backup plan to ensure that the solution to the nominal problem always allows switching to this backup plan.

1.2.3 Data-Driven Motion Control

The inherent parameter and model uncertainties also motivate the application of data-driven methods within autonomous racing. One of the earliest approaches was the application of Iterative Learning Control (ILC) to the lap-based autonomous racing problem [82, 83]. The vehicle monitored the deviation from the target path on every lap and adjusted the steering behavior successfully by using a PD-ILC and an Q-ILC. The algorithms were tested on an Audi TTS prototype vehicle at Thunderhill Raceway. Another lap-based approach was presented by [29, 84, 85, 86] via data-driven identification of a non-conservative terminal set and cost function. It has been tested on a 1:10 scale autonomous RC-car with significant improvements compared to a non-learning based strategy. The disadvantage of both approaches lies within their restriction towards repetitive driving tasks which make them hard to apply in public road autonomous driving applications or multi-vehicle autonomous racing including opponent interactions.

An different approach using prior model knowledge (a nonlinear vehicle dynamics model) and online refinement via a residual Gaussian Process Regression (GPR) is presented in [87, 88]. While it demonstrates good performance improvements, it shows difficulties in achieving a real-time capable implementation for larger data amounts. This is related to the fact that GPR computational times scale with the number of data points and therefore require a thorough data storage and update concept. A proposal to overcome this limitation is the application of sparse spectrum GPR [89]. In contrast to those approaches starting with an already detailed vehicle model, [90] presents a strategy which purposely utilizes a simpler kinematic vehicle model where all parameters can easily be measured and lump all uncertainties within the learning of the residual GPR error model. Other authors have used similar identification techniques using neural networks instead of gaussian processes to model vehicle dynamics models for simulation and controller design [91, 92].

Instead of estimating a generalistic error model, other authors propose to use data-driven techniques to identify parameters or tire characteristics directly. [93] uses an adaptive neural network to estimate the uncertain tire parameters in the nonlinear region of the vehicle dynamics. A wide spread target in this field is the identification of the maximum tire-road friction potential while driving [94, 95] with effect-based methods based on Kalman Filters or Machine Learning techniques monitoring the vehicle and wheel behavior [96, 97, 98]. An alternative approach is the careful analysis of the road with secondary sensors such as cameras [99].

A different strategy is utilized in [100]: Instead of extending an MPC scheme with residual models, a policy from images to target trajectories is learned with a deep-learning-based model. This approach covers all aspects, from track detection to motion planning. However, this end-to-end approach has difficulties generalizing between different tracks and has yet to be proven applicable under real-world conditions. Other approaches based on reinforcement learning have been presented in [101, 102, 103]. As this is out-of-scope for our work, the reader is referred to the review [35] for more background and further literature on the topic.

1.3 Research Objectives and Outline

There is a large amount of research within the motion control space for autonomous vehicles focusing on comfort-oriented driving situations. While there are recent extensions towards the operation close to the handling limits using exact linearization and model predictive control, the state-of-the-art requires handcrafted nonlinear models which tend to be brittle when used in varying environmental conditions or significant parameter uncertainty. This becomes especially challenging in the case of autonomous racing, where the tires are used continuously in their nonlinear operating region and are subject to significant variation in grip depending on the track characteristic and temperature. The contributions of this thesis will answer the research questions outlined in this section.

While there has been an extensive body of research on the application of nominal nonlinear model predictive control in real-world tests, research considering robust model predictive control have so far been limited to simulation or small scale studies. The central consideration behind using a well-designed robust control method is to acknowledge the fact that there will always be uncertainty within the prediction model when operating a racing vehicle at the handling limits. Instead of working towards increasingly complex and hard to parameterize models, the uncertainty can be included in the design by the utilization of an explicit uncertainty representation which opens up the possibility for new trade-offs between model complexity, uncertainty representation and real-time capabilities. In addition, the introduction of fast low-level control loops seems to be a promising direction of research which has not been well explored for the case of robust model predictive control as high-level controller. These thoughts can be summarized as follows:

- Can Robust MPC help to overcome the shortcomings of Nonlinear MPC when faced with parameter and model uncertainty?
- How should state estimation and low-level controllers look like to achieve maximum robustness, real-time capability and safe operation in conjunction with a Robust MPC on a full scale autonomous racing vehicle?

In addition, the potential of data-driven algorithms should be further investigated. While they have become the de-facto standard in the area of autonomous vehicle perception and traffic participant prediction, their application in high-performance motion planning and control algorithms has been limited so far. We will investigate the applicability of model-free and model-based learning strategies to continuous adjustment to changing environmental conditions such as temperature with these two research tasks:

- Can model-free data driven methods increase the performance of a well-functioning autonomous racing motion planning and control system by refining hyperparameters based on continuous monitoring of the motion control performance?
- Can a real-time capable model-based learning method capture the vehicle dynamics model prediction error and adjust the uncertainty assumptions reliably during operation? How can this result be used within an existing motion control system for performance improvements?

The thesis is structured into two major streams of work (see Figure 2). Chapter 2 describes the fundamentals of vehicle dynamics, the utilized robust model predictive control methods and the background on the machine learning approaches applied. The main part summarizes the contributions of five publications: The papers in Section 3.1 to Section 3.3 focus on the development of a holistic robust control system for an autonomous race car, covering state estimation, robust model predictive control and the influence of multi-vehicle scenarios on the controller performance. The papers in Section 4.1 and Section 4.2 demonstrate independent contributions in the area of data-driven methods for autonomous racing. The former presents a model-free method to achieve the handling limits while considering certain safety indicators, the latter a method for reliable real-time quantification of non-gaussian uncertainties to decrease the model mismatch during runtime of the system. The part is concluded in Chapter 4.3 with a joint evaluation of the algorithms proposed in Section 3.3 and Section 4.2. The discussion of the presented work is presented in Chapter 5 andthe conclusions drawn from the research and potential future work are presented in Chapter 6.

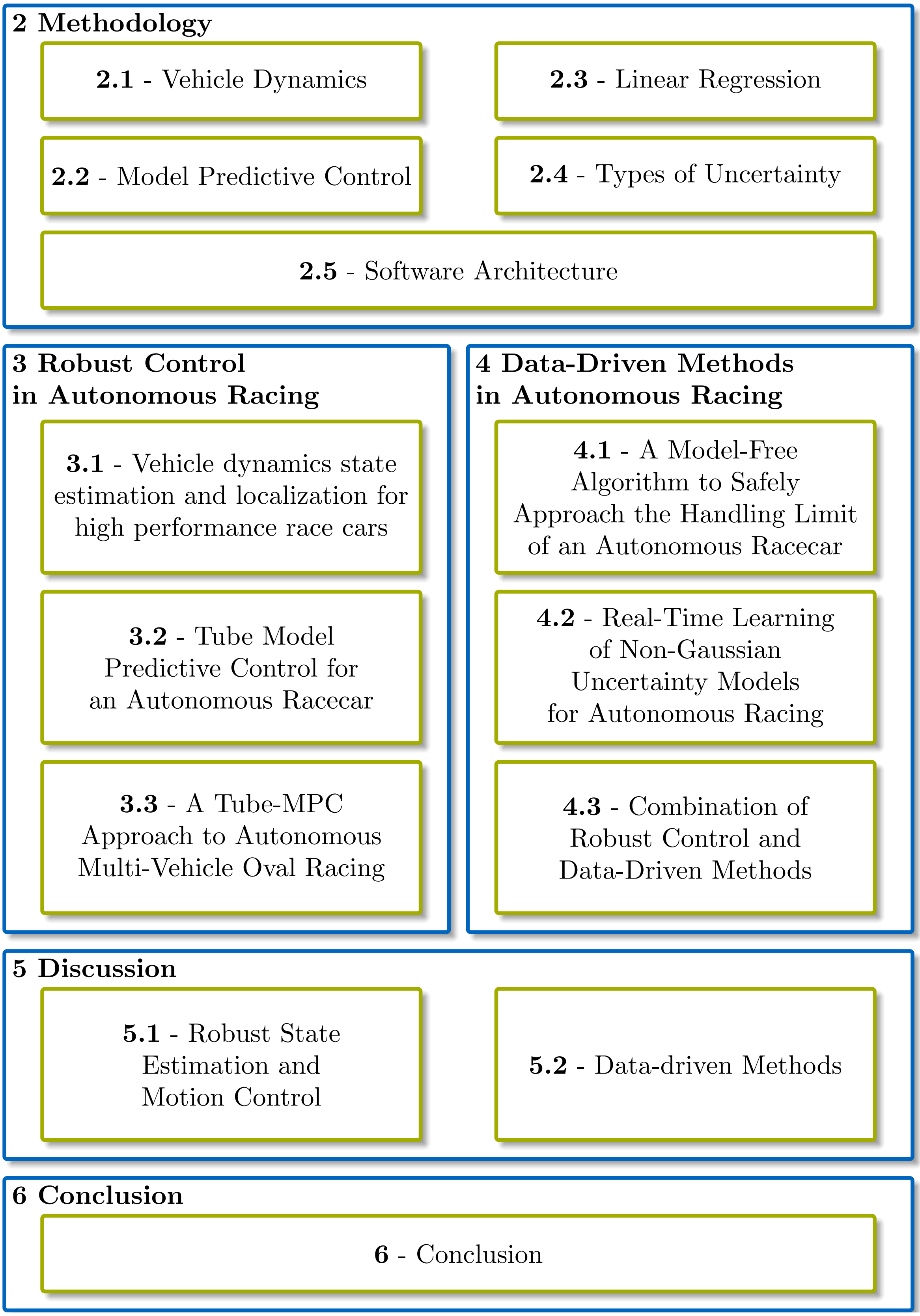

2 Methodology
2.1 - Vehicle Dynamics
2.3 - Linear Regression
2.2 - Model Predictive Control
2.4 - Types of Uncertainty
2.5 - Software Architecture
3 Robust Control in Autonomous Racing
3.1 - Vehicle dynamics state estimation and localization for high performance race cars
3.2 - Tube Model Predictive Control for an Autonomous Racecar
3.3 - A Tube-MPC Approach to Autonomous Multi-Vehicle Oval Racing
4 Data-Driven Methods in Autonomous Racing
4.1 - A Model-Free Algorithm to Safely Approach the Handling Limit of an Autonomous Racecar
4.2 - Real-Time Learning of Non-Gaussian Uncertainty Models for Autonomous Racing
4.3 - Combination of Robust Control and Data-Driven Methods
5 Discussion
5.1 - Robust State Estimation and Motion Control
5.2 - Data-driven Methods
6 Conclusion
6 - Conclusion

Figure 2: Structure of the thesis

2 Methodology

2.1 Vehicle Dynamics

This section is going to introduce the basic formulas and terminology of the vehicle dynamic models used in the subsequent papers. More details can be taken from the available textbooks (e.g. [34, 104, 105]) on this topic by readers who are interested in an in-depth presentation of this subject.

2.1.1 Kinematic Bicycle Model

One of the most wide-spread models for autonomous vehicle behavior is the kinematic bicycle model [106] depicted in Figure 3. It models the vehicle motion in cartesian coordinates p_1 and p_2 and the orientation of the chassis ψ with respect to the p_1 coordinate axis. The vehicle velocity is denoted with v, the longitudinal force acting upon the system with F_x and the vehicle mass with m. It lumps the left and right tires on each axle and assumes tire side-slip to be zero which results in the tire velocity vectors to be aligned with the tire longitudinal axis. The resulting movement can be obtained from the kinematic relations for the center of rotation. The chassis side-slip β is a fixed relation between the front wheel steering angle δ and the position of the center of gravity specified via the distance to the front-axle l_F and the rear-axle l_R.

$$\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \dot{\psi} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v\cos(\psi+\beta) \\ v\sin(\psi+\beta) \\ \frac{v}{l_\mathrm{F}+l_\mathrm{R}}\delta \\ \frac{F_x}{m} \end{bmatrix}, \tag{2.1a}$$

$$\beta = \arctan\left(\frac{l_\mathrm{R}\tan(\delta)}{l_\mathrm{F}+l_\mathrm{R}}\right). \tag{2.1b}$$

This model is often used for low-speed applications as it covers the non-holonomic nature of the vehicle motion well and shows sufficient accuracy as long as the tire side slip angles are small. Its simplicity also promoted the utilization in the robotics community for a variety of other vehicles, such as warehouse robots or small tractors within logistic centers. It can also be extended to serve rear-wheel or all-wheel steering easily, to cover e.g. forklifts and agricultural machinery.

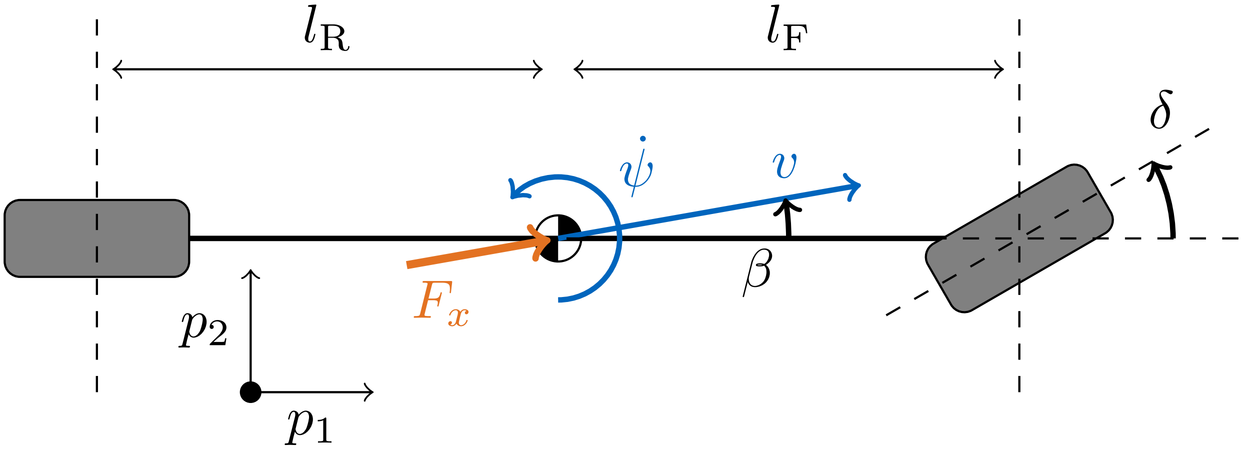


Figure 3: Kinematic bicycle model

2.1.2 Point Mass Model

While the kinematic model is well-suited for vehicle dynamics simulation and algorithms which require a feed-forward prediction of the future behavior, the strong nonlinearities within the position state dynamics and the coupling of the steering angle and the chassis side slip angle make it hard to apply linear control engineering techniques. This issue can be overcome by a reformulation of the dynamics into a point mass model using a curvilinear coordinate frame, aligned with a reference path parameterized via $p_1(s)$ and $p_2(s)$ or a curvature profile $\kappa(s)$, which implicitly specifies the reference path. In this framework, s depicts the coordinate along the reference, d the orthogonal deviation from the reference and $\Delta\psi$ the deviation of the chassis heading from the path heading. The resulting dynamics can be written as:

$$\begin{bmatrix} \dot{s} \\ \dot{v} \\ \dot{d} \\ \Delta\dot{\psi} \end{bmatrix} = \begin{bmatrix} \left(1 - d\kappa(s)\right)^{-1} v\cos(\Delta\psi + \beta) \\ \frac{F_x}{m} \\ v\sin(\Delta\psi + \beta) \\ \frac{v}{l}\delta - \kappa(s)\dot{s} \end{bmatrix}, \tag{2.2a}$$

$$\beta = \arctan\left(\frac{l_R \tan(\delta)}{l_F + l_R}\right). \tag{2.2b}$$

This choice of coordinates is depicted in Figure 4. The advantage is that it reveals the close to linear structure of the trajectory tracking problem if the deviation between the vehicle position and the path is sufficiently small. This important fact is outlined in the following using a couple of assumptions and transformations in the remainder of this paragraph. By

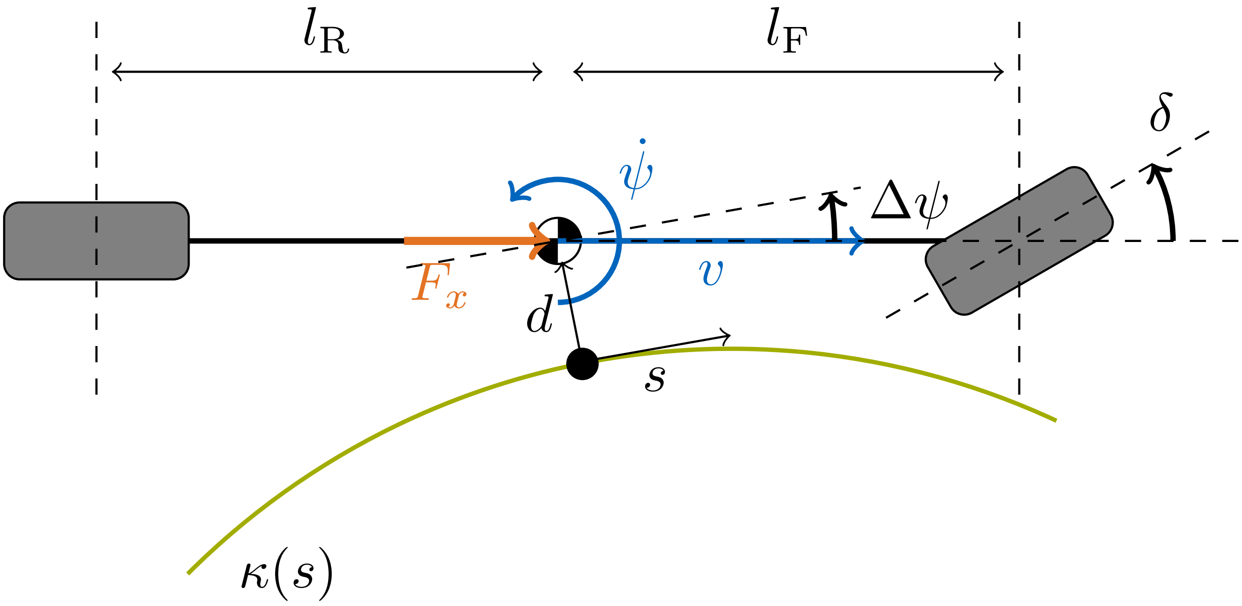


Figure 4: Point mass model in curvilinear coordinates

assuming $\beta \approx 0$, $d \approx 0$ and $\Delta\psi$ to be small, the system can be rewritten as:

$$\begin{bmatrix} \dot{s} \\ \dot{v} \\ \dot{d} \\ \Delta\dot{\psi} \end{bmatrix} = \begin{bmatrix} v \\ \frac{F_x}{m} \\ v\Delta\psi \\ \frac{v}{l}\delta - \kappa(s)v \end{bmatrix}. \tag{2.3}$$

In addition, the velocity is assumed to be constant and the state transformation $\dot{d} = v\Delta\psi$ with the derivative $\ddot{d} = v\Delta\dot{\psi}$ is applied. This results in

$$\begin{bmatrix} \dot{s} \\ \dot{v} \\ \dot{d} \\ \ddot{d} \end{bmatrix} = \begin{bmatrix} v \\ \frac{F_x}{m} \\ \dot{d} \\ \frac{v^2}{l}\delta - \kappa(s)v^2 \end{bmatrix}. \tag{2.4}$$

Finally, the last line of (2.4) can be rewritten by using the lateral acceleration as input instead of the steering wheel angle $a_{\mathrm{y,C}} = \frac{v^2}{l}\delta$. With the control law $a_{\mathrm{y,C}} = \kappa(s)v^2 + \Delta a_{\mathrm{y,C}}$, it becomes clear that the system can be formulated in linear form with respect to the corrective acceleration $\Delta a_{\mathrm{y,C}}$. The model can now be written as

$$\begin{bmatrix} \dot{s} \\ \dot{v} \\ \dot{d} \\ \ddot{d} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ v \\ d \\ \dot{d} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} F_x \\ \Delta a_{\mathrm{y,C}} \end{bmatrix}. \tag{2.5}$$

This reformulation gives important insights on the application of classical control methods such as PID and state-space feedback for vehicle motion control. First, it shows that the longitudinal and the lateral dynamics can be assumed decoupled for sufficiently small deviations from the target trajectory and small velocity changes. Second, it shows that both are double-integrator dynamics which is helpful information for the controller design.

2.1.3 Single Track Model

The single-track model (Figure 5) drops the fundamental assumption of the kinematic vehicle model that the tire motion vector is rigidly aligned with the tire orientation. Instead, the tire is allowed to move in the lateral direction and can therefore create side-slip. The resulting angle between the longitudinal and lateral motion of the tire is named the side-slip angle α. It was found to be the primary influence on the force-generating mechanisms in the rubber-based component for vehicle-road interaction [34]. There is a variety of different tire models and Section 2.1.4 will discuss the most relevant of them. This shift in the modeling strategy leads to a significant extension of the previously presented kinematic bicycle model via the split of the motion velocity into longitudinal v_x and lateral velocity v_y and the addition of yaw rate $\dot{\psi}$ dynamics:

$$\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \dot{\psi} \\ \dot{v}_x \\ \dot{v}_y \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} v_x \cos(\psi) - v_y \sin(\psi) \\ v_x \sin(\psi) + v_y \cos(\psi) \\ \frac{1}{m}\left(F_x - \sin(\delta)\, F_{\mathrm{y,F}}(\alpha_{\mathrm{F}})\right) + v_y \dot{\psi} \\ \frac{1}{m}\left(\cos(\delta)\, F_{\mathrm{y,F}}(\alpha_{\mathrm{F}}) + F_{\mathrm{y,R}}(\alpha_{\mathrm{R}})\right) - v_x \dot{\psi} \\ \frac{1}{J}\left(\cos(\delta)\, F_{\mathrm{y,F}}(\alpha_{\mathrm{F}})\, l_{\mathrm{F}} - F_{\mathrm{y,R}}(\alpha_{\mathrm{R}})\, l_{\mathrm{R}}\right) \end{bmatrix}. \tag{2.6}$$

In addition to the previously introduced variables, J denotes the moment of inertia and $F_{\mathrm{y,F}}$ and $F_{\mathrm{y,R}}$ the lateral forces at the front and the rear axle as a function of the tire side-slip angles α_{F} and α_{R}. A similar formulation in path-aligned coordinates as in (2.2) is common [31] but will be omitted for the sake of brevity. Other variations are found in the literature using the vehicle chassis side slip β and the absolute velocity v as state variables instead of the longitudinal and lateral chassis velocity [6]. The transformation can be done by straight-forward application of basic geometric relations. The relation between the tire

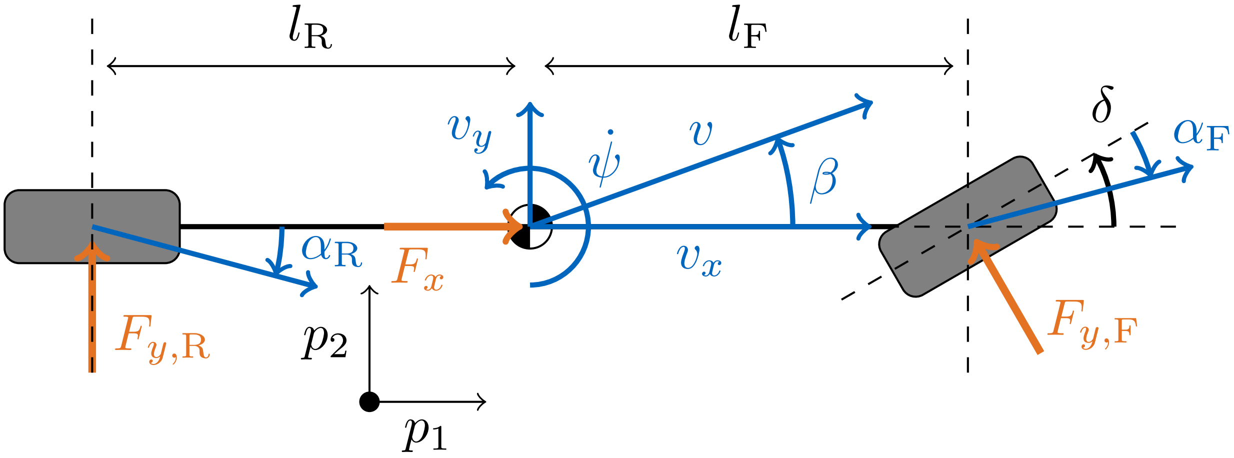


Figure 5: Single track model

side-slip angles and the chassis movement can be written as

$$\alpha_\mathrm{F} = \delta - \arctan\left(\frac{v_y + l_\mathrm{F}\dot{\psi}}{v_x}\right), \tag{2.7a}$$

$$\alpha_\mathrm{R} = -\arctan\left(\frac{v_y - l_\mathrm{R}\dot{\psi}}{v_x}\right). \tag{2.7b}$$

2.1.4 Tire Models

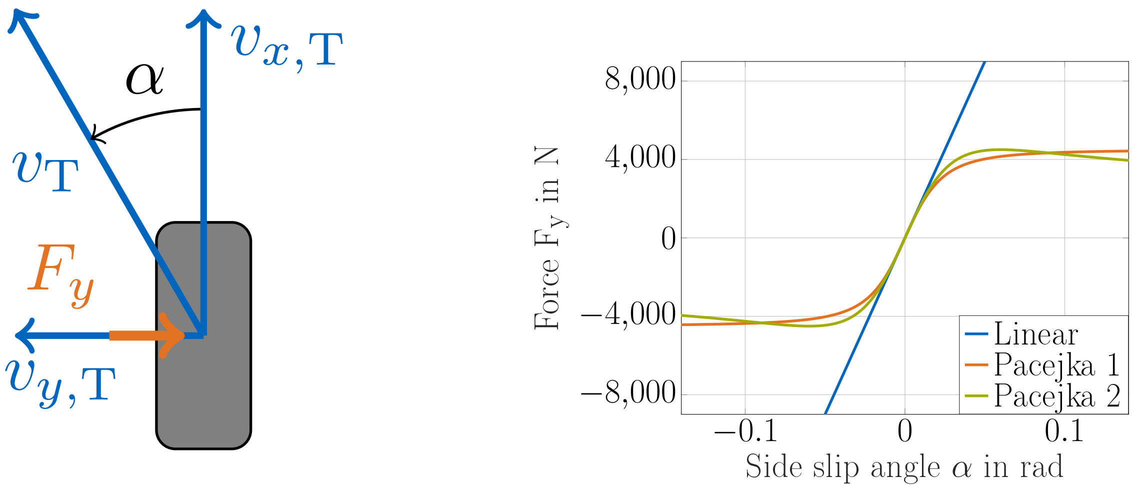
The main mechanism responsible for the generation of contact forces is slip [34]. There are two different formulations, one for the longitudinal and one for the lateral slip. The longitudinal slip is usually defined as the ratio between the speed of the tire contact patch and the speed over ground $v_{x,\mathrm{T}}$,

$$\lambda = \frac{\omega r_\mathrm{T}}{v_{x,\mathrm{T}}}, \tag{2.8}$$

with the wheelspeed ω and tire radius r_T. The lateral slip is usually defined as the so-called side slip angle of the tire

$$\alpha = \arctan\left(\frac{v_{y,\mathrm{T}}}{v_{x,\mathrm{T}}}\right), \tag{2.9}$$

with $v_{y,\mathrm{T}}$ being the lateral speed over ground. The calculation of this quantity depends on the modeling fidelity of the overall vehicle model. The corresponding equations for the single-track model are given in (2.7). A visualization of the calculation of the lateral side slip angle and the resulting force is depicted in Figure 6a.



(a) Tire side-slip definition

(b) Tire model comparison

Figure 6: Tire models and definitions

The most common class of tire models for vehicle dynamics analysis and control system designs are semi-physical models. They are defined via an empirical model using function approximation with a hand-crafted set of well-chosen, meaningful basis functions. Full physical models are usually too complex to apply in real-time and are therefore used only for tire development. The discussion in the remainder of this section will be limited to lateral tire models, as the description of longitudinal tire models follows the same structural techniques.

The simplest model is the linear tire model, where the tire force is calculated using the cornering stiffness c as a linear relation to the side slip in the following way:

$$F_y = c\alpha. \tag{2.10}$$

This model is accurate for low to moderate tire utilization of approx. 30 % to 50 % of the maximum tire force [104], which corresponds to accelerations of $3\,\mathrm{m\,s^{-2}}$ to $5\,\mathrm{m\,s^{-2}}$ on the overall vehicle level for passenger vehicles. It has found wide adoption for the analysis of passenger vehicle dynamics in the well-known linear single-track model. It is a simplified version of (2.6), modeling only the relation between the steering angle input and the yaw rate and vehicle chassis side slip response and neglecting the longitudinal dynamics. Its linear nature and validity domain makes it a good balance between complexity and accuracy for comfort-oriented lateral vehicle motion control systems.

The extension of the validity domain to the nonlinear operating region of the tire can be done via a four coefficient Pacejka model or the fiala brush tire model [34], but many variants of these formulations are in use within the control engineering and vehicle dynamics community. All of them target at accounting for the saturating effects on the forces for large side-slip angles, a key characteristic of the tire close to the friction limit. The four coefficient Pacejka model can be written as:

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

This model depicts the cornering stiffness as $c = BCD$ and the maximum transmittable force via D. The remaining coefficients are referred to as shape factors. Two examples (Pacejka 1 and Pacejka 2) and a comparison to the linear version are depicted in Figure 6b. This tire model is often used for the design of vehicle motion control systems close to the handling limits of the vehicle [31, 43, 60], as it captures the basic dynamics in the linear range as well as the behavior at the limit when the tire force saturates. However, their strong nonlinear characteristic often leads to numerical challenges within the control design process. In addition, the resulting controllers are usually quite sensitive to the exact calibration of the parameters, which makes it challenging to apply these models when the environmental influences are uncertain or time-varying. Even more detailed versions with up to 50 coefficients are used for vehicle dynamics simulations [34]. They are identified via extensive test-rig experiments and fitted via mathematical regression techniques.

When used in a motorsport context instead of passengers cars, tires are usually utilized in the longitudinal and lateral direction at the same time, e.g., when braking while approaching a sharp turn. This affects the tires maximum force generation capabilities as the stretching of its road contact patch is done in two dimensions simultaneously. This effect can be accounted for via the introduction of a relation between the maximum longitudinal and lateral tire force [9] via the following equation:

$$\left(\frac{F_x}{\bar{F}_x} \right)^p + \left(\frac{F_y}{\bar{F}_y} \right)^p \leq 1. \tag{2.12}$$

This equation relates the current tire forces F_x and F_y with the peak forces $\bar{F}_x$ and $\bar{F}_y$ and becomes the well-known Kamm's circle for the case $p = 2$ [104]. It also clarifies that (2.11) only holds for pure lateral tire loads and has to be extended to a function of the lateral and longitudinal slip $F_y(\alpha, \lambda)$ to truly reflect the real tire behavior. While these effects are accounted for in complex simulation frameworks, the resulting models are often too complex to use in a control design.

2.2 Model Predictive Control

The fundamental target of control engineering is the design of a control law which ensures that the closed-loop dynamics of a (nonlinear) dynamic system correspond to the desired stability and convergence properties, even when uncertainties or disturbances are present.

For the following discussion, the open-loop dynamic system is denoted as:

$$\dot{x} = f(x, u, d), \tag{2.13a}$$

$$y = h(x) \tag{2.13b}$$

with the system state x, the control input u, the external disturbances d and the system output y. The function $f(x, u, d)$ characterizes the system dynamics and the function $h(x)$ the state to output relation. Most controller design strategies (e.g. linear quadratic regulators, sliding mode control, exact linearization, ...) aim at obtaining a control law $u = k(x)$ which is given in the form of a explicit function. This makes it easy to analyze the closed-loop properties and computationally cheap to apply the controllers on embedded hardware. However, it is difficult to consider system and input constraints in complex systems, handle the trade-offs between conflicting control targets at runtime or utilize knowledge about the future target trajectory of the system. MPC promises to address these challenges by a conceptual change to the controller design paradigm [107]: It reformulates the target behavior of the system under control as a finite-horizon optimization problem based on the open-loop system dynamics and constraints on the control and state variables. Using this optimization problem, it defines an implicit control law $u(x)$ via repeatedly applying the first input u from the optimal solution. This section introduces the standard formulations for nominal and robust MPC, which lay the basis for the controller designs in the main part of the thesis.

2.2.1 Nominal MPC

The MPC problem for the nominal case (no disturbances acting upon the system and no uncertainties present) can be written as

$$\min_{\bar{u}} \sum_{i=0}^{N-1} \left(x_i^T Q x_i + u_i^T R u_i \right) + x_N P x_N, \tag{2.14a}$$

$$\text{s.t.} \quad x_0 = x(k), \tag{2.14b}$$

$$x_{i+1} = f_{\mathrm{d}}\left(x_i, u_i\right), \quad \forall i \in [0, N-1], \tag{2.14c}$$

$$x_i \in \mathbb{X}, \quad \forall i \in [0, N-1], \tag{2.14d}$$

$$u_i \in \mathbb{U}, \quad \forall i \in [0, N-1], \tag{2.14e}$$

$$x_N \in \mathbb{X}_{\mathrm{T}}, \tag{2.14f}$$

with the quadratic stage cost matrices Q and R, the terminal cost matrix P, the discretized system dynamics $f_{\mathrm{d}}(x, u)$, the stage constraint sets $\mathbb{X}$ and $\mathbb{U}$ as well as the terminal constraint set $\mathbb{X}_{\mathrm{T}}$. The index i is preferred over the notation $x(k)$ to refer to the prediction of the system behavior as part of the optimization structure to clarify the difference to the actual system

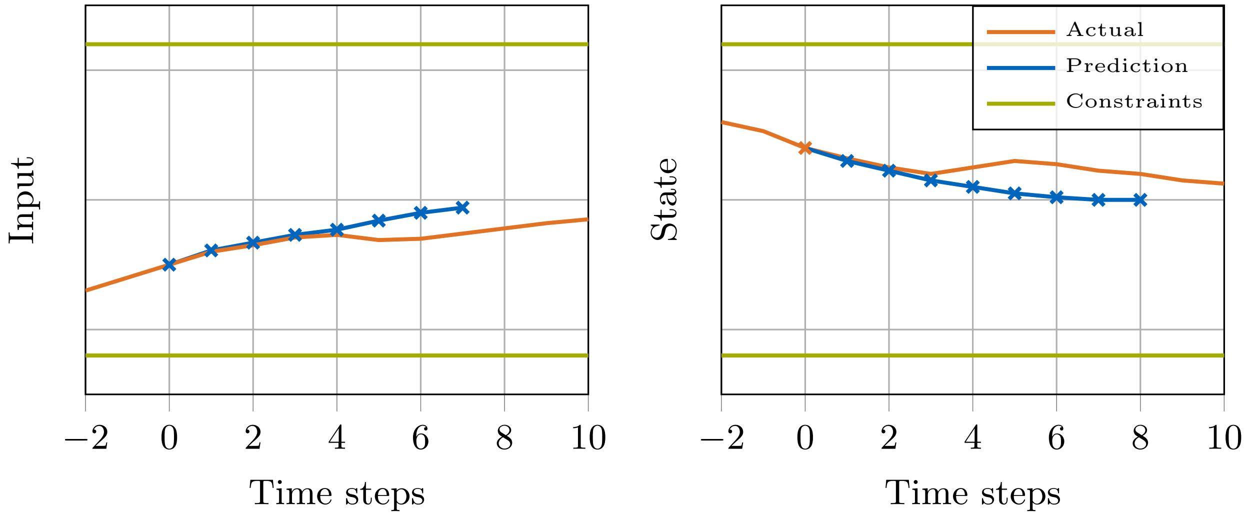


Figure 7: Visualization of the model predictive control concept. The blue predictions visualize the optimization variables and predictions of the system behavior at time instant 0. The orange graph depicts the actual behavior of the system observed after repeated application of the control inputs.

behavior $x(k)$ at the time instant k. The input sequence u_0 to u_{N-1} is lumped to a vector $\bar{u}$ and serves as the set of optimization variables. The control input is now defined as the first element of this vector obtained as the solution to the above optimization problem and can be written as $u(x) = u_0$.

The idea behind this optimization based concept is visualized in Figure 7. At time instant 0, the above optimization problem is solved while considering the system dynamics as well as the constraints with a prediction horizon of 8 steps. The control inputs serve as optimization variables and are varied such that the resulting state response minimizes the cost function. The cost design is usually chosen such that a trade-off between fast convergence of the state variables towards the target and the control efforts is found. The controller now applies the first element of the optimal input sequence to the system until the next state measurement (at time instant 1) is observed. This measurement allows the controller to take into account deviations from the prediction model and react in a feedback loop. The controller now starts a new optimization process, applies the first element of the new optimal sequence and waits until state feedback is observed.

This strategy is well-suited for handling nonlinear and constrained dynamics as the optimization framework allows (theoretically) for arbitrary complex system models. It is a straight-forward consequence that the control performance depends heavily on the quality of the prediction model as well as the influence of external disturbances upon the system. The example depicted in Fig. 7 shows the effects of such external disturbances at time instant 4 and 5. The state deviates significantly from the assumed behavior and therefore the MPC has to react to this by increasing the control action. If the disturbances have significant effect on the future behavior, they can impact the overall performance heavily. Strategies to mitigate those impacts are discussed in Section 2.2.2.

In addition to the prediction model, the horizon length is one of the most important hyperparameters in the MPC design process. From a theoretical perspective it is beneficial to choose the optimization horizon N to be as large as possible to decrease the approximation error to the infinite horizon problem. However, computational restrictions usually limit the maximum number of optimization variables and therefore the horizon length. This trade-off can be understood as follows: If the MPC optimization problem can not account for important aspects of the dynamics in the future, the solution is likely to be sub-optimal in a global sense compared to the infinite horizon solution. In addition, it can be hard to ensure that the control law acts such that a solution to the MPC optimization problem will exist for all future time instants if state constraints are present. The analysis of this problem is referred to as recursive feasibility analysis within the literature [107, 108] and is of significant practical relevance, even though it is often hard to formally ensure under real-world conditions. There are two main strategies to handle this issue: First, the design of appropriate terminal cost and terminal constraints can ensure that the optimization problem shows sufficient understanding of what happens after the end of the optimization horizon or is sufficiently cautious towards the end of the horizon [73, 107, 109]. Second, certain cost structures in conjunction with sufficiently long control horizons can ensure that the optimization horizon is long enough to consider all relevant effects [110, 111]. From a practical perspective, the optimization horizon is usually selected to be sufficiently long such that the prediction converges to a steady-state equilibrium prior to reaching the end of the horizon.

Even though the formulation in form of an optimization problem is an elegant solution from a control theoretic perspective, it comes with challenges from a numerical optimization point of view. When the system dynamics or constraints are nonlinear, the optimization problem has to be formulated as a nonlinear program (NLP). There are many different strategies to approach these problems, the most common scheme in MPC applications is the real-time iteration [112, 113]. Its a modified sequential quadratic program (SQP) strategy, where the problem is linearized around a trajectory and a local quadratic program is solved. In contrast to classical SQP, the real-time iteration only performs a single linearization and a single optimization for each solution of the optimization problem. This strategy is viable due to the special characteristics of the MPC task, where the solution in the current timestep is often close to the previous solution. The implementation of the optimization problems can either be done via holistic frameworks such as *acados* [114] or the utilization of efficient quadratic program (QP) solvers such as *HPIPM* [115] or *OSQP* [116]. While the former is suitable for standard problem formulations and allows rapid-prototyping of several different system dynamics, the manual integration of the QP solvers allow more efficient implementations and fine-tuning of the linearization strategy to the specific problem.

2.2.2 Robust MPC

The central part of MPC is the utilization of a dynamic prediction model to reason over the potential future behavior of the system under control. However, real-world applications always imply model uncertainty and external disturbances. The former manifests in various ways, from parametric uncertainty up to non-modeled dynamics and might destabilize the closed-loop system if too large. The latter is slightly less critical, as its impact is usually limited to performance degradation (as long as the disturbance is reasonably small). However, both types of deviation from the prediction model can break the recursive feasibility property of the nominal MPC [109, 117] which will lead to a failure to satisfy the system constraints and failure to generate appropriate control actions from the optimization. While the latter can be mitigated via the introduction of slack variables [118], the former is a significant challenge from a theoretical as well as from an engineering perspective.

The solution to address those constraint violations is the use of robust MPC methods [73, 107]. The resulting optimization problems ensure that the closed-loop dynamics have a certain amount of caution and leave sufficient room to counter-act external disturbances and model uncertainties without input or state constraint violations. After initial works on reformulations of the standard problem in (2.14) as a minimax problem to account for all potential disturbances [72, 119], the dominant stream of research nowadays is based on Tube-MPC approaches [108, 109, 117, 120]. The basic idea is to replace the complex optimization over a high-dimensional disturbance set and arbitrary control policies with an optimization over a set of uncertain predictions and a fixed pre-stabilizing control policy $u = \kappa(x) + v$ with the virtual input v. While [109] and [121] leverage a robust positive invariant set to create a tractable optimization problem, [117] and [120] use reachable sets for the same purpose. The second strategy is easier to apply within control engineering applications as it allows to modify parameters during runtime and can handle less theoretical rigorous implementations for nonlinear systems, e.g. linearization schemes along trajectories similar to the Extended Kalman Filter [122]. In addition, it does not require to calculate a robust positive invariant set as the method proposed by [109], which is a challenging task for nonlinear systems and can lead to overly conservative results. Following these arguments, the reachable set strategy by [117] has been used in this thesis and therefore the presentation in this section will be restricted to this approach. Readers interested in details on different strategies are referred to the literature mentioned previously.

With respect to the expression of the reachable sets, multiple strategies are used in the literature: The computationally most efficient one is the representation using level sets [120, 123]. However, they tend to be overly conservative in case of larger numbers of states. In contrast, polytopes are the most flexible and accurate representations, however, also the computationally most expensive [124]. To allow suitable trade-offs, other represention are used within the literature: [125] bases the proposed constraint tightening on ellipsoidal reachsets, as they can be represented with a fixed complexity representation (a matrix of dimension $\mathbb{R}^n$) and the impact of the system dynamics can be expressed analytically and computationally efficient. Another intermediate complexity representation are zonotopes, which provide a lot more flexibility but the required set arithmetics usually lead to an increase in representation complexity (as the required generators need to be combined) [124, 126]. The work in this thesis will utilize ellipsoids due to its fixed representation complexity and computationally efficient arithmetics.

It should be noted at this point, that the approaches presented in this thesis are restricted to the application of Tube-MPC to linearized system dynamics. This simplifies the addition of robust elements to the optimization problem and ensures that it stays computationally tractable. It is similar to the schemes applied within state estimation within the Extended Kalman Filter (EKF) framework. At the same time, the uncertainty framework naturally allows to lump uncertainties stemming from the linearization process with already present uncertainties within the model itself. The reader is referred to [76, 108, 120] for a more in-depth discussion of recent achievements towards fully nonlinear robust MPC schemes.

Another alternative to the approach taken by Tube-MPC is the utilization of stochastic models to account for the inaccuracies and disturbances [73, 127]. This results in an enlarged design space, as it is possible to account for varying levels of probabilities and correlations of uncertainties. However, the downside of this flexibility is the increased complexity of the numerical techniques and computational complexity. In addition, it comes with a greater challenge in parameterizing those uncertainties. This thesis is therefore focused on the application of deterministic methods and will not consider stochastic methods.

The following paragraph will introduce the notation of the Tube-MPC scheme (based on the work by [117]) applied within this thesis. The system under control can be written as

$$x(k+1) = Ax(k) + Bu(k) + Ed(k), \tag{2.15}$$

and the constrained linear MPC control problem could be formulated as follows

$$\begin{aligned}
\min_{\bar{u}} \sum_{i=0}^{N-1} \left(x_i^T Q x_i + u_i^T R u_i\right) + x_N P x_N, & \qquad (2.16a)\\
\text{s.t.} \quad x_0 = x(k), & \qquad (2.16b)\\
x_{i+1} = Ax_i + Bu_i, \quad \forall i \in [0, N-1], & \qquad (2.16c)\\
H_{x,i} x_i \leq g_x, \quad \forall i \in [0, N-1], & \qquad (2.16d)\\
H_{u,i} u_i \leq g_u, \quad \forall i \in [0, N-1], & \qquad (2.16e)\\
H_{x,\mathrm{T}} x_i \leq g_{x,\mathrm{T}}, & \qquad (2.16f)
\end{aligned}$$

where $H_{x,i}$, $H_{x,i}$ and $H_{x,\mathrm{T}}$ denote the linear inequality matrices and g_x, g_u and $g_{x,\mathrm{T}}$ the corresponding upper bounds. Using a pre-stabilizing control law $u = -Kx + v$, the reachable sets $\mathbb{X}_{\mathrm{R},i}$ (for time instant i) of the closed-loop dynamics of (2.15) for a bounded disturbance d can be calculated using ellipsoids:

$$E(p, M) := \left\{ x \in \mathbb{R}^n | (x-p)^T M^{-1} (x-p) \leq 1 \right\}, \tag{2.17}$$

with the center p and M the corresponding shape matrix. Using the over-approximation by [128], the Minkowski sum of two ellipsoids can be written as

$$\begin{aligned} &E(p_1, M_1) \oplus E(p_2, M_2) \\ \subset &E\left(p_1 + p_2, \left(1 + c^{-1}\right) M_1 + (1+c) M_2\right), \end{aligned} \tag{2.18}$$

with $c = \sqrt{\mathrm{Tr}(M_1)/\mathrm{Tr}(M_2)}$ and $\mathrm{Tr}(M)$ the trace of the Matrix M. Together with the following law for the affine transformation

$$A \cdot E(p, M) + b = E(Ap + b, AMA^T), \tag{2.19}$$

all mathematical tools required to compute the reachable sets $\mathbb{X}_{\mathrm{R},i}$ via consecutive application of the set arithmetics to the system dynamics (2.15) are available. This allows to tighten the constraints of (2.16) such that $\mathbb{X}_{\mathrm{R},i} \subset \mathbb{X}_i$ holds for all admissable nominal predictions $x_i \subset \bar{\mathbb{X}}_i$ using the approach from [125, 129]. Finally, the inequalities can be written as

$$H_{x,i,k} p_i + \sqrt{H_{x,i,k} M_i H_{x,i,k}^{\mathrm{T}}} \leq g_{x,k}, \tag{2.20}$$

where the index k denotes the corresponding rows and vector elements of $H_{x,i}$ and g_x. The reformulation for the input and terminal set inequalities can be done accordingly.

These modifications allow to formulate a new MPC problem, using the nominal inputs v as optimization variables and the nominal state predictions p for the dynamics propagation. The tightened constraints guarantee that this modified problem fulfills the constraints for all potential disturbances d in the original problem:

$$\min_{\bar{v}} \sum_{i=0}^{N-1} \left(p_i^T Q p_i + v_i^T R v_i \right) + p_N P p_N, \tag{2.21a}$$

$$\text{s.t.} \quad p_0 = x(k), \tag{2.21b}$$

$$p_{i+1} = A p_i + B v_i, \quad \forall i \in [0, N-1], \tag{2.21c}$$

$$H_{x,i} p_i + \sqrt{H_{x,i,k} M_i H_{x,i,k}^{\mathrm{T}}} \leq g_x, \quad \forall i \in [0, N-1], \tag{2.21d}$$

$$H_{u,i} v_i + \sqrt{H_{u,i,k} K M_i K^{\mathrm{T}} H_{u,i,k}^{\mathrm{T}}} \leq g_u, \quad \forall i \in [0, N-1], \tag{2.21e}$$

$$H_{x,\mathrm{T}} p_i + \sqrt{H_{x,\mathrm{T},k} M_N H_{x,\mathrm{T},k}^{\mathrm{T}}} \leq g_{x,\mathrm{T}}. \tag{2.21f}$$

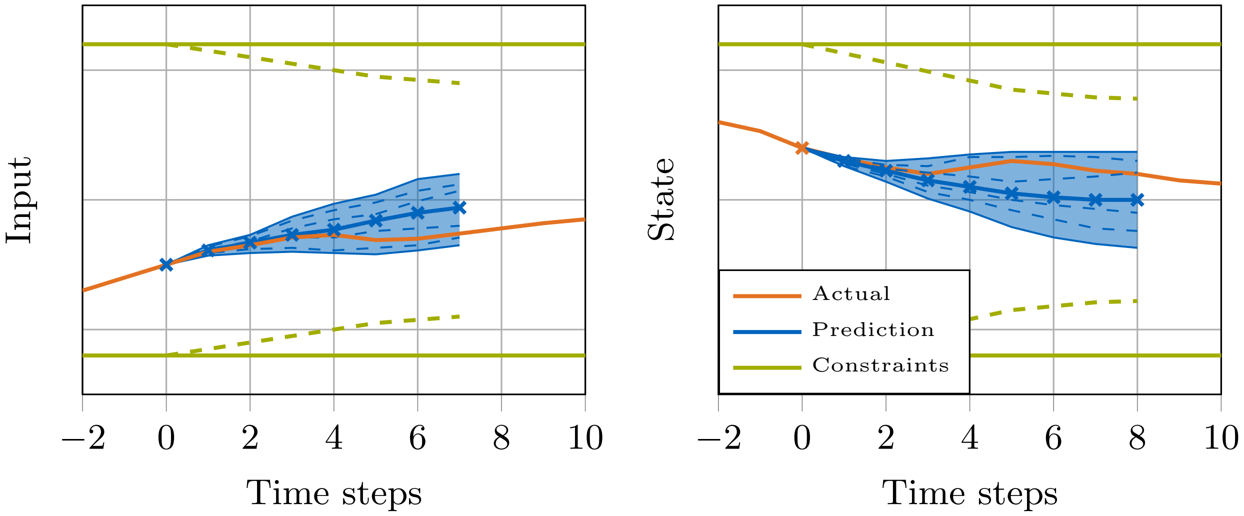


Figure 8: Visualization of the model predictive control concept for the uncertain case. The blue predictions visualize the optimization variables and predictions of the system behavior at time instant 0. The shaded areas depict the set-based prediction of the potential dynamic behaviors. The orange graph depicts the actual behavior of the system observed after repeated application of the control inputs.

The intuition behind the above reformulation is depicted in Figure 8. The prediction of the nominal behavior (solid lines) follows the same dynamics as in the standard MPC case. In addition, the Tube-MPC framework assumes that bounded disturbances act in the future which are unknown at the current time instant. These could potentially lead to the dashed outcomes for the actual system behavior. The Tube-MPC framework calculates upper and lower bounds for the behavior with respect to a certain state x and can therefore enhance the nominal prediction with a robust prediction of the future system behavior. The impact of these bounds on the optimization problem can be incorporated via the tightening of the constraints (dashed green lines) into the nominal optimization problem. Mathematically speaking, this tightening reflects the impact of the square root terms in the above optimization problem. The key advantage of this method is that it does not impact the computational requirements of the optimization problem. The only additional overhead is the calculation of the reachable sets prior to the formulation of the optimization problem, however, the impact can be neglected compared to the optimization when using the presented ellipsoidal arithmetic.

2.3 Linear Regression

Linear regression aims at the identification of a model linear in its parameters to approximate an unknown relationship between different variables [130]. These variables are split into input variables x and output variables y. Even though these symbols have been used for different purposes in the previous sections, they are re-used in this section to keep consistency with the standard literature on the topic. The following section introduces the basics of the two most common linear regression techniques: Classical basis function regression using a quadratic loss function and GPR. The former belongs to the class of parametric models, the latter is a non-parametric regression technique. However, it should only serve as a brief introduction into the topic and its notation. For a thorough and detailed explanation of the topic, the reader is referred to the literature [130, 131, 132].

2.3.1 Parametric Models

In the parametric model case, the relation between input and output variables is modeled via a finite set of basis functions $\phi(x)$. It can be written as

$$y = \phi(x)^{\mathrm{T}} w + \epsilon, \tag{2.22}$$

with the basis function weights w and the remaining model error ϵ. This model error can either be handled in a deterministic least squares sense or in a probabilistic framework. Both approaches lead to similar results. In the next steps, the weights which approximates the data best in the sense of a quadratic loss function

$$L(x, y, w) = \sum_{i=0}^{N} \left(y_i - \phi(x_i)^{\mathrm{T}} w \right)^2, \tag{2.23}$$

can be identified. In this equation, i denotes the current data point and N is the set of the training data set. The main advantage of the quadratic loss function lies within the availability of an analytic solution for the optimal weights w_{o}. This can be calculated via the pseudo-inverse

$$w_{\mathrm{o}} = \left(X^{\mathrm{T}} X \right)^{-1} X^{\mathrm{T}} y \tag{2.24}$$

of the design matrix

$$X = \begin{bmatrix} \phi(x_0)^{\mathrm{T}} \\ \vdots \\ \phi(x_N)^{\mathrm{T}} \end{bmatrix}. \tag{2.25}$$

More complex cases with constraints on the parameters or the outcome of the approximated function as well as non-quadratic loss functions can be handled via the application of standard nonlinear optimization techniques. Common loss functions include the absolute error function, also referred to as L1-norm regression [130].

Parametric regression is usually a good choice when the modeled relation is based upon some known characteristics which can be used to design a set of meaningful basis functions $\phi(x)$. In this case, the model tends to be low dimensional and the inverse of $X^{\mathrm{T}}X$ can be computed with low computational effort. In addition, these computations can be formulated in a recursive way [130] which makes them especially suited for embedded systems which operate on long time-frames with restricted hardware. These formulations show close connections to the Kalman Filter and are therefore well-known in the control engineering community.

2.3.2 Non-parametric Models

In contrast to parametric regression, non-parametric models do not require the a-priori specification of basis functions. The most prominent member of this type is GPR. Instead of basis functions, it uses a similarity measure, the kernel function $k(x, x)$, to construct the prediction. Data points which are close to each other in the sense of the kernel function are assumed to have similar output values y. The common derivation of this approach [130] is done in a stochastic way via the joint normal probability function of training data y and predictions y_*

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim \mathcal{N}\left(m(x), \begin{bmatrix} k(x,x) + \sigma_{\mathrm{M}}^2 I & k(x, x_*) \\ k(x_*, x) & k(x_*, x_*) \end{bmatrix} \right), \tag{2.26}$$

with the mean value prior $m(x)$, the measurement uncertainty σ_{M}, and I being the identity matrix. The posterior prediction mean $\mu(x_*)$ and covariance $\sigma(x_*)$ can then be written as

$$\mu(x_*) = m(x) + k(x_*, x) \left(k(x,x) + \sigma_{\mathrm{M}} I \right)^{-1} \left(y - m(x) \right), \tag{2.27a}$$

$$\sigma(x_*) = k(x_*, x_*) - k(x_*, x) \left(k(x,x) + \sigma_{\mathrm{M}} I \right)^{-1} k(x, x_*). \tag{2.27b}$$

One of the key motivations to utilize GPR in the literature is the ability to infer upper and lower confidence bounds from the posterior covariance function [125]. However, it should be noted that the reliability of these bounds heavily depends on an accurate calibration of the measurement uncertainty σ_{M} and the kernel parameters. As the online adaption of these parameters is computationally heavy, they are usually determined a-priori to the systems operation using hyperparameter optimization [130]. As (2.27b) is not dependent on the actual data and only on the number of data points and their spread, it becomes obvious that the posterior covariance is closer to a measure for the coverage of the data points within the feature space rather than a true reflection of uncertainty within the trained model.

The identification of a GPR model is usually more expensive from a computational point of view compared to parametric regression. The matrix inversion in (2.27a) and (2.27b) scales with the number of data points instead of the number of basis functions as in (2.24). As a result, non-parametric regression is usually applied to problems where either little knowledge about the shape of the estimated function is available or the data is distributed in a high-dimensional feature space where it becomes hard to define reasonable basis functions.

2.4 Types of Uncertainty

The concept of uncertainty is used within various science and engineering disciplines. This wide-spread adoption has led to a variety of definitions and categorizations, from where we will present the most relevant for the upcoming work in this section.

Starting from a control engineering perspective, the main task of a feedback controller is to minimize the influence of uncertainties on the performance of a technical system. This minimization is mathematically formulated with two basic requirements [133]: First, the closed-loop control system has to be stable. Second, the closed-loop system has to limit the maximum tracking error of the target value under control within a certain range, which is derived from the operational circumstances of the system. The thorough theoretical analysis of these requirements has led to a more granular classification of uncertainties [133]: The first category are *external disturbances*. Their key characteristic is, that their timely evolution does not depend or correlates with the state or inputs of the system under control. In the case of autonomous vehicle motion control, these are for example wind or inclination. The second category are *parametric uncertainties* of the system under control. Within the autonomous vehicle control domain, these could be a tire model parameter mismatch. The third category are *dynamic uncertainties* of the system under control, e.g. unmodelled actuator dynamics. There are various ways to represent those uncertainties and judge their impact on the desired closed-loop properties. Details on these categories and consequences for the control system design are discussed in the textbooks [133, 134, 135].

The categories above have been designed to group uncertainties according to their impact on stability and performance analysis of the closed-loop control systems. However, there is another way to look at uncertainty when introducing data-driven techniques: They can be separated into aleatoric and epistemic uncertainties [136]. Aleatoric uncertainty describes uncertainties which are inherently random and can not be explained via the model. In contrast, epistemic uncertainties stem from a lack of knowledge, which can be decreased via the consideration of more data or information. Following this characterization, it becomes clear that there are uncertainties which can be learned via data-driven techniques (epistemic uncertainties) and others which can not be learned (aleatoric uncertainties). Even though they characterize similar phenomenons, these categories are different from the control engineering definitions. While there is a tendency that epistemic uncertainties uncertainties are also parametric or dynamic uncertainties, there are counter-examples to this: There are external disturbances, e.g. the external torque impacting the dynamic behavior of a electrical machine, which can be identified when there is additional information about the overall system which is powered by the machine. It is therefore also considered an epistemic uncertainty.

Taking a closer look at the aleatoric and epistemic uncertainty definitions, they show ambiguity and are less precise than the control engineering way to look at it presented above. One could argue, that in a world where deterministic physical laws govern the timely evolution *every uncertainty* is epistemic uncertainty as it could be incorporated in the model by thorough analysis, experimentation and validation. However, engineering applications usually seek the trade-off between modeling accuracy, effort to build the model and robustness in the application. Therefore, the classification as aleatoric and epistemic unertainty is inherently dependent on the modeling process [137]. A good example for this is the design of chassis and suspension components: While this design requires careful analysis via large-scale FEM simulations, this modeling detail is by no means required for the design of an autonomous vehicle motion control algorithm. In the former case, the material parameters would be considered epistemic uncertainties as they could be identified with careful material testing and validation of the FEM simulation. In the latter case, the impact of chassis and suspension stiffness is usually considered a neglectable uncertainty and not dealt with explicitly. However, they still impact the overall model behavior and therefore belong to the class of aleatoric uncertainties. It is therefore concluded, that the classification of uncertainty as epistemic comes with a requirement to be able to identify the uncertain aspects with the data available and reasonable effort.

2.5 Software Architecture

A central aspect of the work in this thesis has been the embodiment of the proposed algorithms into a full software stack and thorough investigation of the performance in non-idealized scenarios. The information in this section cover the software architecture for the IAC software, as the software stack for the Roborace competition was restricted to planning and control algorithms. The overall architecture of this stack is depicted in Figure 9 and follows a similar structure as many others [3, 21]. More information on the various software modules are available in [18].

The stack is split into three main areas: The perception part handles the localization and object detection tasks based on the environment perception sensors (LIDAR, Radar, Cameras and GPS). It applies independent detection pipelines for each sensor modality. The tracking algorithm matches the detected objects to the already tracked objects and creates new tracked objects when required. The prediction module forecasts the most likely behavior of other vehicles on track, while taking into account the interactions between those vehicles. The trajectory planning part is responsible for the decision making and coarse

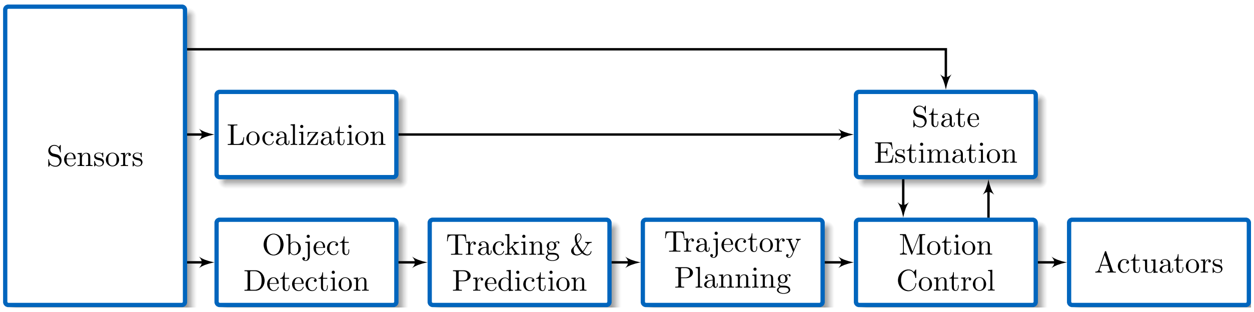


Figure 9: Software structure of the TUM Autonomous Motorsport stack

trajectory planning. Finally, the state estimation and motion control modules refine the planned trajectories and compute a smooth target steering angle, a target throttle position and a target brake pressure which is sent the low-level vehicle controller.

All of the software modules run on a single computer and communicate via the open-source middleware ROS2. The operating system is a standard Ubuntu 20.04, however, the system configuration has been optimized to improve its real-time behavior. The average update rates of the perception and planning software modules are designed to be 5 Hz to 10 Hz. All of these modules run in an asynchronous fashion and account for data transport delay and algorithm execution times. This enables a heavily concurrent execution and high utilization of all CPU-cores. The control and state estimation module are executed with a fixed update rate of 100 Hz.

It is important to point out, that all of the modules consistently adjust the vehicle behavior to whats happening around it based on new information coming from the diverse sensor setup. The positions and velocities of the tracked objects are consistently corrected based on the outputs of the different object detection pipelines. This creates the basis for accurate predictions of how vehicles in the considered scene are going to behave in the next couple of seconds. The planning takes into account this prediction to generate an optimal target trajectory for the ego vehicle. However, this plan has to be updated frequently to incorporate novel knowledge about the behavior of other vehicles. It should be noted, that this behavior is significantly different to situations where only static obstacles and the track bounds have to be considered as its the case in single-vehicle racing. Those static scenarios lead to smaller deviations of the solution between consecutive trajectory planning steps and are therefore also easier to handle on the motion control side. The motion control finally takes responsibility for handling and attenuating the impact of uncertainties and disturbances on the vehicle motion to robustly execute this target behavior.

3 Robust Control in Autonomous Racing

3.1 Vehicle dynamics state estimation and localization for high performance race cars

This paper [8] (full text in Section 7.1) explains the fundamentals behind the state estimation strategy used within the software architecture outlined in Section 2.5 and applied during all subsequent projects discussed in this thesis. Its scope is three-fold: First, it provides a holistic framework to fuse several sensor information available at different rates and even potentially asynchronous measurements. Second, it provides reliable motion state estimation in case of single sensor failures. Third, the obtained position and motion states are robust against measurement errors and timing inaccuracies as they are combined with a variety of information.

This target is achieved by the utilization of an easily configurable algorithm based on the Kalman Filter equations. It is updated in each execution step to incorporate only available sensor data via adaption of the output matrix and utilizes a high base frequency to minimize the effect of jitter of the incoming sensor signals. In addition, a vehicle motion dependent configuration of the uncertainty assumptions is proposed as it is common that the motion in the longitudinal direction is influenced much more via timing uncertainties within the sensor data path. Furthermore, the paper demonstrates that a rather simplistic point-mass model can outperform a more sophisticated single track model in terms of estimation bias. It leverages the fact that the data coming from the inertial measurement unit can be used directly as an input for the dynamic model update equations. This strategy renders the (often highly uncertain) tire models unnecessary and can therefore demonstrate uniform state estimation quality for the full range of driving dynamics.

The paper is concluded with the application of the proposed algorithm on the Roborace DevBot plattform with speeds of up to $150\,\mathrm{km\,h^{-1}}$ and two localization pipelines based on Global Navigation Satellite System (GNSS) and LIDAR localization. It should be noted, that this algorithm has been used with only minor modifications during all the subsequent publications and has therefore delivered its promise of universal applicability.

3.2 Tube Model Predictive Control for an Autonomous Racecar

This paper [14] (full text in Section 7.2) introduces the fundamentals of the Tube-MPC scheme used within this and the following publication. It has been developed to overcome the limitations of classical feedback control for autonomous driving applications by improving the control performance with knowledge about the applicable vehicle dynamics constraints and the upcoming target trajectory for the vehicle. At the same time, its formulation as a robust control problem helps to mitigate well-known difficulties with model predictive control such as infeasibility of the constrained optimization problem due to the influence of external disturbances.

The paper proposes an MPC scheme based on a friction limited point-mass model in conjunction with a set of low-level acceleration feedback controllers. This split creates a low-dimensional state-space model with the state-space variables lateral deviation, the derivative of the lateral deviation and the deviation from the target velocity. The inputs are assumed to be the accelerations with respect to those error coordinates. The resulting optimal control problem is linearized and solved within an real-time SQP fashion with the ADMM-based solver OSQP. The modifications to achieve robustness are done in the style of [117] via a pre-stabilizing control law and constraint tightening based on the construction of reachable sets with the pre-stabilizing controller. The uncertainties are specified as external accelerations acting upon the closed-loop system.

The paper is completed with an extensive study of the influence of the pre-stabilizing controller as well as the uncertainty assumptions on the closed-loop performance. While the former does not tend to show a significant impact for reasonably chosen control parameters, the latter proves to be a theoretically justified tuning parameter with explainable influence on the robustness of the overall system. The resulting Tube-MPC outperforms a nominal MPC scheme as well as an LQR controller (presented in [1]) significantly in terms of constraint satisfaction at a given lap time. The results are obtained on a Hardware-in-the-Loop simulator utilizing a sophisticated nonlinear dual track model, actuator and sensor models as a replacement for the vehicle behavior. It has shown to be real-time capable on a rapid prototyping ECU from Speedgoat.

3.3 A Tube-MPC Approach to Autonomous Multi-Vehicle Oval Racing

This paper [16] (full text in Section 7.3) builds upon the concept proposed in the previous publication. It adds details on the low-level acceleration controllers and the hardware interfaces to run the controller on the Indy Autonomous Challenge AV-21 platform at oval racetracks in Indianapolis and Las Vegas. In addition, it proposes a reformulation of the cost function to enable simultaneous feedback control and fine-optimization of the target trajectory. This enables the graph-based trajectory planning algorithm to operate on a coarser grid and therefore to achieve faster update rates even in complex multi-vehicle scenarios.

The low-level acceleration controllers are split into a longitudinal and a lateral control path. The former is a simple P-controller in conjunction with an inverse powertrain and brake system map to calculate appropriate throttle and brake values. The latter is a PI-like controller with a nonlinear anti-windup strategy. In addition, the concept employs a classical PI-feedback loop for the tracking of the target steering angle to mitigate steady-state error on the steering actuator level. The banking of up to 20° on the oval racetracks require a suitable compensation of the effects on the measured accelerations. This is handled via a transformation of the measured accelerations prior to feeding them into the low-level acceleration controllers, which leads to a computational efficient strategy in contrast to extending the MPC optimization problem with the nonlinear influence of the banking angle.

The second part of the paper is a simulation comparison between the proposed robust control approach based on the limited friction point-mass model and an MPC with a nonlinear tire model. It is found, that the simplistic model in conjunction with the robustness extensions achieves better performance than the nominal MPC with the nonlinear tire model. This is attributed to the model uncertainties which are inevitable in the nonlinear MPC setting and are handled more structured in the Tube-MPC. In addition, the influence of the length of the optimization horizon, the tuning of the low-level controller as well as the performance for a-priori unknown trajectory changes are analyzed.

The paper concludes with the presentation of experimental data on the AV-21 in single-vehicle and two-vehicle scenarios. The latter is challenging for the controller, as the trajectory planner updates frequently to adjust the driving behavior towards the opponent. However, the concept handles these dynamic changes in the target trajectory smoothly during real-world tests at the Las Vegas Motor Speedway at speeds up to $265\,\mathrm{km\,h^{-1}}$ and accelerations of $21\,\mathrm{m\,s^{-2}}$.

4 Data-Driven Methods in Autonomous Racing

4.1 A Model-Free Algorithm to Safely Approach the Handling Limit of an Autonomous Racecar

This paper [10] (full text in Section 7.4) is motivated by the relation between the physical tire-road friction potential and the ability of the control system to handle it. While the necessity of a tire-road friction estimate is motivated in many papers, the required safety margin for the tire-road friction value used for trajectory planning to allow the control system to operate safely is often not included in the studies. This can lead to overly optimistic estimates and unsafe behavior when operating on the racetrack. This publication proposes an estimation strategy for the maximum *utilizable* tire-road friction coefficient based on safety metrics such as tire slip and lateral control error.

This estimation strategy is realized via a safe Bayesian Optimization algorithm based on GPR. Three safety metrics are evaluated on a per-lap basis: First, the maximum longitudinal wheel slip as a measure for the tire utilization in braking and acceleration scenarios. Second, the difference between the front and the rear side slip angle as a measure for the tire utilization in cornering scenarios. Third, the lateral control error as a general quality measure of the control performance. The algorithm requires the specification of an upper bound for each of those metrics which is deemed the safety constraint. The actual behavior of these metrics is then identified via a single Gaussian process for each of them and updated after the observation of the metric for each lap. Based on those safety constraints, the algorithm chooses a tire-road friction coefficient used for planning of the velocity profile for the next lap. The possibility to model the exploration uncertainty within the GPR framework allows the vehicle to start exploring the maximum utilizable tire-road friction coefficient in a cautious manner without violating the safety metrics.

The paper demonstrates that the proposed framework is capable of handling various vehicle conditions and adjust the maximum accelerations used by the trajectory planner to the current closed-loop performance of the autonomous driving software. This is shown via a simulation study with varying brake pad friction coefficients, a parameter which is usually unknown and can only estimated on empiric data in real-world applications. Furthermore, it often varies over time with wear and operating conditions such as temperature and is therefore hard to estimate reliably. Finally, the system is applied on the full-scale Roborace prototype DevBot 2.0 and demonstrates its capabilities under real-world conditions. These experiements are conducted with the trajectory planning proposed by [4] and the LQR trajectory tracking controller proposed in [1].

4.2 Real-Time Learning of Non-Gaussian Uncertainty Models for Autonomous Racing

This paper [16] (full text in Section 7.5) proposes a real-time learning algorithm to identify a mean prediction and reliable uncertainty estimates for a static relation between input and output variables in a standard linear regression setup. It can be applied to improve static feedforward control laws or other static linear regression models during runtime. The algorithm is based on a combination of a recursive least squares algorithm and a batch version of the recursive quantile estimator presented in [138]. In contrast to other algorithms (such as GPR) utilized for online improvements in the control community, it splits the handling of epistemic (via the linear regression) and aleatoric uncertainty (via the quantile estimator) in distinct steps. This allows to deliver reliable uncertainty estimates for arbitrary (e.g. non-gaussian) uncertainty distributions in a computational efficient way.

The real-time learning algorithm is realized via a two-stage recursive algorithm: The first stage is based on a recursive least squares approach and allows to identify a feature-dependent mean value model to capture the general trend of the data. It is realized via a standard gradient-descent scheme and does therefore not rely on the subsequent calculation of a covariance matrix. This saves significant computational resources for models with a larger number of features at the price of a non-optimal convergence behavior. The second stage utilizes a recursive version of a sliding window approach for the estimation of an upper or lower quantile. Its is realized via a nonlinear filter with sign dependent gains. The paper includes a convergence analysis to a robust positive invariant set of weights and (as a consequence) the convergence of the quantile estimators.

The proposed algorithm is evaluated in comparison to a Bayesian linear regression and a GPR. It can be shown that all approaches deliver reliable uncertainty estimates for Gaussian model deviations. However, only the proposed approach algorithm delivers reliable estimates models with non-gaussian uncertainties. As this is the standard case in real-world applications, the proposed method is superior to the standard approaches when reliability of the uncertainty estimates is key such as in safety critical applications. In addition, the algorithm is benchmarked on data collected with the Roborace DevBot 2.0 to learn a residual model for the lateral feed-forward control law and is shown to outperform its competitors. This reinforces the findings on artificial test datasets and makes the algorithm suitable for application of real-time critical uncertainty model learning.

4.3 Combination of Robust Control and Data-Driven Methods

Finally, this section will discuss how the robust and data-driven workstreams proposed in this thesis can be combined to achieve increased performance while maintaining safe driving behavior. Even though this section is based on simulation results only, it outlines the potential of the developed algorithms to handle the uncertainty within the vehicle control task efficiently. As these results have not been published elsewhere, this section will present them in a holistic way rather than summarizing the contributions of the corresponding paper as in the previous sections.

The design of the combined robust and data-driven controller is based on a thorough analysis of the uncertainties which should be handled by the different components. They are matched into to the categories used in control engineering and the uncertainty quantification (discussed in Section 2.4) in Figure 10, to build a better understanding of the impact and handling strategies within the control system. While the former (represented by the horizontal ordering in the figure) follows clear technical characteristics, the split into epistemic and aleatoric uncertainties depends on the chosen design assumptions. The following work will consider effects as epistemic, if the sensor setup on the Indy Autonomous Challenge AV-21 is capable of measuring the impact of the uncertainty (depends on the available measurement values, their accuracy and timing resolution) and its impact is reproducible within test sessions with comparable environmental conditions. Starting with the external disturbances, there are three significant effects present in oval racing: track banking, track irregularities and wind. The former is considered to be an epistemic uncertainty, as it is consistent for a track and can be deduced from Inertial Measurement Unit (IMU) data with careful data postprocessing. In contrast, track irregularities (deviations from a banked but flat track

	Epistemic uncertainties	Aleatoric uncertainties
External disturbances	Track banking	Track irregularities Wind
Parametric uncertainty	Tire parameters Vehicle mass & inertia	Chassis compliance Vibrations
Dynamic uncertainty	Steering actuator dynamics Engine dynamics Brake system dynamics	Tire patch response Steering rack backlash Setpoint latency

Figure 10: Categorization of the uncertainties present in autonomous racing

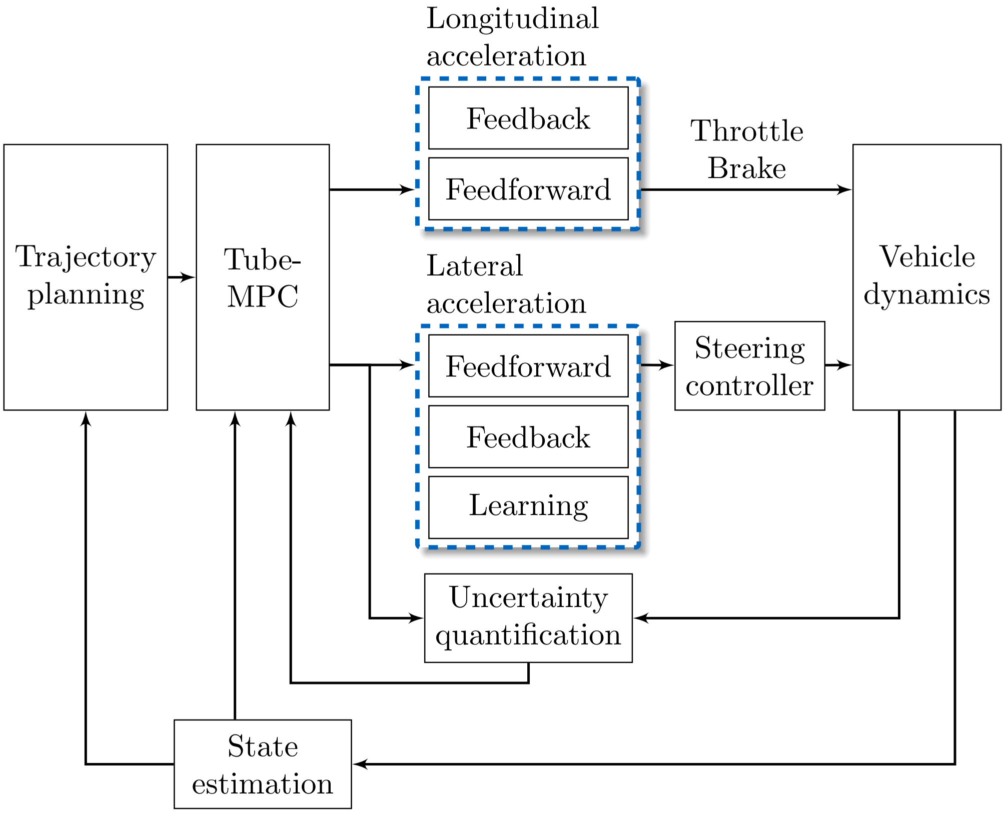


Figure 11: Structure of data-driven Tube-MPC controller

surface) mainly impact the high frequency response of suspension and chassis components. The AV-21 does not provide the required measurement values (e.g. damper potentiometers) at the required rate and accuracy and therefore they are considered aleatoric. The impact of wind on the vehicle dynamics is profound at speeds above $200\,\mathrm{km\,h^{-1}}$, however, even with perfect measurements it would not be reproducible. Tire parameters and vehicle mass and inertia are some of the most influential parameters on the overall vehicle dynamics. However, they are also subject to significant uncertainty or slow change (depending on fuel consumption and vehicle setup). We consider them epistemic parametric uncertainties. In contrast, chassis compliance and vibration characteristics show similar properties as track irregularities and are therefore considered aleatoric parametric uncertainties. Finally, the response of the actuators is considered to be sufficiently measurable via careful comparison of the setpoints and the acceleration response and is therefore epistemic dynamic uncertainty. The remaining dynamic uncertainties, such as tire track contact patch response, steering rack backlash or data transport latencies are considered to be either non-reproducible or too small to be measured independently from each other.

Based on this analysis, we propose the overall control structure in Figure 11 and match the responsibilities of the different components to handling specific uncertainty characteristics.

The key components of the motion control system are the Tube-MPC in conjunction with the low-level acceleration controllers (presented in Section 3.2) and the model-based uncertainty learning framework proposed in Section 4.2. The learning scheme improves a feed-forward control law with a static characteristic continuously based on the behavior observed from the vehicle sensors, however, it does not have access to the track location. It is therefore dedicated to handle the epistemic parametric uncertainties. The aleatoric uncertainties are jointly handled by the acceleration controllers. Even though they might not be identifiable from the sensor data, they can be successfully suppressed in case their influence impacts the overall vehicle dynamics by the low-level feedback loop. The remaining epistemic uncertainties are not considered explicitly in the controller design, however, they tend to be constant for a track and vehicle pairing. We therefore utilize the uncertainty quantification from Section 4.2 to measure this mismatch and adjust the amount of caution employed by the Tube-MPC accordingly. The overall system for the upcoming simulation experiments is completed by the state estimation presented in Section 3.1 and the trajectory planning consisting of a simple raceline replay planning, which outputs a pre-computed target raceline for easy evaluation of different controller designs.

The feedfoward learning employs a recursive least squares algorithm (Section 4.2) to achieve computational efficient improvements upon the basic feedfoward control law utilized in the standard version of the Tube-MPC. It utilizes the following model

$$\Delta\delta = w_0 + w_1 a_{y,\mathrm{T}}^3 \tag{4.1}$$

to predict the steering offset $\Delta\delta$ from the neutral-steer feedforward required to achieve the desired lateral target acceleration. It consists of two parts: A constant offset w_0 to compensate for miscalibrations in the steering actuator or suspension and a polynomial term $w_1 a_{y,\mathrm{T}}^3$ to compensate additional understeer introduced from the overall vehicle design, the suspension setup and the aerodynamics. The training samples $\Delta\delta$ are computed from the corrective steering angles applied by the lateral acceleration feedback controller. The model will therefore store the required feedback steering angles to improve the transient response for the next time when a similar operating point is requested. In addition, the features of the regression model are normalized such that both are weighted equally in the learning process.

The second part of the employed learning framework is the recursive quantification of the remaining uncertainty in the lateral acceleration low-level control loop (see section 4.2 for details). The uncertainty is calculated as the deviation of the target and the actual lateral acceleration, as this matches the Tube-MPC assumption about the input uncertainty. The quantile estimator constantly monitors this deviation, normalizes it with respect to the current acceleration limit and estimates the corresponding 90 %-quantile. The result is used to adapt the disturbance estimates used for the calculation of reachable sets within the

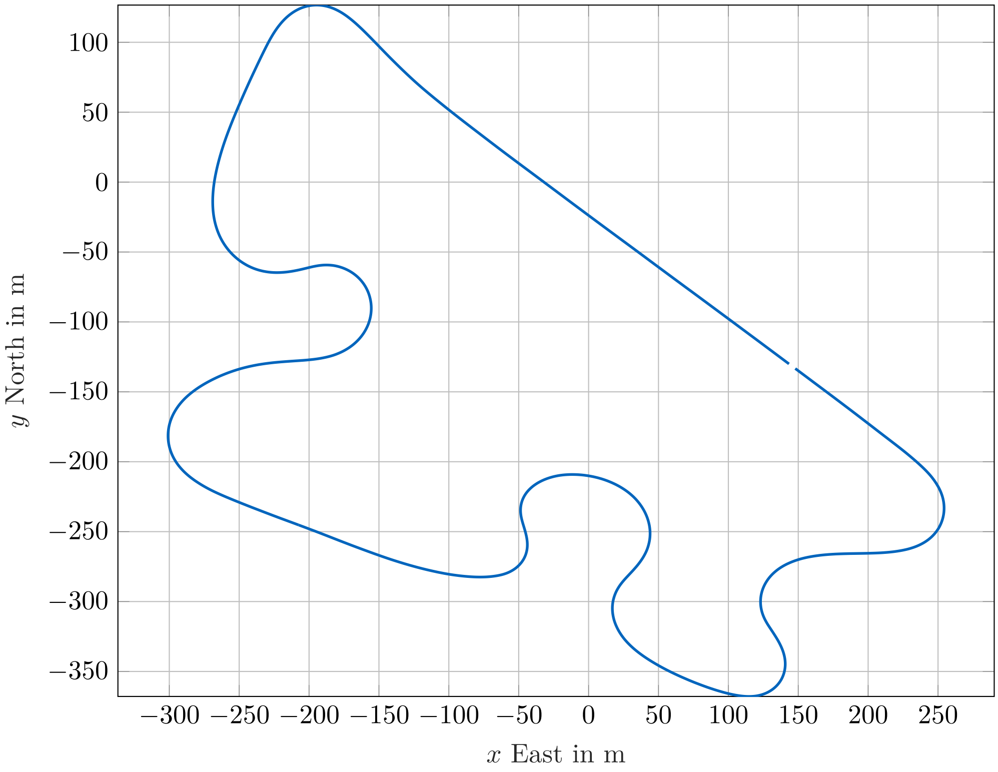


Figure 12: Layout of the Modena racing circuit

Tube-MPC algorithm. As a result, the Tube-MPC is more cautious when the uncertainty is estimated to be large and more aggressive when it is estimated to be small. It can therefore adjust its behavior to changing external conditions as well as to the improvements made by the feedforward learning.

The evaluation scenario is the Modena racing circuit (see Figure 12), a road course in Italy, because of its more complex structure with left and right corners at different speeds in contrast to oval racetracks. The vehicle has been parameterized similarly to the Indy Lights vehicle used in the IAC. In addition, a slight steering offset of 0.01 rad was added to mimic imperfect actuator calibration. Furthermore, the front axle tire-road friction coefficient is decreased by 30 % and increased the rear tire-road friction coefficient by 20 % to mimic that the vehicle exhibits significant understeer. These changes are considered extreme cases and they have been chosen to outline that the learning algorithm is capable to adjust to scenarios where the vehicle configuration significantly deviates from the prior assumptions.

The first experiment is a simulation for 40 laps with target trajectories reflecting competitive speed profiles and a usage of the vehicle dynamics potential of close top 100 %. This setup allows to analyze the transient performance of the proposed learning scheme. The results are depicted in Figure 13. Note, that the focus is on the lateral tracking performance in this comparison and therefore it is always referred to acceleration rather than lateral

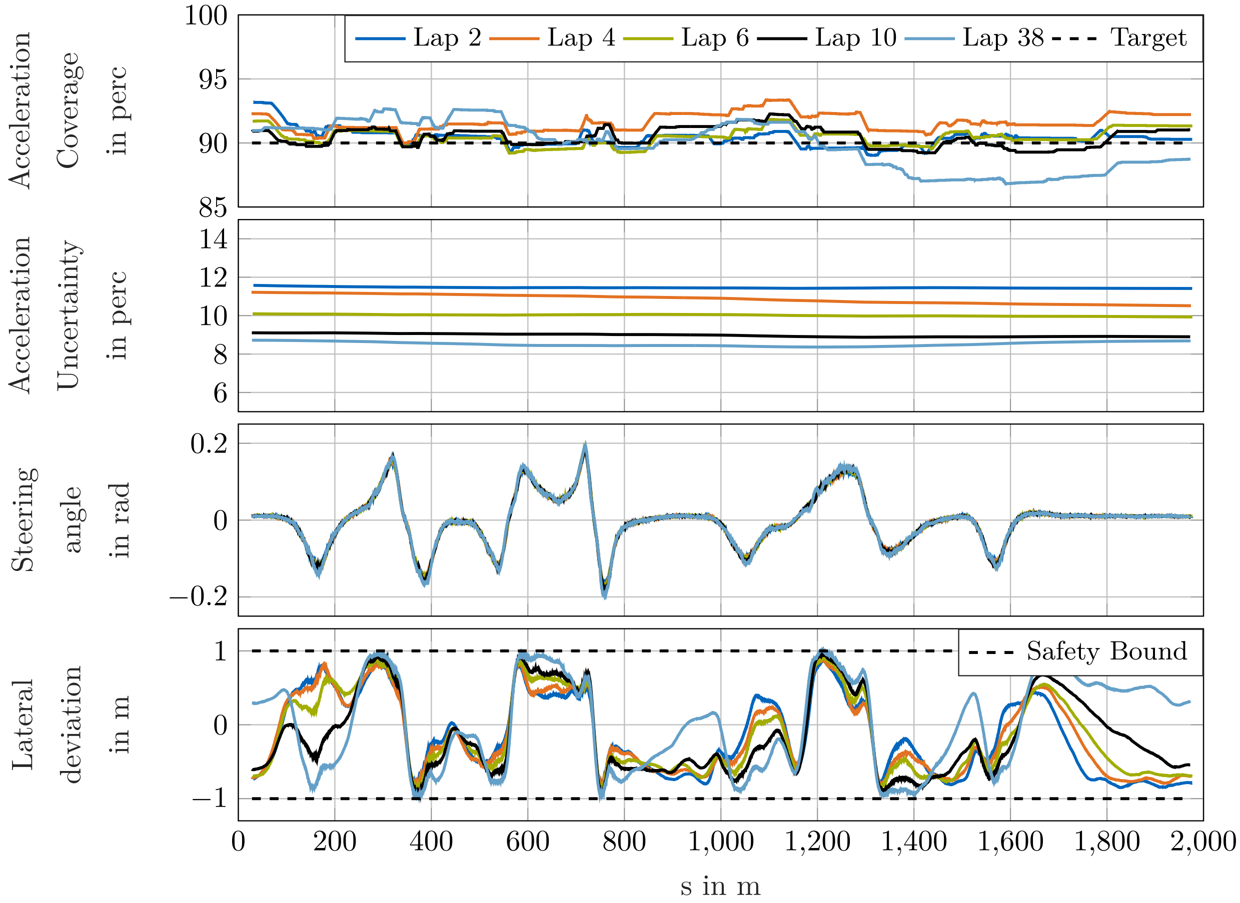


Figure 13: Closed-loop results for various laps with the proposed learning algorithms for the low-level acceleration controller within the Tube-MPC framework

acceleration for the sake of brevity. The first graph showcases the coverage of the current estimated acceleration uncertainty based on a sliding window of 60 s. This estimate is the core driver for the adjustment of the acceleration uncertainty depicted in the second graph. If it is above the target of 90 %, the acceleration uncertainty is lowered and vice-versa (this happens according to the algorithm proposed in Section 4.2). The graphs depict that the learning algorithm is able to decrease the residual uncertainty due to parameter and model mismatch within the low-level acceleration control loop from 11.5 % to 9 %. Consequently, the parallel updating of the uncertainty assumption within the Tube-MPC leads to a bolder driving style and the system exploits the admissable driving tube in form of the lateral path deviation better, as depicted in the bottom graph. This change in performance is complemented with increased steering angles to compensate for the understeer observed at the handling limits. In addition, the laptime is reduced from 77.8 s to 76.5 s due to the decrease in uncertainty and the resulting more aggressive driving style employed by the Tube-MPC. These improvements are significant at the limits of handling and show that the proposed

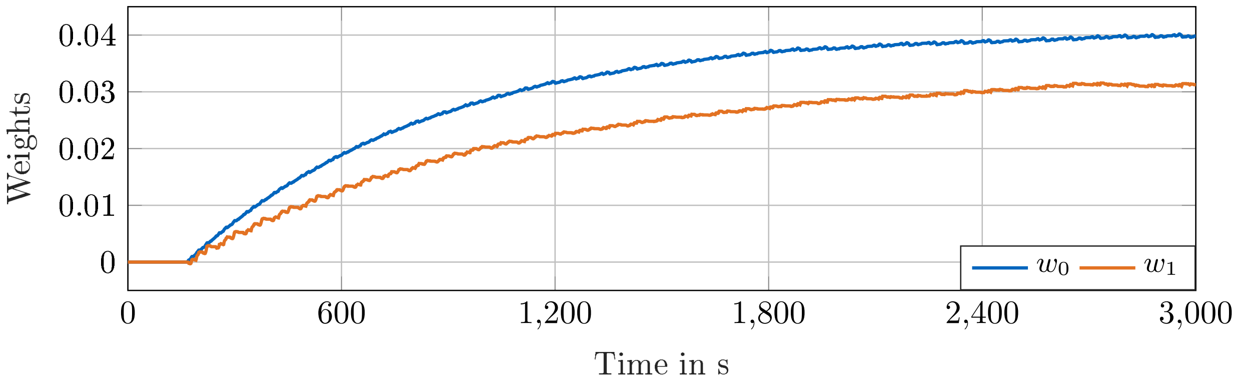


Figure 14: Timely evolution of the learning weights w_0 and w_1

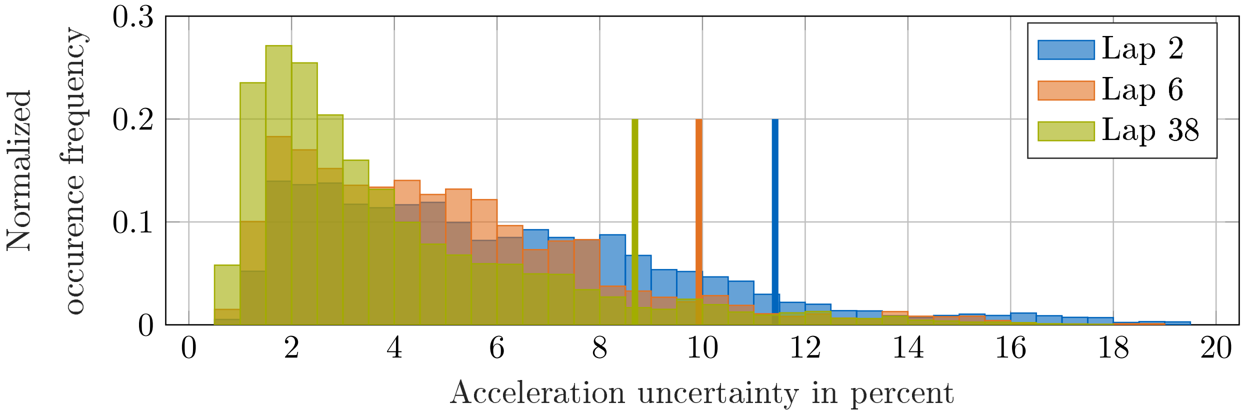


Figure 15: Distribution of the model errors during the learning process. The thick bars mark the estimate of the 90 % quantile for each of the laps.

learning strategy can help to decrease the manual tuning efforts while approaching the handling limit safely. However, the results also show that there is further uncertainty within the closed-loop system which cannot be reduced via the basic feedforward learning applied here. These can be attributed to non-modelled dynamic effects (e.g. steering actuation and yaw dynamics) as well as the simplified choice of the regression model with only two parameters.

Overall, it takes the learning algorithm about ten laps (approximately 800 s) to complete the majority of the learning process, the gains from lap 10 to lap 38 are marginal. The corresponding evolution of the weights is depicted in Figure 14. It is interesting to see, that even though the weights are still changing at this point in time, the improvements in the overall uncertainty of the lateral acceleration control loop are minor. This behavior can be explained via the interplay between the feedforward parts and the feedback parts in the controller. Even though the feedforward law has not fully converged, the remaining uncertainties are small enough to be handled with sufficient performance by the feedback parts. The convergence speed of the algorithm is set to be slow to allow the rest of the control system to keep up with the improvements and to prevent failure due to an overshoot of the learned model weights. This parameter can be tuned via the learning rate of the recursive least squares algorithms, however, it should be treated with caution as this process is carried out on a real vehicle and not in simulation.

Finally, Figure 15 depicts the improvements in the lateral acceleration control performance in a model mismatch histogram. The histogram for lap 2 reflects the situation before the learning process starts (this is triggered in lap 3) and shows that the distribution of the model mismatch is broad and ranges from 2 % to 20 %. The bars depict the 90 % quantiles of the corresponding distributions. With the start of the feedforward learning algorithm, the uncertainty within the lateral acceleration control loop can be reduced until it reaches its minimum. Even though there are still outliers, the majority of the distribution has been shifted significantly to the left which is reflected in the reduction of the 90 % quantile and the improved overall system performance outlined in the previous paragraphs.

5 Discussion

The autonomous driving community has seen a tendency to leverage increasingly complex models in model-based motion control algorithms in recent years. Prominent examples are the utilization of advanced nonlinear tire models for real-time control applications, either in analytical control design approaches [23, 24, 43] or in model predictive control designs [57, 58, 60, 63]. While these works demonstrated significant performance improvements in simulation benchmarks and in real-world scenarios where the model is well-calibrated within the operating region, they come with significant manual engineering and parameter tuning effort for each demonstration run. The various real-world tests conducted with the Roborace DevBot 2.0 and the Indy Autonomous Challenge platform during the work of this thesis have surfaced this fact many times caused by multiple influencing factors: The most prominent example are varying vehicle dynamics due to tire pressure and temperature, which depend both on the outside temperature and the driving behavior. This uncertainty is extended via the impact of the track condition and the resulting tire-road contact behavior which tends to vary between different racetracks and sometimes even within the same track. Other causes for significant uncertainty have been the temperature influence on brake behavior and the impact of the autonomy sensors on aerodynamic behavior.

This thesis investigates whether robust and data-driven control methods are suitable to overcome these challenges and to design motion control systems which are capable to operate safely at the handling limit of the vehicle under the presence of uncertainties and model-mismatch. First, a robust control system design based on a Kalman-Filter for state estimation and a Tube-MPC in conjunction with fast low-level control loops for vehicle motion control have been proposed in Chapter 3. Second, two different strategies leveraging model-free and model-based learning algorithms for adjusting the control system and the vehicle dynamics model have been developed in Chapter 4. The chapter is concluded with the integration of the model learning algorithm with the robust control system proposed previously. The following sections are going to discuss the achievements and shortcomings of the various approaches.

5.1 Robust State Estimation and Motion Control

This thesis presents multiple contributions towards the development of a robust motion control system which explicitly considers uncertainties within the design. It is split into a state estimation algorithm (Section 3.1) based on a Kalman-Filter algorithm and a motion controller based on Tube-MPC (Section 3.2 and Section 3.3). All model-based parts are built on well-known and simple dynamic equations (such as Newtonian mechanics). Effects which are not addressed by the model are tackled via the uncertainty assumptions and the addition of low-level feedback loops. From a broader perspective, this approach can be seen as a strategy to accept that uncertainties are present in the autonomous vehicle motion control problem and classify them as aleatoric, and therefore irreducible, uncertainties instead of treating them as epistemic uncertainties and trying to refine the model to the finest detail. The topics of state estimation and motion control will be discussed individually in the upcoming sections.

5.1.1 State Estimation

The presented state estimator (Section 3.1) based on a point-mass model leveraging basic Newtonian mechanics has shown to have less estimation bias than an alternative implementation using a nonlinear single track model at the expense of increased estimation noise. This trade-off is similar to the well-known bias-variance trade-off [130] in machine learning. It outlines an important challenge with increasingly complex models: Even though they seem to provide higher fidelity and can help to reduce estimation variance, they might introduce significant estimation bias in case of miscalibrated parameters or structural modeling errors. This is especially challenging when the models consider parameters which vary with environmental circumstances such as the tire-road friction coefficients in the nonlinear single track model. Control systems leveraging such biased estimates can not achieve precise tracking of the true vehicle motion states such as speed and acceleration. This can lead to dangerous situations, e.g. an autonomous vehicle approaching a corner too fast and failing to handle it appropriately.

In contrast, the proposed point-mass model is built on Newtonian mechanics and its estimates are unbiased under the assumption that the IMUs are calibrated correctly. However, this approach does not come without disadvantages: First, it requires the vehicle to be equipped with a precisely calibrated (and therefore expensive) high-update rate IMUs for acceleration measurements and reliable localization algorithms. Second, the proposed model assumes that the vehicle motion is planar. Therefore, deriving precise acceleration values from the sensor data has shown to be a challenge during the application of the algorithm on the banked oval tracks at the Indy Autonomous Challenge. The banking led to a mismatch of the planar dynamics used within the model and the actual accelerations observed from the sensor. While this was compensated via an offline generated banking map of the track, it requires cumbersome fine-tuning of the track model to achieve good performance. The impact of banking map errors on state estimation residuals is shown in Figure 16. They serve as a proxy for a ground-truth comparison as the used high-quality GNSS system is

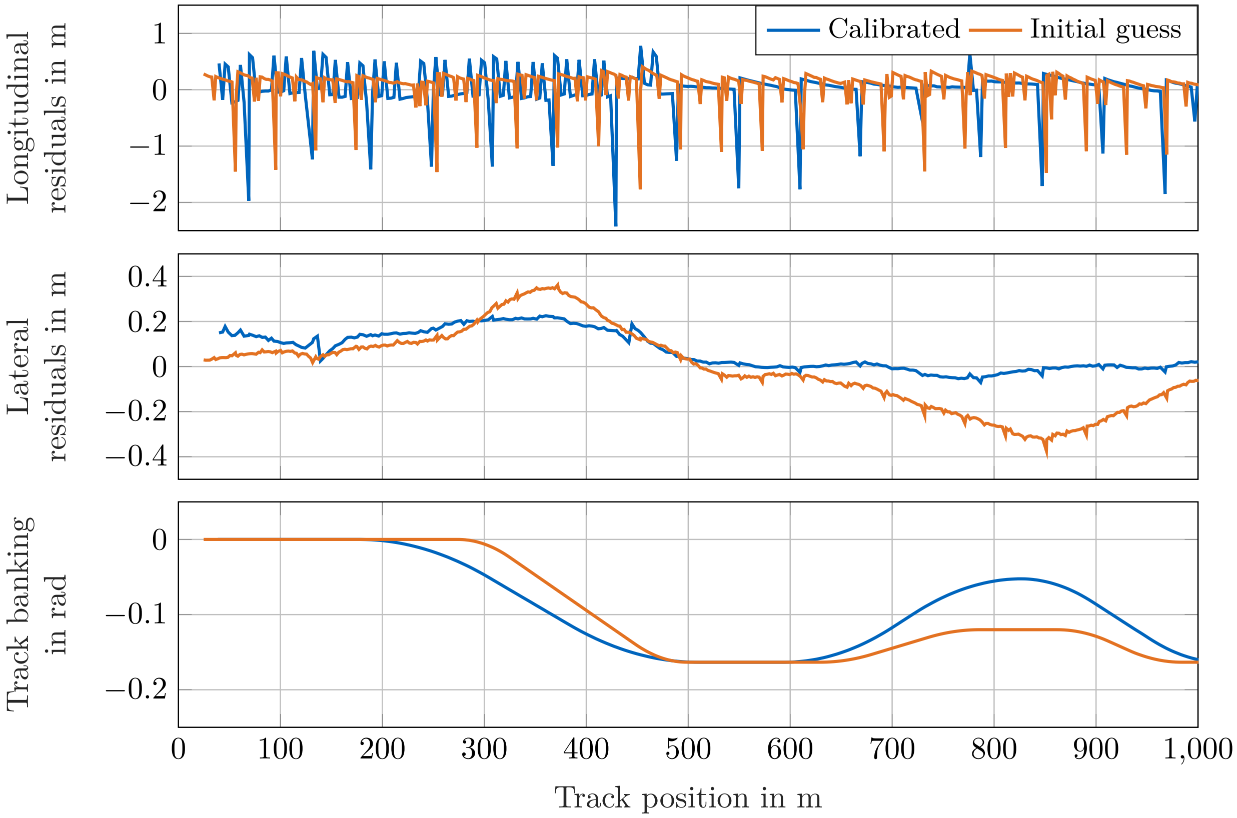


Figure 16: Impact of errors in the banking map on state estimation results

assumed to be precise and therefore the predictions of the model are expected to match well in case the model assumptions are fulfilled. The figure compares the state estimation results for an initial guess of the banking profile at the Indianapolis Motor Speedway with a refined version calibrated on measurement data. The mismatch of the real-world banking and the assumed banking profile led to significant residuals in the lateral direction for the initial guess of the banking map. This is due to a systematic error in the transformation of the accelerations from the vehicle coordinate frame into the planar coordinate frame of the state estimator based on the wrong banking assumption. While the manually calibrated banking map showed better results, a more sustainable strategy would be the introduction of a true three dimensional motion model and additional calibration states to identify calibration mismatch in the IMU itself.

The longitudinal residuals are dominated with timing effects related to the distributed operation of the sensor data receivers and the state estimation algorithm: At top speed (around $75\,\mathrm{m\,s^{-1}}$), the vehicle moves approximately $0.75\,\mathrm{m}$ within $10\,\mathrm{ms}$. This leads to high residuals when the data package is delayed (shown in the top plot of Figure 16), as the current implementation does not compensate for these effects. This effect is even more pronounced due to the utilization of ROS 2, an asynchronous messaging system, on a centralized high-performance compute unit. This is due to the fact, that the middleware and scheduling strategy does not guarantee a certain order of execution for message processing and scheduled tasks. Future implementations should be extended via a holistic handling of asynchronous and delayed sensor data. In addition, this requires to synchronize the clocks of all sensors via technologies like Precision Time Protocol (PTP) and pay in-depth attention to real-time capabilities.

Finally, it was found that the estimation algorithms have to be extended with capabilities to reliably detect faulty sensor signals. While the presented implementation had the ability to deactivate sensors which stopped sending data or provided a self-diagnosis flagging bad quality data, it assumes sensor data to be correct in all other circumstances. However, even high-quality measurement systems can have faults from time to time in the challenging racing environment resulting from vibrations and continuous high accelerations, leading to dangerous situations at high speeds due to faulty localization data caused by incorrect sensor data and therefore wrong corrective actions of the control system. A potential strategy to overcome this is the application of fault detection techniques and the subsequent utilization of these results in the sensor fusion algorithms. While the literature proposes algorithms to approach this problem [139], it has to be further investigated in the context of high-performance autonomous racing applications. In particular, this has to be done in conjunction with a thorough system engineering approach considering the required reliability targets already in the sensor selection and positioning to prevent increased cost and weight from too many redundant sensors on the vehicle.

5.1.2 Motion Controller

The proposed control system utilizes a model predictive controller based on the Tube-MPC scheme introduced in [117]. The prediction model is a friction limited point-mass model within a curvilinear coordinate frame associated with the reference path. This focus on the Newtonian mechanics allows the Tube-MPC to use the vehicles longitudinal and lateral accelerations as control input and therefore work with a low-dimensional state space model in contrast to more complex single-track models. The system is completed by the introduction of fast low-level control loops to track the target values for the accelerations. The application of this controller design during the various testing sessions and competitions of the Indy Autonomous Challenge has shown that it is a promising concept to decrease the dependence on accurate parameter estimates and deliver driving capabilities at the handling limits with little knowledge of model parameters (results and comparison to an MPC with a nonlinear tire model can be found in Section 3.3). In the presented design, only the wheelbase, the vehicle weight and a coarse engine and brake characteristic have to be known. It does not require details about the tire characteristic or suspension design.

The differences of the Tube-MPC controller used in this thesis to a standard nominal

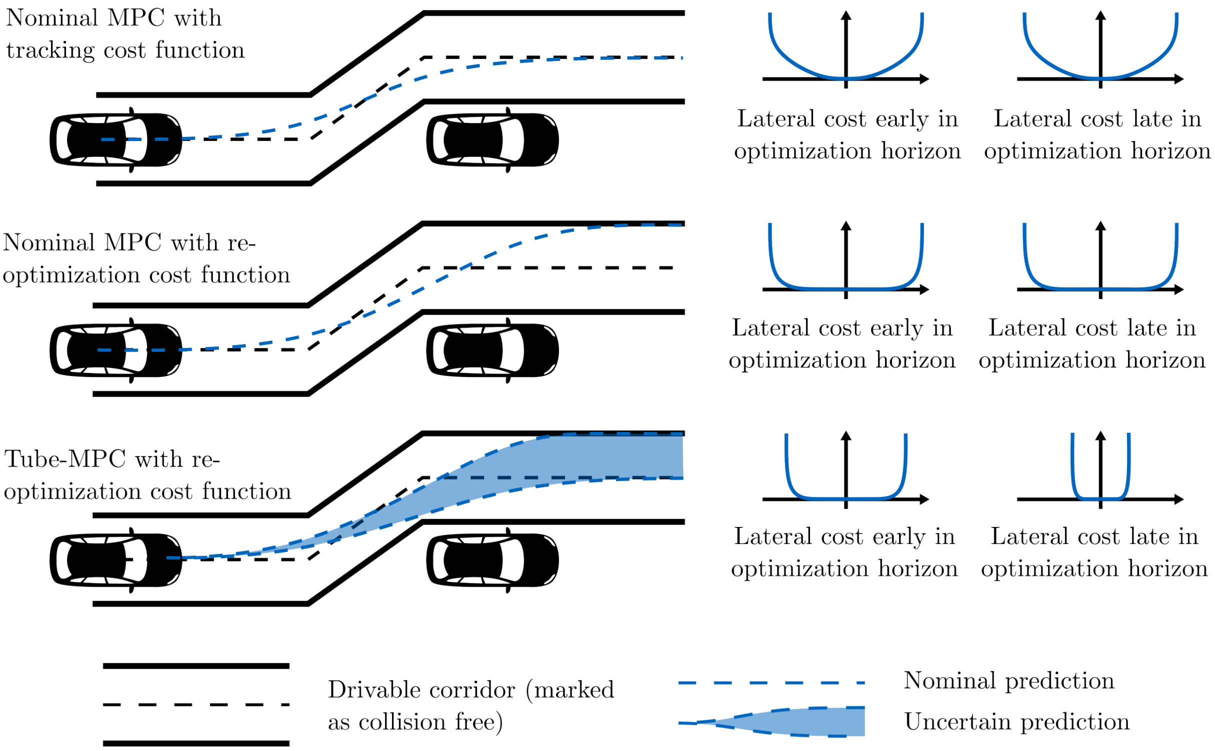

Nominal MPC with tracking cost function
Lateral cost early in optimization horizon
Lateral cost late in optimization horizon
Nominal MPC with re-optimization cost function
Lateral cost early in optimization horizon
Lateral cost late in optimization horizon
Tube-MPC with re-optimization cost function
Lateral cost early in optimization horizon
Lateral cost late in optimization horizon
Drivable corridor (marked as collision free)
Nominal prediction
Uncertain prediction

Figure 17: Impact of cost function structure on overall optimization result

tracking MPC are visualized in Figure 17. The figure shows the optimization result for different controller variants and the impact of the cost function design on the cost with respect to the lateral deviation from the reference line. The input for the optimization problem is a drivable corridor which is declared collision free by a high-level trajectory planner. The top row shows that a nominal tracking MPC formulation would choose to stay close to the middle of this drivable corridor to ensure proper tracking of the reference line. The cost increases significantly already for small deviations. It should be noted, that the cost visualization here includes the impact of the constraints as these are implemented as soft constraints in the problems shown in this thesis.

In contrast, the re-optimization case (middle row) has near zero cost for lateral deviation until it reaches the bounds of the drivable corridor represented as soft constraints. The advantage of this re-optimization cost function is that it can handle rather coarse target trajectories with smooth driving behavior and therefore achieve higher overall performance. In the nominal tracking case, already small deviations are penalized which leads to closer tracking of the middle of the drivable corridor. However, driving close to the constraints is risky when the motion prediction of the controller is not exact as the vehicle might leave the admissible driving corridor.

This challenge is overcome by the second difference to nominal tracking MPC: The proposed Tube-MPC utilizes a prediction over a set of possible trajectories (under the influence of uncertainties) instead of a single prediction (bottom row). As a result, the controller employs a safety distance to the constraints for long-term predictions (translating to 1.5 s to 2.5 s for the racing application) while still exploiting the full constraints for short-term predictions (from 0.5 s to 1 s). This behavior is achieved by the constrained tightening, which leads to a broad and gentle increase of the cost-function for short-term predictions and more narrow and steep increase for long-term predictions.

Even though the theoretical guarantees on recursive feasibility can not be applied directly to the real world system, the comparison to a nominal MPC (Section 3.2) and the extensive real-world testing on various racetracks have shown that this strategy to incorporate assumptions on the external acceleration disturbances is an effective way to adjust the aggressiveness of the closed-loop driving behavior. The assumed disturbance value acts as a risk-equivalent parameter and can be used to account for various parametric uncertainties (e.g. tire parameters), even though they have not been considered explicitly or with an accurate physical representation within the disturbance model. This finding questions what level of uncertainty modeling is required for successful application and motivates to focus on the outcome of the disturbance as a prediction uncertainty rather than on the detailed representation of their physical causes. While there is a significant amount of literature on the theoretical analysis of Robust MPC controllers and various uncertainty representation, a detailed comparison of their application performance should be conducted in the future.

However, the application of Tube-MPC can not eliminate all challenges associated with the real-world application of MPC. In practical applications, the constraints of the optimization problem are usually implemented in a soft-constraint fashion to prevent infeasibility of the optimization problem in edge-cases close to the constraints. This is done via the introduction of linear or quadratic penalty terms (or both) with large weights. For the case of linear cost terms and sufficiently high weights, it can be shown that they lead to the so called exact penalty property [118], which means that the soft-constrained optimization problem recovers the solution of the hard-constrained problem if its feasible. The same strategy applies for the used Tube-MPC concept: After calculating the uncertainty tube and conducting the constraint tightening procedure, the resulting (stage-dependent) constraints are implemented using a linear penalty function. This leads to similar challenges as in the nominal case when the optimization problem becomes infeasible. Even though this should not happen after careful choice of the uncertainty assumptions, its hard to prevent completely as sudden changes in the target trajectory or in the system dynamics, e.g. resulting from an actuator degradation, might render the chosen parameters slightly too optimistic. As a consequence, the optimization problem becomes challenging to solve from a numerical perspective due to the sudden increase in the cost function via the penalty term, which results in longer solver times which might violate real-time requirements. In addition, its hard to predict and tune how the optimal reaction to such cases looks like. The large impact on the cost function can

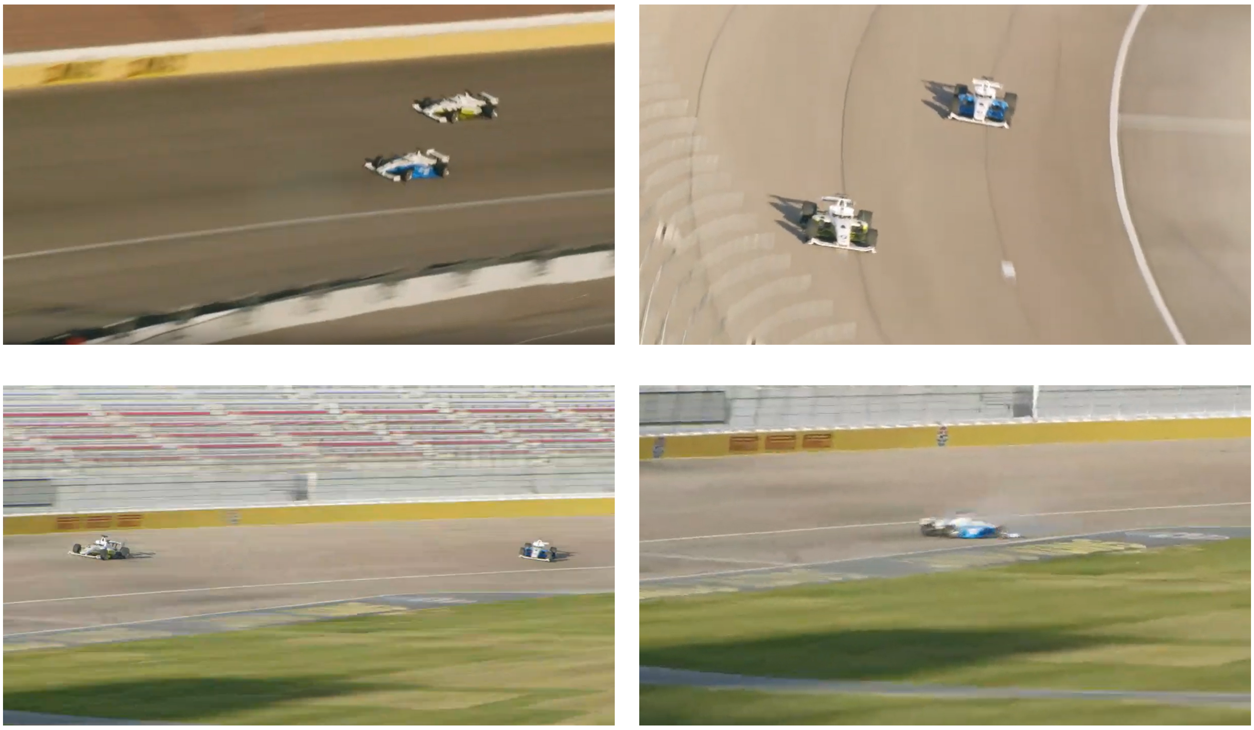


Figure 18: The final overtake in the final of the Autonomous Challenge at Las Vegas Motor Speedway and spin-out of the TUM vehicle at $250\,\mathrm{km\,h^{-1}}$. *Image credits: Indy Autonomous Challenge. Full video available at [33].*

lead to large corrective actions in the control commands, which can further destabilize the closed-loop dynamics as the actuators operate outside their modeling assumptions. Future work should investigate how the careful design of the penalty terms can be done to account for these effects and achieve the desired closed-loop behavior. A different strategy could be, to take an H_∞ interpretation to the robust MPC design problem: This would shift the focus onto shaping the cost-function such that it directly supports the robustness targets rather than relying only on the constraint tightening to achieve it. Furthermore, it would be in-line with many practical control engineering approaches, where constraint satisfaction is enforced only implicitly via sufficiently good control performance. This approach could increase the performance of the Tube-MPC in cases where the disturbance assumptions are slightly violated as it would only lead to slight performance degradation in contrast to the often erratic behavior observed when the problem becomes numerically hard to solve. Since pushing the vehicle to its maximum is a common scenario in racing, these alternative design strategies could be of significant interest for future work.

Finally, the proposed interface between the Tube-MPC and the low-level controllers, lateral and longitudinal acceleration, was found to perform well within stable driving situations but

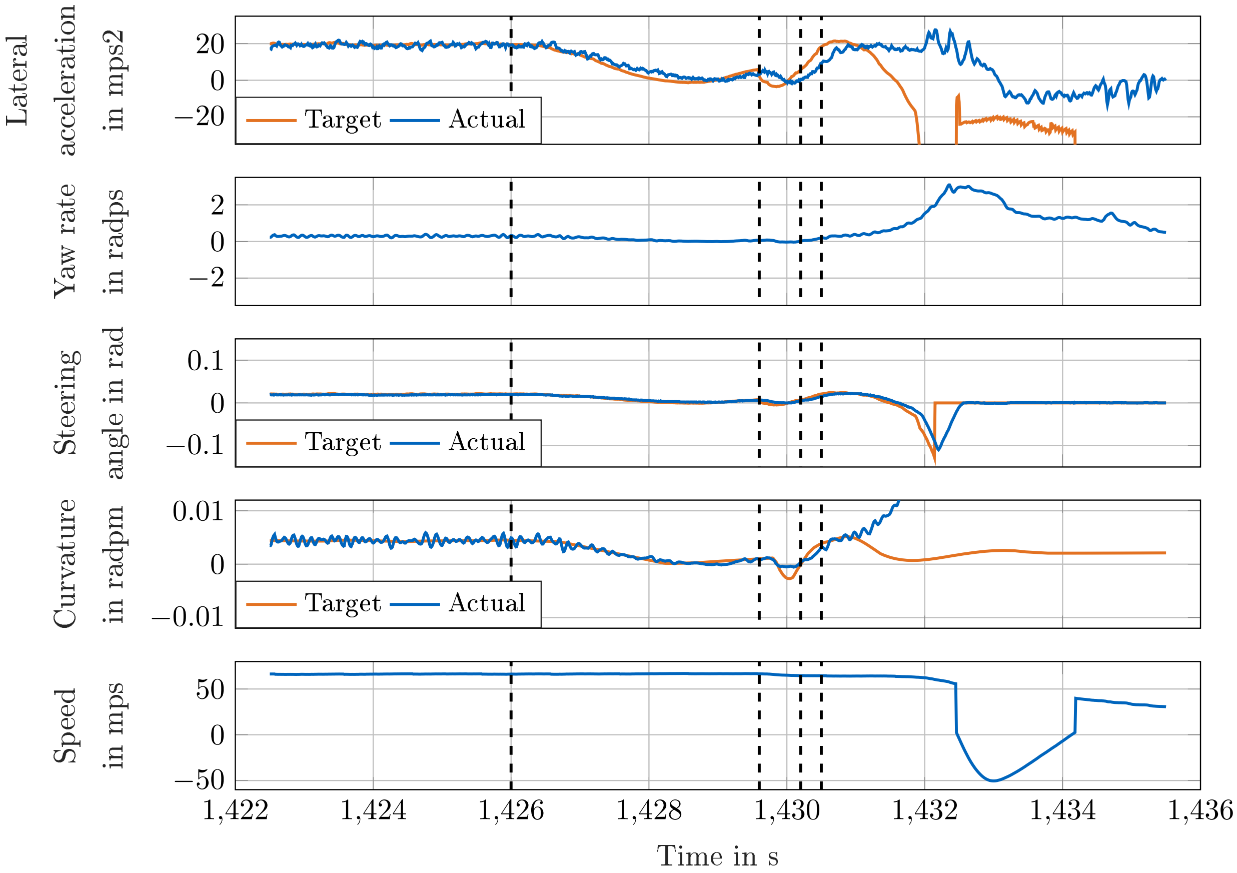


Figure 19: Vehicle dynamics data of the final overtake of the Autonomous Challege at the Las Vegas Motor Speedway. Positive values resemble left-hand turns.

lacks the ability to counteract potentially destabilizing situations. The interface was chosen, as it showed best performance in handling uncertainties in the self-steering behavior of the vehicle and therefore minimized the impact of the vehicle dynamics setup on the motion tracking performance. However, during unstable driving situation, such as the final of the Las Vegas race, depicted in Figure 18, it showed shortcomings. The detailed vehicle dynamics data related to this incident are shown in Figure 19.

The situation starts with the PoliMOVE vehicle (green) overtaking the TUM vehicle (blue) on the outside of the track. After this overtake was complete, both vehicles approached the apex of the long start-finish curve at the Las Vegas Motor Speedway (around 1426 s to 1429 s, first marker). At 1429.5 s (second marker) the perception stack of the TUM vehicle detected a false positive in front of the vehicle and initiated a sudden evasion maneuver to the right, followed by a strong correct to the left (starting at 1430.2 s, third marker). The requested target acceleration in this left-hand correction overloaded the rear-axle and led to the spin-out (starting at 1430.5 s, fourth marker, with the increase in yaw rate) and the short section of reverse movement (from 1433 s to 1434 s). Even though required, the controller did not counter-steer until its too late to regain control of the vehicle. This can be attributed to the controller only taking into account the lateral acceleration and the vehicle position. Additional consideration of the yaw rate, which is the first measurement value to indicate the spin-out, and the side slip angle could improve this behavior. Future research activities could investigate how the strengths of the low-level acceleration feedback can be combined with existing concepts based on yaw-rate and side-slip angle feedback. On top of that, it will be necessary to design this low-level control system as a joint longitudinal and lateral dynamics controller to handle a large variety of challenging driving scenarios. It should be kept in mind that the performance of these systems is fundamentally limited by the response times of the actuators. A control system which will match human driver capabilities in racecar handling will require the same reaction speeds as human drivers have, otherwise it will have a fundamental disadvantage. Therefore, further improvements in actuator technology are a necessary prerequisite for advancements in vehicle dynamics control design.

5.2 Data-Driven Methods

The second part of this thesis investigated two different data-driven methods to reduce the manual parameter and model adaptation effort required in state-of-the-art autonomous racing algorithms. The first approach, presented in Section 4.1, targets the online adjustment of a velocity profile scaling factor to achieve minimum laptime for a given pre-optimized raceline. This is done via a Safe Bayesian Optimization algorithm which considers several motion control performance indicators. The key characteristic of this algorithm is that it does not take into account any knowledge about the vehicle behavior and can therefore be considered a model-free learning algorithm. In contrast, the second approach, presented in Section 4.2 is a strategy to refine the feedforward control law based on model identification and uncertainty quantification. It leverages a recursive least mean squares algorithm in conjunction with a recursive quantile estimator. The resulting algorithm is embedded into the Tube-MPC motion controller presented earlier in this thesis in Section 4.3. It is used to reduce the average acceleration tracking error and adjust the uncertainty assumptions during operation.

The Safe Bayesian Optimization algorithm (based on the framework presented by [140]) utilizes individual Gaussian Processes to model the dependence of several safety indicators on the optimization variables (in this case a scale factor for the velocity profile of the optimal racing line). The relation is refined during the operation of the vehicle via measuring the safety indicators when executing a trial with a specific velocity scale factor. The qualifying scenario for autonomous racing allows to always pick the worst-case of a safety indicator over the course of a full lap. As a result, the vehicle adjusts the target scale factor every lap until it reaches the minimum lap-time without violating a constraint on one of the safety factors. The exploration speed is adjusted via the kernel parameters in the Gaussian Processes. The approach has shown to be well-suited to replace conventional tire-road friction coefficient estimation by using safety indicators which are good in revealing whether the vehicle is close to the handling limits, such as lateral control error, longitudinal wheel slip and the difference between front and rear axle side slip. This overcomes the disadvantage that tire-road friction coefficient estimation does not consider the actual capabilities of the control system driving the autonomous racing vehicle. Utilizing such a parameter based estimate naively in a motion control system with moderate driving capabilities can be compared to asking a novice driver to execute a raceline which was successfully driven by a professional driver, because the vehicle is capable to do this. In contrast, the proposed algorithm considers the safety indicators as a proxy for the joint capabilities of the motion control algorithm and the tire-road friction coefficient and can therefore safely achieve the handling limit for the current software configuration.

However, the downside of this approach is that it has to rely on measured and representative samples for the safety indicators. In the qualifying case, this is done via driving a full lap on a circuit which makes the system take a while to converge as it can only update the vehicle behavior after the completion of a full lap. In addition, the system will always pick the worst-case situation from the full lap to adjust the speed appropriately. This might lead to neglecting potential on certain sectors of the lap. The slow convergence over the course of several laps and minutes and reliance of a repeatable task such as qualifying laps make it hard to apply this approach to more complex situations such as multi-vehicle racing. In addition, the approach can not react *during* the lap in case external factors (such as rain) impact the optimization procedure. Another scenario where a change during a lap would be required, is if the algorithm has chosen an overly optimistic target value to evaluate and the vehicle barely completes the first turn. A human driver would adjust the driving style already here, however, this behavior can not be naturally embedded into the proposed algorithm. While there might be rule-based strategies to overcome these shortcomings, they are likely to be very specific to a certain situation and will have difficulties to generalize.

The discussed challenges with the model-free approach have led to a focus on model error learning strategies to enhance the capabilities of the existing Tube-MPC algorithm for the remainder of the thesis. The most popular framework here has been the learning of a GPR error model to cover non-modeled effects in the dynamic equations (e.g. [31, 125, 141]), however, it suffers from significant computational complexity for cases where larger datasets are required. In addition, the standard formulation can not deliver reliable uncertainty estimates for the case where the measurement uncertainty is not known exactly prior to data collection and learning (see Section 7.5 for a detailed discussion of this issue). The Recursive Uncertainty Model Identification algorithm proposed in Section 4.2 overcomes these challenges via a two-stage algorithm. The first stage, responsible for dealing with epistemic uncertainty, is handled via a recursive least squares algorithm to provide a computational efficient alternative to GPR. The second, dealing with the remaining aleatoric uncertainty, is done via a recursive quantile estimation strategy, which can quantify prediction errors with arbitrary distribution type. The key advantage of the proposed algorithm is that it's fully recursive and therefore well-suited for embedded applications with update frequencies in the range of 100 Hz. Furthermore, this resolves the problem of adjusting the learned model to changing environmental circumstances. In contrast to classical GPR, the recursive formulation includes a forgetting mechanism. Previous proposals in the literature implemented this with a resource intense data management strategy on top of the GPR [31] which makes them hard to apply on embedded control units.

Another advantage of the proposed algorithm is its consistency in terms of uncertainty estimation. Classical recursive least-squares and GPR assume the uncertainties to be known as hyperparameters prior to identification of the mean model. Their uncertainty estimate therefore only covers the effects of the unknown parameters within the mean value approximation (corresponding to the epistemic uncertainty within the problem) and do not adjust to the remaining uncertainty in the observed data (corresponding to aleatoric uncertainty). In fact, a standard GPR based uncertainty estimate will always converge to the pre-specified measurement uncertainty as soon as a sufficient number of measurements is available. If this parameter does not reflect the actual uncertainty within the process, the uncertainty estimate, and therefore the assumptions of the Tube-MPC, can be arbitrary wrong. This makes them a dangerous choice when using it in conjunction with a Tube-MPC algorithm as it can lead to overly optimistic driving style. While Bayesian Linear-Regression proposes a way to jointly estimate the mean regression model and the uncertainty parameters, it is restricted to gaussian error distributions. The same holds for joint optimization of model parameters and hyperparameters for the case of GPR. The recursive quantile estimation strategy overcomes this shortcoming by using the actual coverage of the data as an identification target which directly corresponds to a statistical quantile interpretation. However, the solution to this problem also causes a disadvantage: The identification of a quantile from measurement data without prior assumptions on the distribution requires a large dataset to achieve low variance and therefore sufficient reliability. This leads to a sample size requirement in the area of several thousand datapoints with data representative of the uncertainty which should be covered. Despite the raw memory requirements (which can be handled on state-of-the-art compute units), this leads to long observation time windows. Assuming a sample rate of 100 Hz this means that a low variance quantile estimation can easily take 30 s to 60 s. As a result, the response time to changes in the uncertainty characteristic can not be reduced arbitrarily. Even higher quality or frequency sensors would not resolve the problem as the samples also have to be representative and rather uncorrelated.

The final part of the thesis has been Section 4.3 with a combination of the Tube-MPC algorithm proposed in Section 3.3 and the learning algorithm proposed in Section 4.2. Even though the evaluation presented is only based on a simulation study, it sheds light on the potential of the combination of these two approaches. It aims to handle the overall uncertainty in the system via two different ways: First, the mean-error model learning reduces the epistemic uncertainty via refining the feedforward control law using the data collected while driving. Second, the uncertainty quantification algorithm measures the remaining uncertainty within the system and uses this value to parameterize the uncertainty assumptions of the Tube-MPC. It can be considered to quantify the aleatoric uncertainty and utilizing this value to apply sufficient caution. The proposed mean-error model manages to reduce the overall lateral acceleration uncertainty from 11.5 % to 9 %. As a result, the lap times have been reduced from 77.8 s to 76.5 s. Even though this improvement is remarkable considered the level of driving performance, the remaining uncertainty level indicates that the learning approach has not fully succeeded in identifying the non-modeled dynamics. This shortcoming is related to the fact that the adjusted feedforward control law could only account for static characteristics and did not imply possibly modification of dynamic response parameters or even the dynamic structure itself. While further degrees of freedom for the learning algorithm might improve the performance when it is done with parameter identification, an increasing number of optimized parameters makes it harder to predict the behavior of the learned feedforward law prior to convergence. This might lead to erratic reactions or other unwanted behavior.

6 Conclusion

Autonomous vehicle motion control has been a topic of on-going research for several decades. With increased maturity of driver-assistance systems, the focus of the motion control research community shifted towards situations where the vehicle is operated at the handling limits and the tire behavior becomes nonlinear. This was supported by the rise of multiple autonomous racing competitions, where research teams from all over the world get the chance to benchmark their algorithms on racecars of different size. The competitive nature of racing requires the motion control algorithms to operate safely close to the handling limit with only very little tolerance for error. While there are many ways to approach this challenge, model predictive control has shown to be one of the most promising strategies to handle the complex dynamics, safety constraints and combined steering, throttle and brake inputs. However, many approaches focus on the design of well-calibrated nonlinear dynamic models with a large number of states, which leads to complex and fragile numerical optimization strategies. This thesis aimed at reducing the dependency of the motion control algorithms on the availability of detailed nonlinear vehicle dynamics model via the application of robust and data-driven algorithms to the autonomous racing task.

The work was conducted in two separate streams: The first stream (focussed on robust control) presented a state estimator which is robust against parametric uncertainties within the vehicle model and a Tube-MPC based on a simple friction limited point-mass model in conjunction with fast low-level acceleration PI-feedback loops. Both concepts were evaluated under real-world conditions at speeds up to $265\,\mathrm{km\,h^{-1}}$ and accelerations of $21\,\mathrm{m\,s^{-2}}$ during the Indy Autonomous Challenge. Even though they only utilize basic information about the vehicle dynamic behavior, they achieve stable driving behavior at the handling limits and outperform strategies with more complex models in cases where their parameters are not well-calibrated. This emphasizes the strength of fast low-level feedback control, when the remaining uncertainties are handled within a robust framework such as Tube-MPC in the high-level controller. Another strength of the proposed concept is its re-optimization capability: The controller could jointly apply feedback and re-optimize the target trajectory such that smooth steering, throttle and brake actuations can be achieved. This is important in the multi-vehicle racing task, as the trajectory planner has to solve a complex, non-convex optimization problem due to the many available options. As a result, a coarser discretization during the graph-based trajectory planning could be applied which led to lower execution times and there fore faster reactions. The resulting motion control system dealt with an extensive amount of the vehicle dynamics range of the Indy Autonomous Challenge vehicle on varying oval racetracks with limited information (vehicle mass, wheelbase and coarse estimates of the brake and engine characteristics) about the dynamic system characteristics. In particular, it did not use information about the tire characteristics which is a significant advantage in comparison to many Nonlinear MPC approaches.

The second stream introduced two different ways of leveraging data-driven algorithms to improve the performance of an autonomous racing vehicle: The first approach adjusted the velocity profile scale factor of the trajectory planner to achieve minimum lap-time without violation of safety constraints, such as tire slip or lateral control error, using a Safe Bayesian Optimization approach. However, its strong focus on the optimization of a repeatable tasks limited its applicability in more complex situations as multi-vehicle racing. The second approach combined a recursive least-squares estimator with a recursive quantile estimator to adjust a linear (in the parameters) relation between features and output variables and estimate the residual variance. Finally, this approach was combined with the work from the first stream and demonstrated to increase the control performance via adjusting the feedforward control law for the lateral acceleration controller in a simulation study. Even though the decrease in lateral acceleration tracking error was significant (from 11.5 % to 9 %), the learning algorithm was not able to eliminate the uncertainty completely.

One of the unique contributions of this thesis was that most of the algorithms were benchmarked on a full-scale autonomous racing vehicle, starting with the DevBot 2.0 by Roborace and followed by the AV-21 at the Indy Autonomous Challenge on various racetracks in Europe and the United States. The resulting variance in vehicle dynamics as well as tire and track conditions posed a significant hurdle which was mainly overcome by the application of concepts which depend only on a coarse characterization of the vehicle rather than requiring a detailed model identification. The utilization of fast low-level acceleration feedback loops was a key enabler for this and showed to be more influential than the utilization of either of the learning algorithms. While these could be used successfully for certain fine-tuning aspects, they did not show the same fundamental impact on the overall system performance. These findings indicate that the inherent and irreducible uncertainty (aleatoric uncertainty) dominated in the applications evaluated for this thesis in contrast to epistemic uncertainties which could have been overcome by the application of learning algorithms. Considering all the potential influences on the vehicle dynamics (load dependent steering actuator characteristics, temperature dependent brake and engine characteristics, varying asphalt conditions on the same racetrack, temperature and wear dependent tire conditions, and many more) it seems to be a hard task to increase the model complexity of the learning schemes to an extent which is sufficiently rich to consider the aleatoric uncertainties as epistemic uncertainties.

While one might argue that professional race drivers are capable of handling all of these uncertainties, it remains questionable whether they do so via a long-term (on the scale of multiple laps or situations) oriented learning scheme or if they use other techniques, such as immediate and nuanced feedback to the vehicle behavior. Following the results of this work, it's more promising to refine the combination of robust high-level MPC and fast low-level feedback controllers. In particular, more advanced but still robust (independent from tire-road contact characteristics) low-level feedback laws are a promising direction of research, e.g. by taking into account a combination of yaw rate, lateral and longitudinal accelerations and the chassis side-slip angle and potentially designing a joint lateral and longitudinal feedback controller rather than two separate ones. In addition, improving the general capabilities of the vehicle to respond well in such fast feedback loops is likely to be required to achieve human performance. This includes improved sensor data quality, improved compute and networking capabilities to reduce the overall latency from sensor data to actuation requests, as well as reduced response times of the steering, brake and throttle actuation.

Finally, it remains to discuss the relation of the results of this work onto the development of highly automated and autonomous vehicles for highway and urban scenarios. The general design of the high-level MPC and the low-level acceleration feedback loops could be transferred to passenger car applications in a wide range of vehicle dynamics. The application for commercial vehicles such as semi-haulers is likely to require extensions to the vehicle motion model in more dynamic situations to cover for the impact of the trailer. However, one of the challenges which need to be overcome is the compensation of inclination and banking influence on the acceleration controllers. This could either be done similar as in the racing case, using a high-definition map of the environment or would require enhancements to the sensor fusion and additional information from the environment perception sensors. In addition, the significantly larger variance in road surface and weather conditions on the road is challenging for the proposed concept. While this variance could be handled via uncertainty assumptions, it does not make sense to handle all of these influences with a single uncertainty or parameter setup. This would lead to overly cautious driving behavior, e.g. the vehicle would always account for the possibility of a sudden thunderstorm during sunny weather. The handling of these different environment conditions should be done via other strategies, e.g. access of the motion controller to information provided by an environment perception system and other external sources such as weather forecasts.

References

Own Publications

[1] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, “Minimum curvature trajectory planning and control for an autonomous race car”, *Vehicle System Dynamics*, vol. 58, no. 10, pp. 1497–1527, 2019.

[2] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer, B. Lohmann, and M. Lienkamp, “What can we learn from autonomous level-5 motorsport?”, *Proceedings of the 9th International Munich Chassis Symposium 2018*, Springer Fachmedien Wiesbaden, 2018, pp. 123–146.

[3] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, L. Hermansdorfer, T. Stahl, T. Herrmann, and M. Lienkamp, “A software architecture for the dynamic path planning of an autonomous racecar at the limits of handling”, *Proceedings of the 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, IEEE, 2019, pp. 1–8.

[4] T. Stahl, A. Wischnewski, J. Betz, and M. Lienkamp, “Multilayer graph-based trajectory planning for race vehicles in dynamic scenarios”, *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, pp. 3149–3154.

[5] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer, and M. Lienkamp, “A software architecture for an autonomous racecar”, *Proceedings of the 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, IEEE, 2019, pp. 1–6.

[6] F. Christ, A. Wischnewski, A. Heilmeier, and B. Lohmann, “Time-optimal trajectory planning for a race car considering variable tyre-road friction coefficients”, *Vehicle System Dynamics*, vol. 59, no. 4, pp. 588–612, 2019.

[7] T. Stahl, A. Wischnewski, J. Betz, and M. Lienkamp, “ROS-based localization of a race vehicle at high-speed using LIDAR”, *E3S Web of Conferences*, vol. 95, pp. 1–6, 2019.

[8] A. Wischnewski, T. Stahl, J. Betz, and B. Lohmann, “Vehicle dynamics state estimation and localization for high performance race cars”, *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 154–161, 2019.

[9] J. Betz, A. Heilmeier, A. Wischnewski, T. Stahl, and M. Lienkamp, “Autonomous driving—a crash explained in detail”, *Applied Sciences*, vol. 9, no. 23, pp. 1–23, 2019.

[10] A. Wischnewski, J. Betz, and B. Lohmann, “A model-free algorithm to safely approach the handling limit of an autonomous racecar”, *Proceedings of the 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, IEEE, 2019, pp. 1–6.

[11] A. Wischnewski, J. Betz, and B. Lohmann, “Real-time learning of non-gaussian uncertainty models for autonomous racing”, *Proceedings of the 2020 59th IEEE Conference on Decision and Control (CDC)*, IEEE, 2020, pp. 609–615.

[12] F. Passigato, A. Wischnewski, A. Gordner, and F. Diermeyer, “Two approaches for the synthesis of a weave-wobble-stabilizing controller in motorcycles”, *Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 3496–3501.

[13] T. Herrmann, A. Wischnewski, L. Hermansdorfer, J. Betz, and M. Lienkamp, “Real-time adaptive velocity optimization for autonomous electric cars at the limits of handling”, *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 665–677, 2021.

[14] A. Wischnewski, M. Euler, S. Gümüs, and B. Lohmann, “Tube model predictive control for an autonomous race car”, *Vehicle System Dynamics*, vol. 60, no. 9, pp. 3151–3173, 2022.

[15] M. Rowold, A. Wischnewski, and B. Lohmann, “Constrained bayesian optimization of a linear feed-forward controller”, *IFAC-PapersOnLine*, vol. 52, no. 29, pp. 1–6, 2019.

[16] A. Wischnewski, T. Herrmann, F. Werner, and B. Lohmann, “A tube-MPC approach to autonomous multi-vehicle racing on high-speed ovals”, *IEEE Transactions on Intelligent Vehicles*, 2022.

[17] A. Wischnewski *et al.*, “Indy Autonomous Challenge - autonomous race cars at the handling limits”, *Proceedings of the 12th International Munich Chassis Symposium 2021*, Springer Berlin Heidelberg, 2022, pp. 163–182.

[18] J. Betz *et al.*, “TUM Autonomous Motorsport: An autonomous racing software for the indy autonomous challenge”, *arXiv.org*, 2022.

Other Publications

[19] McKinsey & Company, “Why the automotive future is electric”, 2021. [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/why-the-automotive-future-is-electric.

[20] Society of Automotive Engineers (SAE), “SAE J3016:2021 (Levels of Driving Automation)”, 2021.

[21] Z. Chai, T. Nie, and J. Becker, *Autonomous Driving Changes the Future.* Springer Singapore, 2021.

[22] L. Liu, Q. Zhang, R. Liu, X. Zhu, and Z. Ma, “Adaptive cruise control system evaluation according to human driving behavior characteristics”, *Actuators*, vol. 10, no. 5, pp. 1–14, 2021.

[23] J. Y. Goh, T. Goel, and J. C. Gerdes, “Toward automated vehicle control beyond the stability limits: Drifting along a general path”, *Journal of Dynamic Systems, Measurement, and Control*, vol. 142, no. 2, 2019.

[24] E. Wachter, M. Alirezaei, F. Bruzelius, and A. Schmeitz, “Path control in limit handling and drifting conditions using state dependent riccati equation technique”, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 234, no. 2-3, pp. 783–791, 2019.

[25] Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, “A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles”, *Vehicle System Dynamics*, vol. 52, no. 6, pp. 802–823, 2014.

[26] B. Yi, “Integrated planning and control for collision avoidance systems”, Ph.D. dissertation, Karlsruhe Institute of Technology, 2018.

[27] C. Rathgeber, “Trajektorienplanung und -folgeregelung für assistiertes bis hochautomatisiertes Fahren”, Ph.D. dissertation, Technische Universität Berlin, 2016.

[28] A. Agnihotri, M. O'Kelly, R. Mangharam, and H. Abbas, “Teaching autonomous systems at 1/10th-scale”, *Proceedings of the Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, ACM, 2020, pp. 657–663.

[29] U. Rosolia and F. Borrelli, “Learning how to autonomously race a car: A predictive control approach”, *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2713–2719, 2020.

[30] E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia, “Autonomous racing using linear parameter varying-model predictive control (LPV-MPC)”, *Control Engineering Practice*, vol. 95, pp. 1–8, 2020.

[31] J. Kabzan *et al.*, “AMZ driverless: The full autonomous racing system”, *Journal of Field Robotics*, pp. 1267–1294, 2020.

[32] S. Nekkah *et al.*, “The autonomous racing software stack of the KIT19d”, *SAE International Journal of Connected and Automated Vehicles*, vol. 5, no. 1, pp. 73–86, 2022.

[33] Indy Autonomous Challenge, “Video of the final passing competition of the indy autonomous challenge at las vegas motor speedway”, 2022. [Online]. Available: `https://www.youtube.com/watch?v=df9f4Qfa0uU`.

[34] H. B. Pacejka, *Tire and vehicle dynamics.* Amsterdam/Boston: Elsevier, 2012.

[35] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, “Autonomous vehicles on the edge: A survey on autonomous vehicle racing”, *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 458–488, 2022.

[36] L. Segel, “Theoretical prediction and experimental substantiation of the response of the automobile to steering control”, *Proceedings of the Institution of Mechanical Engineers: Automobile Division*, vol. 10, no. 1, pp. 310–330, 1956.

[37] E. Dickmanns and A. Zapp, “Autonomous high speed road vehicle guidance by computer vision”, *IFAC Proceedings Volumes*, vol. 20, no. 5, pp. 221–226, 1987.

[38] H. Peng and M. Tomizuka, “Vehicle lateral control for highway automation”, *Proceedings of the 1990 American Control Conference*, IEEE, 1990, pp. 788–794.

[39] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles”, *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[40] S. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. Eng, D. Rus, and M. Ang, “Perception, planning, control, and coordination for autonomous vehicles”, *Machines*, vol. 5, no. 1, 2017.

[41] S. Thrun *et al.*, “Stanley: The robot that won the DARPA grand challenge”, *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[42] N. R. Kapania and J. C. Gerdes, “Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling”, *Vehicle System Dynamics*, vol. 53, no. 12, pp. 1687–1704, 2015.

[43] M. Werling, “Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien”, Ph.D. dissertation, Karlsruhe Institute of Technology, 2011.

[44] L. König, “Ein virtueller Testfahrer für den querdynamischen Grenzbereich”, Ph.D. dissertation, Renningen, 2009.

[45] S. Antonov, A. Fehn, and A. Kugi, “A new flatness-based control of lateral vehicle dynamics”, *Vehicle System Dynamics*, vol. 46, no. 9, pp. 789–801, 2008.

[46] S. Fuchshumer, K. Schlacher, and T. Rittenschober, “Nonlinear vehicle dynamics control - a flatness based approach”, *Proceedings of the Proceedings of the 44th IEEE Conference on Decision and Control*, IEEE, 2005, pp. 6492–6497.

[47] R. Solea, A. Filipescu, V. Minzu, and S. Filipescu, "Sliding-mode trajectory-tracking control for a four-wheel-steering vehicle", *Proceedings of the IEEE ICCA 2010*, IEEE, 2010, pp. 382–387.

[48] M. Manceur and L. Menhour, "Higher order sliding mode controller for driving steering vehicle wheels: Tracking trajectory problem", *Proceedings of the 52nd IEEE Conference on Decision and Control*, IEEE, 2013, pp. 3073–3078.

[49] D. Calzolari, B. Schurmann, and M. Althoff, "Comparison of trajectory tracking controllers for autonomous vehicles", *Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2017, pp. 1–8.

[50] R. Y. Hindiyeh and J. C. Gerdes, "A controller framework for autonomous drifting: Design, stability, and experimental validation", *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 5, 2014.

[51] L. del Re, F. Allgöwer, L. Glielmo, C. Guardiola, and I. Kolmanovsky, *Automotive Model Predictive Control.* Springer London, 2010.

[52] 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*, vol. 3, no. 2-4, pp. 265–291, 2005.

[53] 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*, vol. 15, no. 3, pp. 566–580, 2007.

[54] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "A model predictive control approach for combined braking and steering in autonomous vehicles", *Proceedings of the 2007 Mediterranean Conference on Control & Automation*, IEEE, 2007.

[55] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads", *Proceedings of the Proceedings of the ASME 2010 Dynamic Systems and Control Conference*, ASME Digital Collection, 2010, pp. 265–272.

[56] A. Katriniok, J. P. Maschuw, F. Christen, L. Eckstein, and D. Abel, "Optimal vehicle dynamics control for combined longitudinal and lateral autonomous vehicle guidance", *Proceedings of the 2013 European Control Conference (ECC)*, IEEE, 2013, pp. 974–979.

[57] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars", *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2014.

[58] T. Novi, A. Liniger, R. Capitani, and C. Annicchiarico, "Real-time control for at-limit handling driving on a predefined path", *Vehicle System Dynamics*, vol. 58, no. 7, pp. 1007–1036, 2019.

[59] R. Verschueren, S. D. Bruyne, M. Zanon, J. V. Frasch, and M. Diehl, “Towards time-optimal race car driving using nonlinear MPC in real-time”, *Proceedings of the 53rd IEEE Conference on Decision and Control*, IEEE, 2014, pp. 2505–2510.

[60] J. L. Vazquez, M. Bruhlmeier, A. Liniger, A. Rupenyan, and J. Lygeros, “Optimization-based hierarchical motion planning for autonomous racing”, *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 2397–2403.

[61] D. Kloeser, T. Schoels, T. Sartor, A. Zanelli, G. Prison, and M. Diehl, “NMPC for racing using a singularity-free path-parametric model with obstacle avoidance”, *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14 324–14 329, 2020.

[62] J. K. Subosits and J. C. Gerdes, “From the racetrack to the road: Real-time trajectory replanning for autonomous driving”, *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 309–320, 2019.

[63] A. Raji, A. Liniger, A. Giove, A. Toschi, N. Musiu, D. Morra, M. Verucchi, D. Caporale, and M. Bertogna, “Motion planning and control for multi vehicle autonomous racing at high speeds”, *arXiv.org*, 2022.

[64] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, “Aggressive driving with model predictive path integral control”, *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 1433–1440.

[65] G. Raffo, G. Gomes, J. Normey-Rico, C. Kelber, and L. Becker, “A predictive controller for autonomous vehicle path tracking”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 92–102, 2009.

[66] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, “Kinematic and dynamic vehicle models for autonomous driving control design”, *Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2015, pp. 1094–1099.

[67] M. Brown, J. Funke, S. Erlien, and J. C. Gerdes, “Safe driving envelopes for path tracking in autonomous vehicles”, *Control Engineering Practice*, vol. 61, pp. 307–316, 2017.

[68] 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*, vol. 25, no. 4, pp. 1204–1216, 2017.

[69] A. Liniger and L. V. Gool, “Safe motion planning for autonomous driving using an adversarial road model”, *Proceedings of the Robotics: Science and Systems 2020*, Robotics: Science and Systems Foundation, 2020.

[70] A. Liniger and J. Lygeros, “Real-time control for autonomous racing based on viability theory”, *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 464–478, 2019.

[71] A. Carvalho, S. Lefévre, G. Schildbach, J. Kong, and F. Borrelli, “Automated driving: The role of forecasts and uncertainty—a control perspective”, *European Journal of Control*, vol. 24, pp. 14–32, 2015.

[72] A. Bemporad and M. Morari, “Robust model predictive control: A survey”, *Robustness in identification and control. Lecture Notes in Control and Information Sciences*, pp. 207–226, 1999.

[73] D. Mayne, “Robust and stochastic MPC: Are we going in the right direction?”, *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 1–8, 2015.

[74] J. Yu, X. Guo, X. Pei, Z. Chen, M. Zhu, and B. Gong, “Robust model predictive control for path tracking of autonomous vehicle”, *SAE Technical Paper Series*, 2019.

[75] E. Alcalá, V. Puig, J. Quevedo, and O. Sename, “Fast zonotope-tube-based LPV-MPC for autonomous vehicles”, *IET Control Theory & Applications*, vol. 14, no. 20, pp. 3676–3685, 2020.

[76] R. Soloperto, J. Kohler, F. Allguwer, and M. A. Muller, “Collision avoidance for uncertain nonlinear systems with moving obstacles using robust model predictive control”, *Proceedings of the 2019 18th European Control Conference (ECC)*, IEEE, 2019, pp. 811–817.

[77] A. Carvalho, Y. Gao, S. Lefevre, and F. Borrelli, “Stochastic predictive control of autonomous vehicles in uncertain environments”, *Proceedings of the 12th International Symposium on Advanced Vehicle Control*, 2014.

[78] D. Lenz, T. Kessler, and A. Knoll, “Stochastic model predictive controller with chance constraints for comfortable and safe driving behavior of autonomous vehicles”, *Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2015, pp. 292–297.

[79] A. Liniger, X. Zhang, P. Aeschbach, A. Georghiou, and J. Lygeros, “Racing miniature cars: Enhancing performance using stochastic MPC and disturbance feedback”, *Proceedings of the 2017 American Control Conference (ACC)*, IEEE, 2017, pp. 5642–5647.

[80] J. V. Carrau, A. Liniger, X. Zhang, and J. Lygeros, “Efficient implementation of randomized MPC for miniature race cars”, *Proceedings of the 2016 European Control Conference (ECC)*, IEEE, 2016, pp. 957–962.

[81] J. P. Alsterda, M. Brown, and J. C. Gerdes, “Contingency model predictive control for automated vehicles”, *Proceedings of the 2019 American Control Conference (ACC)*, IEEE, 2019, pp. 717–722.

[82] N. R. Kapania and J. C. Gerdes, “Path tracking of highly dynamic autonomous vehicle trajectories via iterative learning control”, *Proceedings of the 2015 American Control Conference (ACC)*, IEEE, 2015, pp. 2753–2758.

[83] N. R. Kapania and J. C. Gerdes, “Learning at the racetrack: Data-driven methods to improve racing performance over multiple laps”, *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8232–8242, 2020.

[84] U. Rosolia, A. Carvalho, and F. Borrelli, “Autonomous racing using learning model predictive control”, *Proceedings of the 2017 American Control Conference (ACC)*, 2017, pp. 5115–5120.

[85] U. Rosolia and F. Borrelli, “Learning model predictive control for iterative tasks: A computationally efficient approach for linear system”, *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3142–3147, 2017, 20th IFAC World Congress.

[86] M. Brunner, U. Rosolia, J. Gonzales, and F. Borrelli, “Repetitive learning model predictive control: An autonomous racing example”, *Proceedings of the 2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 2545–2550.

[87] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, “Learning-based model predictive control for autonomous racing”, *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.

[88] L. Hewing, A. Liniger, and M. N. Zeilinger, “Cautious NMPC with gaussian process dynamics for autonomous miniature race cars”, *Proceedings of the 2018 European Control Conference (ECC)*, 2018, pp. 1341–1348.

[89] B. van Niekerk, A. Damianou, and B. Rosman, “Online constrained model-based reinforcement learning”, *arXiv.org*, 2020.

[90] A. Jain, M. O’Kelly, P. Chaudhari, and M. Morari, “Bayesrace: Learning to race autonomously using prior experience”, *Proceedings of the Proceedings of the 2020 Conference on Robot Learning*, J. Kober, F. Ramos, and C. Tomlin, Eds., ser. Proceedings of Machine Learning Research, vol. 155, PMLR, 2021, pp. 1918–1929.

[91] L. Hermansdorfer, R. Trauth, J. Betz, and M. Lienkamp, “End-to-end neural network for vehicle dynamics modeling”, *Proceedings of the 2020 6th IEEE Congress on Information Science and Technology (CiSt)*, 2020, pp. 407–412.

[92] N. A. Spielberg, M. Brown, N. R. Kapania, J. C. Kegelman, and J. C. Gerdes, “Neural network vehicle models for high-performance automated driving”, *Science Robotics*, vol. 4, no. 28, 2019.

[93] X. Ji, X. He, C. Lv, Y. Liu, and J. Wu, “Adaptive-neural-network-based robust lateral motion control for autonomous vehicle at driving limits”, *Control Engineering Practice*, vol. 76, pp. 41–53, 2018.

[94] L. Hermansdorfer, J. Betz, and M. Lienkamp, “A concept for estimation and prediction of the tire-road friction potential for an autonomous racecar”, *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, pp. 1490–1495.

[95] S. Khaleghian, A. Emami, and S. Taheri, “A technical survey on tire-road friction estimation”, *Friction*, vol. 5, no. 2, pp. 123–146, 2017.

[96] M. Wielitzka, A. Busch, M. Dagen, and T. Ortmaier, “Unscented kalman filter for state and parameter estimation in vehicle dynamics”, in *Kalman Filters*, G. L. de Oliveira Serra, Ed., Rijeka: IntechOpen, 2017, ch. 3.

[97] M. Acosta and S. Kanarachos, “Tire lateral force estimation and grip potential identification using neural networks, extended kalman filter, and recursive least squares”, *Neural Computing and Applications*, vol. 30, no. 11, pp. 3445–3465, 2017.

[98] L. Chen, M. Bian, Y. Luo, and K. Li, “Real-time identification of the tyre–road friction coefficient using an unscented kalman filter and mean-square-error-weighted fusion”, *Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering*, vol. 230, pp. 788–802, 2015.

[99] M. Nolte, N. Kister, and M. Maurer, “Assessment of deep convolutional neural networks for road surface classification”, *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 381–386.

[100] T. Weiss and M. Behl, “DeepRacing: Parameterized trajectories for autonomous racing”, *arXiv.org*, 2020.

[101] A. Remonda, S. Krebs, E. Veas, G. Luzhnica, and R. Kern, “Formula RL: deep reinforcement learning for autonomous racing using telemetry data”, *arXiv.org*, 2021.

[102] F. Fuchs, Y. Song, E. Kaufmann, D. Scaramuzza, and P. Duerr, “Super-human performance in Gran Turismo Sport using deep reinforcement learning”, *arXiv.org*, 2020.

[103] Y. Song, H. Lin, E. Kaufmann, P. Durr, and D. Scaramuzza, “Autonomous overtaking in Gran Turismo Sport using curriculum reinforcement learning”, *Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 9403–9409.

[104] W. Milliken, *Race car vehicle dynamics.* Warrendale: SAE International, 1995.

[105] D. Schramm, M. Hiller, and R. Bardini, “Single track models”, in *Vehicle Dynamics*, Springer Berlin Heidelberg, 2014, pp. 223–253.

[106] P. Polack, F. Altche, B. d'Andrea Novel, and A. de La Fortelle, “The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?”, *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2017, pp. 812–818.

[107] J. Rawlings, *Model predictive control : theory, computation, and design.* Santa Barbara, California: Nob Hill Publishing, 2020.

[108] D. Q. Mayne, “Model predictive control: Recent developments and future promise”, *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[109] D. Mayne, M. Seron, and S. Raković, “Robust model predictive control of constrained linear systems with bounded disturbances”, *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[110] L. Grüne, *Nonlinear model predictive control theory and algorithms.* Cham: Springer, 2017.

[111] D. Limon, T. Alamo, and E. Camacho, "Stable constrained MPC without terminal constraint", *Proceedings of the Proceedings of the 2003 American Control Conference, 2003.*, IEEE, pp. 4893–4898.

[112] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control", *SIAM Journal on Control and Optimization*, vol. 43, no. 5, pp. 1714–1736, 2005.

[113] M. Diehl, R. Findeisen, H. Bock, F. Allgöwer, and J. Schlöder, "Nominal stability of real-time iteration scheme for nonlinear model predictive control", *IEEE Proceedings - Control Theory and Applications*, vol. 152, no. 3, pp. 296–308, 2005.

[114] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "Acados—a modular open-source framework for fast embedded optimal control", *Mathematical Programming Computation*, vol. 14, no. 1, pp. 147–183, 2021.

[115] G. Frison and M. Diehl, "HPIPM: A high-performance quadratic programming framework for model predictive control", *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.

[116] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs", *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[117] L. Chisci, J. Rossiter, and G. Zappa, "Systems with persistent disturbances: Predictive control with restricted constraints", *Automatica*, vol. 37, no. 7, pp. 1019–1028, 2001.

[118] E. Kerrigan and J. Maciejowski, "Soft constraints and exact penalty functions in model predictive control", *Proceedings of the UKACC International Conference Control 2000*, 2000.

[119] J. Loefberg, "Min-max approaches to robust model predictive control", Ph.D. dissertation, Linköping University, 2003.

[120] J. Köhler, M. A. Müller, and F. Allgöwer, "A novel constraint tightening approach for nonlinear robust model predictive control", *2018 Annual American Control Conference (ACC)*, pp. 728–734, 2018.

[121] S. V. Raković, "Invention of prediction structures and categorization of robust MPC syntheses", *IFAC Proceedings Volumes*, vol. 45, no. 17, pp. 245–273, 2012.

[122] D. Simon, *Optimal State Estimation - Kalman, H Infinity, and Nonlinear Approaches.* New York: John Wiley & Sons, 2006.

[123] F. Bayer, M. Burger, and F. Allgower, "Discrete-time incremental ISS: A framework for robust NMPC", *Proceedings of the 2013 European Control Conference (ECC)*, IEEE, 2013, pp. 2068–2073.

[124] M. Althoff, “Reachability analysis and its application to the safety assessment of autonomous cars”, Ph.D. dissertation, Technische Universität München, 2010.

[125] T. Koller, F. Berkenkamp, M. Turchetta, J. Boedecker, and A. Krause, “Learning-based model predictive control for safe exploration and reinforcement learning”, *arXiv.org*, 2019.

[126] M. Althof, “An introduction to CORA”, *Proceedings of the Workshop on Applied Verification for Continuous and Hybrid Systems*, 2015.

[127] A. Mesbah, “Stochastic model predictive control: An overview and perspectives for future research”, *IEEE Control Systems*, vol. 36, no. 6, pp. 30–44, 2016.

[128] A. Kurzhanski, “Ellipsoidal calculus for estimation and feedback control”, in *Systems and control in the twenty-first century*, ser. Progress in Systems and Control Theory, C. I. Byrnes, Ed., vol. 277, Boston: Birkhäuser, 1997, pp. 229–243.

[129] D. van Hessem and O. Bosgra, “Closed-loop stochastic dynamic process optimization under input and state constraints”, *Proceedings of the 2002 American Control Conference*, IEEE, 2002, pp. 2023–2028.

[130] K. P. Murphy, *Machine learning: a probabilistic perspective.* Cambridge, Mass.: MIT Press, 2013.

[131] C. M. Bishop, *Pattern Recognition and Machine Learning.* Berlin, Heidelberg: Springer, 2006.

[132] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning.* Cambridge: MIT Press, 2005.

[133] O. Föllinger, *Regelungstechnik: Einführung in die Methoden und ihre Anwendung*, German, Hardcover. VDE Verlag GmbH, 2016.

[134] J. C. Doyle, B. A. Francis, and A. Tannenbaum, *Feedback Control Theory -.* New York: Dover, 2009.

[135] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, English, Paperback. Pearson, 1995, p. 616.

[136] E. Hüllermeier and W. Waegeman, “Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods”, *Machine Learning*, vol. 110, no. 3, pp. 457–506, 2021.

[137] A. D. Kiureghian and O. Ditlevsen, “Aleatory or epistemic? does it matter?”, *Structural Safety*, vol. 31, no. 2, pp. 105–112, 2009.

[138] A. Yazidi and H. Hammer, “Multiplicative update methods for incremental quantile estimation”, *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 746–756, 2019.

[139] S. X. Ding, *Model-based fault diagnosis techniques: Design schemes, algorithms, and tools.* Springer, 2008.

[140] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with gaussian processes", *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 491–496.

[141] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using gaussian process regression", *IEEE Transactions on Control Systems Technology*, pp. 1–8, 2019.

7 Appendix

7.1 Vehicle dynamics state estimation and localization for high performance race cars

Contributions: Alexander Wischnewski, the author of this dissertation, contributed to the overall system architecture as well as the different estimator designs. Tim Stahl contributed to the design of the visual localization algorithms and the network communication. Johannes Betz and Boris Lohmann contributed equally to the conception of the research project and revised the paper critically for important intellectual content.

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-8 (2019) 154–161

Vehicle Dynamics State Estimation and Localization for High Performance Race Cars [★]

Alexander Wischnewski [*] **Tim Stahl** [**] **Johannes Betz** [***]
Boris Lohmann [****]

[*] *Chair of Automatic Control, Department of Mechanical Engineering, Technical University of Munich, Germany (e-mail: alexander.wischnewski@tum.de)*
[**] *Chair of Automotive Technology, Department of Mechanical Engineering, Technical University of Munich, Germany (e-mail: stahl@ftm.mw.tum.de)*
[***] *Chair of Automotive Technology, Department of Mechanical Engineering, Technical University of Munich, Germany (e-mail: betz@ftm.mw.tum.de)*
[****] *Chair of Automatic Control, Department of Mechanical Engineering, Technical University of Munich, Germany (e-mail: lohmann@tum.de)*

Abstract: Autonomous driving requires accurate information about the vehicle pose and motion state in order to achieve precise tracking of the planned trajectory. In this paper we propose a robust architecture to localize a high performance race car and show experimental results with speeds up to 150 $\mathrm{km\,h^{-1}}$ and utilizing approximately 80% of the available friction level. The concept has been applied using the development vehicle *DevBot* taking part in the Roborace competition. To achieve robust and reliable performance, we use two independent localization pipelines, one based on GPS and one on LIDARs. We propose to fuse them via a Kalman Filter based on a purely kinematic model and show, that it can outperform a high fidelity model under realistic race conditions. An outstanding property of this concept is that it does not depend on any of the vehicles parameter and is therefore robust to varying tire and track conditions. Further we present an adaption method for the measurement covariances based on the track layout. This allows to combine the strengths of each localization method.

Keywords: State Estimation, Sensor Fusion, Robust Performance, Autonomous Vehicles, Kalman Filters

1. INTRODUCTION

Racing has been driving the advance of automotive technology for decades. The combination of high-performance vehicles and people's enthusiasm allows to identify potential research directions and shortcomings of already available technology, since the vehicle's limits are pushed in order to minimize the achieved lap times.

Fig. 1. Research Plattform - *DevBot*

A research group of the TUM is taking part in the Roborace Competition, which aims at being the first ever full size racing series for autonomous vehicles (Betz et al. (2018)). All competitors use identical cars which are equipped with several vehicle dynamics and environment perception sensors. The experimental setup is a full-sized race car called *DevBot* (see Figure 1). The computations are carried out on a Speedgoat Mobile Target Machine and a NVIDIA Drive PX2. Details on the vehicle can be found in Betz et al. (2018). The TUM software stack reached a near-human performance level, was able to drive

[★] Research was supported by the basic research fund of the Institute of Automotive Technology of the Technical University of Munich.

10.1016/j.ifacol.2019.08.064

at $150\,\mathrm{km\,h^{-1}}$ and utilized the friction circle up to 80% of the physical limit.

In this paper, we present a Kalman Filter based algorithm to fuse the independent localization pipelines for GPS and LIDAR with an odometry based on vehicle dynamics sensors. The motorsports application demands that the full range of vehicle dynamics, changing track conditions and setup changes are covered without quality degradation. At the same time, it is crucial that the system is robust and easy to tune since testing time is limited and costly. In the following, we show that filter models based purely on kinematic equations can outperform a high fidelity vehicle model in terms of performance and robustness. Furthermore, we apply several modifications to the standard Kalman Filter algorithm to maximize the performance for the given race track scenario.

The remaining paper is organized as follows: Section 2 reviews the state of the art of autonomous driving state estimation and localization, while Section 3 recapitulates the theoretical fundamentals of the applied concept. Section 4 outlines the system architecture, Section 5 goes into detail regarding the state estimation concept used and Section 6 presents experimental data from the real vehicle.

2. RELATED WORK

The state estimation applications in vehicle technology can be roughly separated into two tasks:

- Estimation of the vehicle dynamic state (longitudinal and lateral velocity, yaw rate)
- Localization of the vehicle (position and orientation, also known as pose)

The former has been in the focus of research for several decades. A comparison of the results can be found in Guo et al. (2018). In general, one can divide the models used into three types: detailed vehicle models with tire force characteristics, slip-free constant velocity kinematic models and kinematic models also considering accelerations. The most common representative of the first category is the pure lateral single track model (Farrelly and Wellstead (1996); Haiyan and Hong (2006)). The longitudinal velocity is considered as a parameter but not estimated within a joint framework. Zhao et al. (2011) use a three degrees of freedom model to overcome this drawback. Even more complex models were applied but they are usually tailored to the specific use case and lack good generalization properties. Examples are the extension to more degrees of freedom (Wenzel et al. (2006)), use of suspension parameter knowledge (Antonov et al. (2011)) or the addition of parameter estimations (Wielitzka et al. (2014)). In contrast, slip-free constant velocity kinematic models provide higher robustness with respect to parameter uncertainty and are easier to tune since they require only the effective front axle steering angle (Kang et al. (2014)). They are mostly used within robotics to improve pose accuracy (Bonnifait et al. (2001)), rather than in vehicle dynamics related applications since they lack accuracy under significant accelerations. The third approach relies on the use of first principles modelling by the use of kinematic equations (Du and Li (2014); Farrelly and Wellstead (1996)). Instead of modelling the kinematic relations of the vehicle based on the steering angle, they rely on the measured lateral and longitudinal acceleration as the system inputs. This technique is also known as Kinematic Kalman Filter (KKF) (Jeon (2010)). Although it does not consider potentially available detailed knowledge about the model it is considered to be robust to parameter variations.

Following the trend towards autonomous driving, the focus is nowadays shifting towards the localization task and the necessary system architectures. While automotive applications have been concentrating on GPS at first, reliability issues posed the need for more robust localization methods. The progress already made in the robotics community led to the applications of probabilistic methods based on visual sensors like LIDARs and cameras. Although the former is the most common and mature framework (Thrun et al. (2005); Cadena et al. (2016)), algorithms which utilize cameras are emerging recently (Usenko et al. (2015); Mur-Artal et al. (2015)). The strengths and weaknesses of different environment perception sensors are discussed in Kuutti et al. (2018). The LIDAR measurements give direct information about the environment and object distances, which can be used for mapping and localization directly. In contrast, camera-based methods require complex post-processing steps to generate map data.

Several authors propose to increase localization performance by fusing different sensor sources within a single algorithm (Trehard et al. (2015); Suhr et al. (2017)). However, we found that this approach leads to difficulties during practical realization as the algorithm is a single point of failure in the system design, and it is hard to implement online diagnosis and reconfiguration possibilities. Bresson et al. (2016) and Jang et al. (2015) propose ideas for overcoming this limitation using a decision method to switch between different localization algorithms. The main drawback of this concept is that it does not combine all available information with respect to the corresponding characteristics under normal operating conditions.

The contribution of the paper is threefold: First, we propose a highly modular sensor fusion framework based on parallel localization pipelines and vehicle sensors using a Kalman Filter. Second, we show that a simple point mass model is sufficient and can even outperform a more sophisticated model for the full speed and nonlinear tire range based on data gathered with a high performance race car. Third, we introduce a covariance adaption for each localization pipeline based on previously available map data.

3. METHODOLOGY

In this section, we present the vehicle dynamics models used later for the design of the state estimator and the theoretical fundamentals of Kalman Filter based sensor fusion and LIDAR based localization.

3.1 Vehicle dynamics

Sensor fusion requires that the signals are related to each other using a dynamic system model. Due to its good trade-off between complexity and accuracy (Milliken and Milliken (1996)), even for operating points within the

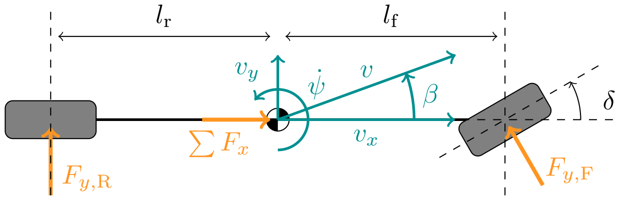


Fig. 2. Nonlinear single track model

nonlinear tire range, the nonlinear single track model is widely applied for control and estimation purposes once the vehicle operates near the handling limit.

The vehicle velocity dynamics (Figure 2) are formulated using the state variables longitudinal v_x and lateral velocity v_y in vehicle coordinates and the yaw rate $\dot{\psi}$. Following first principles modelling using the vehicle mass m, the vehicle inertia J, the longitudinal forces F_x, the lateral forces F_y and the torques T resulting from them, the corresponding differential equations are

$$\dot{v}_x = \frac{1}{m}\sum F_x + \dot{\psi} v_y \ , \tag{1a}$$

$$\dot{v}_y = \frac{1}{m}\sum F_y - \dot{\psi} v_x \ , \text{ and} \tag{1b}$$

$$\ddot{\psi} = \frac{1}{J}\sum T. \tag{1c}$$

Using the lateral tire force characteristics for the front $F_{y,\mathrm{F}}(\alpha_\mathrm{F})$ and the rear axle $F_{y,\mathrm{R}}(\alpha_\mathrm{R})$ depending on the tire side slip angles α_F and α_R, the full nonlinear single track model can be formulated as

$$\dot{v}_x = \frac{1}{m}\left[F_x - F_{y,\mathrm{F}}(\alpha_\mathrm{F})\sin(\delta)\right] + \dot{\psi} v_y \ , \tag{2a}$$

$$\dot{v}_y = \frac{1}{m}\left[F_{y,\mathrm{F}}(\alpha_\mathrm{F})\cos(\delta) + F_{y,\mathrm{R}}(\alpha_\mathrm{R})\right] - \dot{\psi} v_x \ , \text{ and} \tag{2b}$$

$$\ddot{\psi} = \frac{1}{J}\left[F_{y,\mathrm{F}}(\alpha_\mathrm{F})\cos(\delta) l_\mathrm{f} - F_{y,\mathrm{R}}(\alpha_\mathrm{R}) l_\mathrm{r}\right]. \tag{2c}$$

Details on the derivation can be found in Milliken and Milliken (1996). We represent the lateral tire force characteristics by a basic four coefficient Pacjeka model as described in Pacejka (2006).

The autonomous driving task requires the tracking of trajectories in inertial coordinates. The relationship between the vehicle dynamic states and the movement in global coordinate frame can be described via

$$\dot{q}_1 = \cos(\psi) v_x - \sin(\psi) v_y \text{ and} \tag{3a}$$

$$\dot{q}_2 = \sin(\psi) v_x + \cos(\psi) v_y, \tag{3b}$$

with the east coordinate q_1, the north coordinate q_2 and the vehicle heading ψ obtained by integration of the yaw rate $\dot{\psi}$.

3.2 State estimation

In general, the stochastic framework leads to intuitive implementations of state estimation algorithms. Its most common form is the Kalman Filter, which can be used to construct an estimator for linear discrete time dynamic systems of the form

$$x(k+1) = Ax(k) + Bu(k) + w(k) \tag{4a}$$

$$y(k) = Cx(k) + v(k), \tag{4b}$$

with the state vector $x \in \mathbb{R}^n$, the input vector $u \in \mathbb{R}^p$ and the measurement vector $y \in \mathbb{R}^m$. The matrices A, B and C are defined to have appropriate dimensions. The process noise $w(k) \in \mathbb{R}^n$ and the measurement noise $v(k) \in \mathbb{R}^m$ are assumed to have zero mean and to be normally distributed and uncorrelated over time (white noise). Their covariance matrices are denoted with Q for the process noise and R for the measurement noise.

The following overview is based upon the derivations given in Gelb et al. (1974) and Simon (2006). At the core of the Kalman filter algorithm is the propagation of mean and covariance of the state estimate. We will denote the maximum likelihood estimate, which is identical to the mean value due to the gaussian properties of the underlying random variables, by $\hat{x}$.

Prediction Based on the information up to time k we obtain the best possible estimate for time $k+1$ by applying the system model equation. We denote this by

$$\hat{x}(k+1|k) = A\hat{x}(k|k) + Bu(k), \tag{5}$$

where $(k+1|k)$ denotes the prediction for the time instance $k+1$ based on the information available up to the time instance k. Accordingly, $(k|k)$ describes the corrected state estimate based on the information available at time k. Using the definition of covariance and the underying random process (4), we obtain the covariance of the prediction

$$\Sigma_{xx}(k+1|k) = A\Sigma_{xx}(k|k)A^T + Q, \tag{6}$$

where Q describes the process covariance for a one-step update.

Correction The second ingredient of the algorithm is to update the prediction based on the given measurements. This forms the stabilizing mechanism of the estimator. If the system is observable for the pair (C, A), it also guarantees that it converges to the true value. Note that this is only holds under the assumption of zero-mean measurement noise. This fact can be of significant practical relevance due to sensor calibration errors. Based on the prediction step, the measurement residual

$$r(k+1) = y_\mathrm{m}(k+1) - C\hat{x}(k+1|k) \tag{7}$$

is calculated. We denote the actual measured value by y_m. Based on the statistical properties of the system, one can calculate the Kalman Gain matrix

$$K = \Sigma_{xx}(k+1|k)C^T(R + C\Sigma_{xx}(k+1|k)C^T)^{-1}, \tag{8}$$

such that the overall estimator is optimal in terms of minimum variance. It remains to update the predicted state estimate

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + Kr(k+1). \tag{9}$$

Due to the nonlinearity of the state equations (2) and (3), the presented algorithm cannot be applied directly. Several modifications are available in the literature (Simon (2006)); most widely known the Extended Kalman Filter (EKF). It relies either on analytic or numeric linearization. The algorithm itself is conceptually similar to the alternating process between prediction and correction described in this section. This follows from the fact that, assuming that they are exact at the given time instant, the equations can be rewritten using time-varying matrices. It follows, that it is a necessary condition for stable operation that the estimate is properly initialized and stays sufficiently close to the real value.

Table 1. *DevBot* Sensor Configuration

Sensor	Type
Accelerometer	McLaren Applied Technologies 3-axis
Gyroscope	McLaren Applied Technologies 1-axis
Wheelspeeds	McLaren Applied Technologies
Optical speed	Kistler Correvit SF-II
GPS	OXTS 4002
LIDAR	Ibeo Scala B2

3.3 LIDAR Localization

The localization software utilizes the Adaptive Monte Carlo Localization (AMCL) algorithm presented in Thrun et al. (2005) for localization. It is conceptually similar to the Kalman Filter, but it implements a Particle Filter, which is based on a sample based representation of the underlying stochastic distribution. Based on the point clouds from the LIDAR sensors, the vehicle corrects its position by comparing the measurements to the fixed map. To increase the performance in featureless spaces, such as long straights, a motion model is used. The algorithm assumes the track map to be constant, which improves robustness and computational efficiency compared to a SLAM algorithm in a fixed environment. The map is obtained in advance of the race during a slow manual driving lap. Details of the applied concept are presented in Stahl et al. (2018).

4. SYSTEM ARCHITECTURE

The sensor fusion task for the localization and vehicle dynamics of the *DevBot* concentrates on the following points:

- Providing a well-structured framework to increase localization accuracy by combining different data sources
- Reducing the measurement noise (especially of longitudinal and lateral velocities) to improve the control performance
- Handling the asynchronous and delayed timing characteristics of the LIDAR based localization

The relevant sensors of the *DevBot* are listed in Table 1. To achieve a modular design, it was important to keep the localization pipelines itself indepenent of each other. Each of them outputs a pose estimate which is then finally fused with the odometry obtained from the IMU and the optical speed sensor as depicted in Figure 3. The latter is preferred to the wheelspeeds as they cannot measure the slip-free velocity in an all-wheel drive powertrain. The odometry required by the LIDAR localization is purely based on optical speed sensor measurements and is not prefused with any other sensor.

The components of the system are distributed onto both available computation units. The Drive PX2 is better suited for LIDAR data processing due to the large data amounts and longer processing times. The real-time capabilitites of the Speedgoat allow for incorporating high-frequency sensor updates and achieving a smooth interpolation between the localization updates. This improves the control accuracy and enables high frequency feedback control.

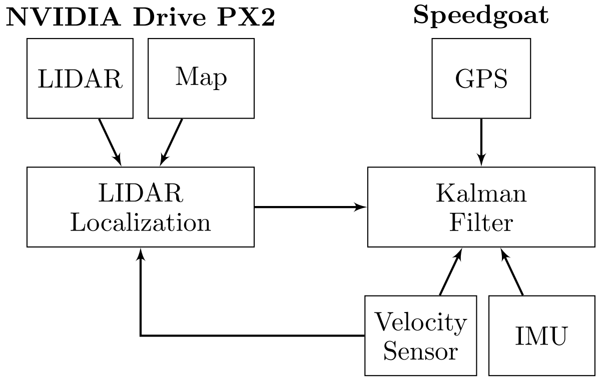


Fig. 3. State estimation architecture with both processing units. The communciation between these two is implemented via UDP.

5. KALMAN FILTER DESIGN

5.1 Model Choice

Comparing the state estimation and filtering approaches presented above, one can observe that the kinematic and the dynamic models are both exact in terms of mean value prediction capabilities (compare (1) and (2)). In practical applications, the estimation performance is highly dependent on the match between the physical system and the differential equation model. This requires that all parameters used in the model must be known exactly or must be estimated during operation. The latter requires precise sensors and slows the transient response times of the estimation. In contrast, the kinematic approach uses measured acceleration values instead of any model parameter to calculate the prediction. This is beneficial in the racing application, since the model parameters will never be exact on different tracks or varying weather conditions. It thus promises a consistent and predictable system performance.

We will compare both approaches in the following, using a nonlinear single track model (STM) and a kinematic point-mass model with acceleration measurements (PM) as inputs. GPS and LIDAR based localization are unlikely to exhibit random noise but can show significant systematic measurement error. In the case of GPS, this can be caused by undesired reflections of the signals from the environment. In case of LIDAR, this can be caused by mapping errors. Since the resulting estimation errors increase significantly with non-zero mean measurement errors, this could have negative impact on the precision of the velocity state estimates. To overcome these drawbacks, additionally a modified version of the point-mass model (MPM) is proposed. It uses a cascaded Kalman Filter with separated velocity and position dynamics. The former uses the IMU measurements as system inputs and (1) as a system model, while the latter takes the fused velocities and the yaw rate as inputs and uses (3) as the underlying system model. This separation prevents the measurement update triggered by the localization from affecting the velocity estimates. However, this setup strictly requires that the optical wheelspeed sensor is available since the

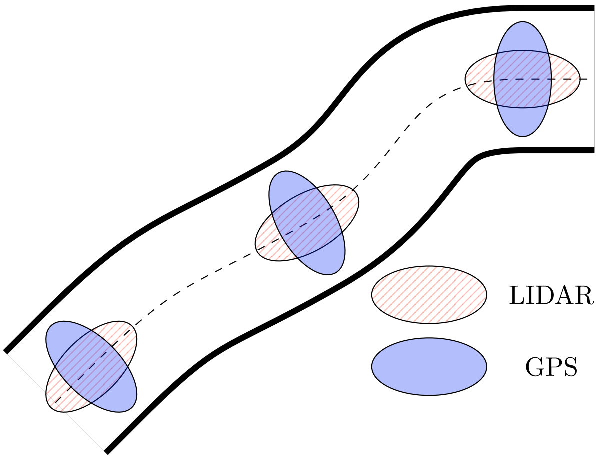


Fig. 4. Visualization of the transformation applied to the covariance matrices of the localization measurements. The resulting uncertainty ellipses are oriented along the track and depicted for three different vehicle positions. The LIDAR covariance in the lateral direction is smaller than the GPS covariance. Therefore, the LIDAR is weighted more strongly in lateral direction. the GPS is weighted more strongly in longitudinal direction.

velocity subsystem becomes otherwise unobservable. Furthermore, it should be noted that both point-mass models are not capable of filtering the yaw rate as no derivative information can be obtained from the current sensor setup.

5.2 Covariance Adaption

Even though the GPS platform used is highly accurate under normal conditions, its absolute performance may be unsufficient dependending on the environment condition. The LIDAR localization was found to be able to locate the vehicle within 20 cm of lateral deviation. However, it struggles to generate exact longitudinal measurements in featureless spaces such as long straights. A small, but important difference between the two systems is, that the LIDARs are positioning the vehicle relative to its obstacles while the GPS is referring to a global coordinate system. In case there are map errors, the former is more robust since it guarantees that the vehicle will stay within the planned safety margin.

The Kalman Filter approach provides a well-suited tool to incorporate the knowledge we have about the individual sensor accuracy. We assign the task of lateral estimation strongly to the LIDAR sensor, while the GPS is taking care of the longitudinal position as depicted in Figure 4. The measurement covariance matrix for each sensor is rotated according to the track orientation ψ_{T} by

$$R_{\mathrm{loc}} = T(\psi_{\mathrm{T}}) \begin{bmatrix} \sigma_{\mathrm{long}} & 0 \\ 0 & \sigma_{\mathrm{lat}} \end{bmatrix} T^{\mathrm{T}}(\psi_{\mathrm{T}}) \tag{10}$$

with

$$T(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{11}$$

being the standard rotation matrix.

5.3 Incorporation of Asynchronous Measurements

The Kalman Filter is propagating mean and covariance of its estimates for every step according to the underlying process and measurement model. This allows time-varying models to be easily incorporated through modification of the corresponding equations. The fact that the localization measurements are available asynchronously is therefore addressed by the use of different measurement vectors. Depending on the available measurement signals, it can be written as:

$$y_{\mathrm{M}} = \begin{cases} \begin{bmatrix} x_{\mathrm{G}} \; y_{\mathrm{G}} \; \psi_{\mathrm{G}} \; v_x \; v_y \end{bmatrix}^{\mathrm{T}}, \\ \text{if only GPS available} \\ \begin{bmatrix} x_{\mathrm{L}} \; y_{\mathrm{L}} \; \psi_{\mathrm{L}} \; v_x \; v_y \end{bmatrix}^{\mathrm{T}}, \\ \text{if only LIDAR available} \\ \begin{bmatrix} x_{\mathrm{G}} \; y_{\mathrm{G}} \; \psi_{\mathrm{G}} \; x_{\mathrm{L}} \; y_{\mathrm{L}} \; \psi_{\mathrm{L}} \; v_x \; v_y \end{bmatrix}^{\mathrm{T}}, \\ \text{if LIDAR \& GPS available} \\ \begin{bmatrix} v_x \; v_y \end{bmatrix}^{\mathrm{T}}, \text{ else} \end{cases} \tag{12}$$

5.4 Time Delay Compensation

Since the processing of LIDAR measurements takes some time, it was found that there is a significant delay of approximately 40 ms between the LIDAR and GPS position signal. This can be compensated for by obtaining an odometry estimate from the measured longitudinal and lateral velocities and the yaw rate. It is calculated by forward integration of (3). The correction itself is then taken from

$$q_{1,\mathrm{comp}}(t) = q_{1,\mathrm{odom}}(t) - q_{1,\mathrm{odom}}(t - 40\,\mathrm{ms}) \tag{13a}$$
$$q_{2,\mathrm{comp}}(t) = q_{2,\mathrm{odom}}(t) - q_{2,\mathrm{odom}}(t - 40\,\mathrm{ms}) \tag{13b}$$
$$\psi_{\mathrm{comp}}(t) = \psi_{\mathrm{odom}}(t) - \psi_{\mathrm{odom}}(t - 40\,\mathrm{ms}). \tag{13c}$$

In fact, the compensation values measure the difference between the purely forward integrated vehicle poses at the actual time and the time when the LIDAR measurement is captured. $q_{1,\mathrm{comp}}(t)$, $q_{2,\mathrm{comp}}(t)$ and $\psi_{\mathrm{comp}}(t)$ is an estimate of the vehicle motion between these two points in time.

6. EXPERIMENTAL RESULTS

The filter structure proposed in this paper was tested in the full-size race car *DevBot* on an airfield. The data was collected in several runs with speeds up to $150\,\mathrm{km\,h^{-1}}$ and acceleration up to $8.5\,\mathrm{m\,s^{-2}}$ in longitudinal and lateral direction for the trajectory planning algorithm. This corresponds roughly to 80 % of the maximum achievable acceleration in steady state cornering. In the following we analyze a 20 s segment, consisting of high speed driving as well as cornering situations. The evaluation is strongly based on the measurement residuals as no ground truth data is available. The residuals capture the mismatch between the model-based prediction and the measured value. A precise definition can be found in (7). The point-mass models utilize an Extended Kalman Filter with analytic derivatives for linearization. The STM is implemented based on an Extended Kalman Filter with numerically obtained derivatives since they cannot be derived trivially as in the point-mass case. Both are discretized using an Euler-Forward scheme with a sample rate of 4 ms.

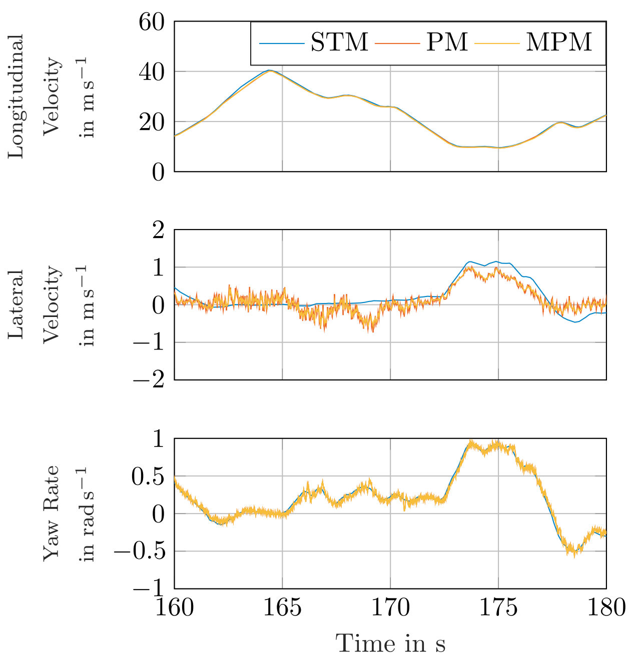


Fig. 5. Estimates of the dynamic states of the vehicle using different process models. The single track model (STM) shows significant offset with respect to point mass models, while the point mass model (PM) and the modified point mass (MPM) model exhibit stronger noise. This effect can be related to the stronger model assumptions of the STM.

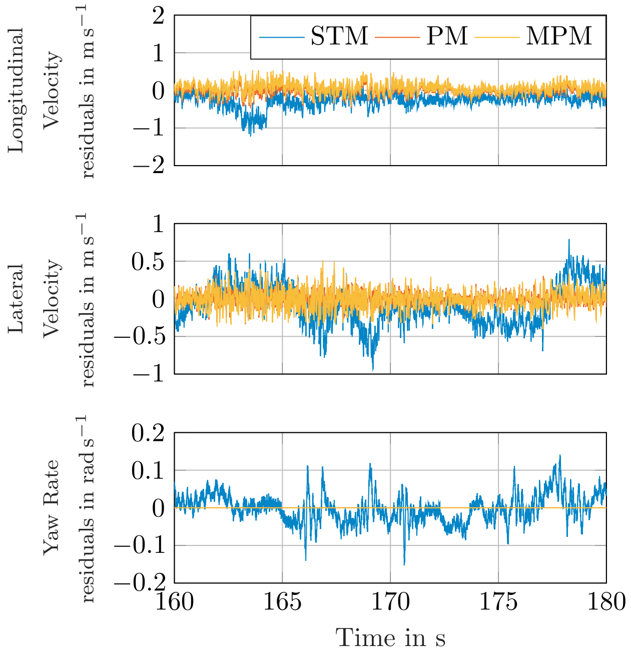


Fig. 6. Residuals of the dynamic state measurements of the vehicle using all available sensors. The single track model (STM) produces non-zero mean residuals which is an indicator for system/model mismatch. In comparison, the point mass model (PM) and the modified point mass model (MPM) perform better. The yaw rate is only estimated by the (STM).

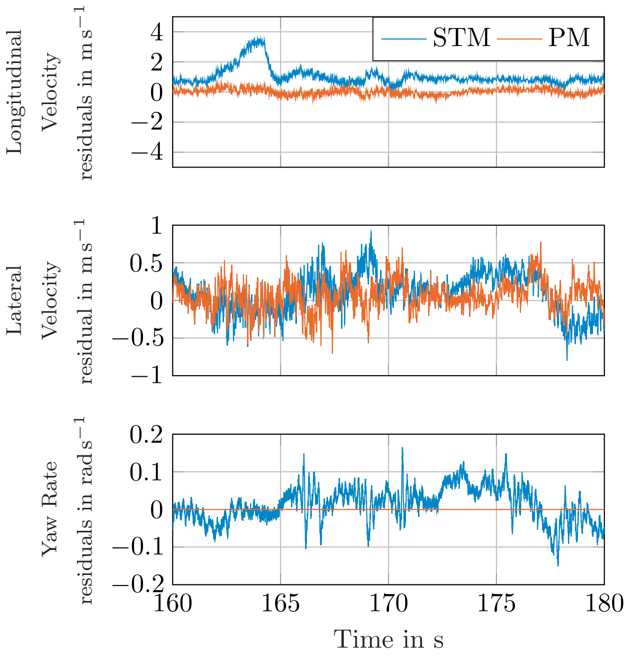


Fig. 7. Residuals of the dynamic state measurements of the vehicle with the optical velocity sensor not available. Therefore the modified point mass model suffers from a loss of observability and could not be compared. The point-mass model significantly outperforms the nonlinear vehicle model in this setting.

We will discuss two cases: All sensors are available in the first case. Figure 5 shows the state estimation performance for all measurements available of the three different structures discussed before. While the PM and the cascaded version MPM show comparable performance, the high-fidelity STM shows better noise rejection properties but also a systematic estimation offset. This is undermined by the residuals depicted in Figure 6, which are non-zero mean for all states estimates of the STM.

In the second case, we assume the optical speed sensor to be unavailable. This case is interesting because the optical velocity sensor is not redundant in our current setup and is also seldom available in other vehicle configurations. The cascaded point mass filter could not be utilized in this case, since the design system becomes unobservable. Figure 7 depicts the residuals of the other models. Note that the velocity residuals were not used for estimation purposes, rather only for validation. In this scenario, the point mass model significantly outperforms the nonlinear single track model, especially for the longitudinal velocity. This mismatch results from the fact that the drivetrain model quality is not sufficient at high speeds. However, the LIDAR odometry is still provided by the optical velocity sensor in this setup. This could be replaced by an odometry based on wheelspeed and IMU sensors to achieve at least a descent prediction quality.

As the last evaluation step, the residuals of the GPS and the LIDAR based position measurement are depicted in Figure 8 for the MPM case with all sensors available. Whereas the residuals of the GPS are small in the longitudinal direction, the LIDAR localization fails to achieve sufficient performance. This is related to systematic errors in

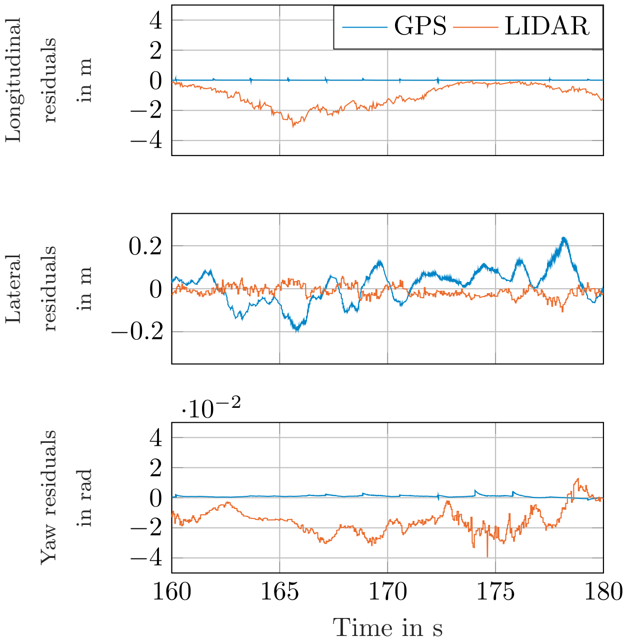


Fig. 8. Residuals of GPS and LIDAR localization measurements of the modified point mass Kalman Filter with optical speed sensor available. The estimate relies heavily on GPS in the longitudinal direction. In the lateral direction the LIDAR measurement is weighted more strongly. Small peaks in GPS residuals are related to a minor timing difference between multiple computation units.

the odometry used combined with the sparse availability of correction features in this direction. The LIDAR is favored over the GPS in lateral direction, which can be seen from the fact that the LIDAR residuals are much smaller than the GPS residuals in the lateral direction. Further, Figure 9 shows the benefical effects of the delay compensation and the covariance adaption. Whereas the former decreases the residuals of the LIDAR position measurement in the longitudinal direction significantly, the latter ensures that the weighting between GPS and the LIDAR is performed as intended. This can be seen from the fact that the GPS longitudinal residuals increase significantly for the case without covariance adaption (NCA). The changes for the lateral localization precision are rather small, but there is an increase in lateral LIDAR residuals for the NCA case. This shows that the lateral LIDAR measurement is weighted more strongly in the case with covariance adaption and that a possible GPS outlier would not disturb the final localization significantly.

7. CONCLUSION

In this paper, we have proposed a novel system architecture for localization and state estimation of autonomous vehicles which proved to be robust and reliable in real-world conditions even in the nonlinear driving range. We evaluated the usability of two purely kinematic models for dynamic state estimation of a race car. Interestingly, they were able to outperform a high-fidelity nonlinear single track model. This is related to the fact that the point-mass model is exact for the whole region of operation of the tires and at high speeds. The main advantage is that

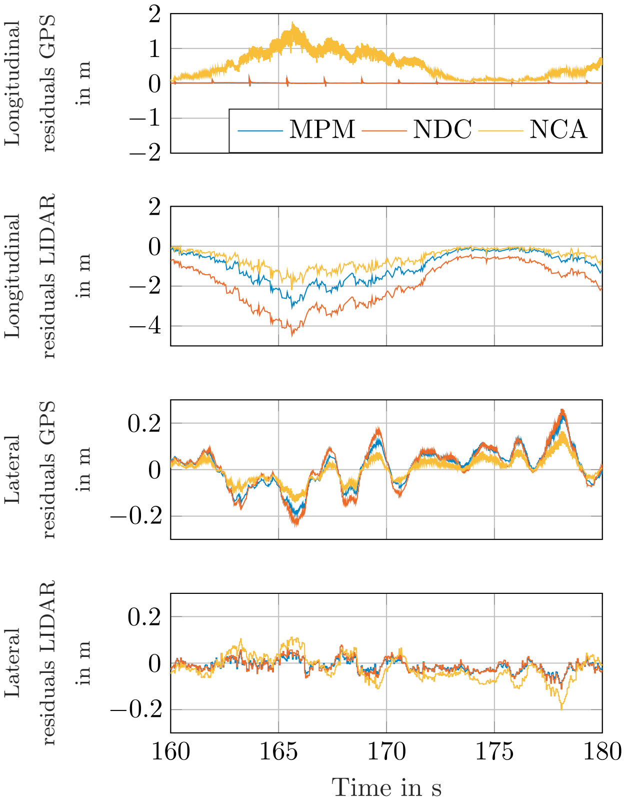


Fig. 9. Residuals of GPS and LIDAR localization measurements of the modified point mass Kalman Filter with different modifications. Whereas the version without the proposed covariance adaption (NCA) suffers from severe GPS residuals in longitudinal direction, the standard version (MPM) weights LIDAR and GPS as expected. This can be seen from small GPS longitudinal and small LIDAR lateral residuals. Without delay compensation (NDC), the longitudinal residuals for the LIDAR increase.

the estimation residuals are zero mean which is important for the design of a high-performance tracking controller. However, this comes at the cost of a slight increase in estimation noise. A positive side-effect is the much lower computational cost and improved robustness due to model simplicity.

Future research will be devoted to automatic reconfiguration of the sensor fusion system in case of sensor failures, e.g. a GPS dropout. The presented sensor fusion implementation itself is capable of handling this already once a failure is known, however the difficulty is the robust detection of those occurences. Furthermore, it is desirable to include camera localization measurements in order to avoid depending on GPS as a secondary localization mechanism at all.

ACKNOWLEDGMENT AND CONTRIBUTIONS

We would like to thank Roborace for the opportunity of evaluating our algorithms on their prototype as well as for their support during the testing sessions and the Berlin event. Further, we thank Madeline Wolz for her support during implementation of the different filter variants and

Johannes Strohm, Mikhail Pak and Leon Sievers for discussions and revision of the paper.

Alexander Wischnewski contributed to the overall system architecture as well as the different estimator designs. Tim Stahl contributed to the design of the visual localization algorithms and the network communication. Johannes Betz and Boris Lohmann contributed equally to the conception of the research project and revised the paper critically for important intellectual content.

REFERENCES

Antonov, S., Fehn, A., and Kugi, A. (2011). Unscented Kalman filter for vehicle state estimation. *Vehicle System Dynamics*, 49(9), 1497–1520.

Betz, J., Wischnewski, A., Heilmeier, A., Nobis, F., Stahl, T., Hermansdorfer, L., Lohmann, B., and Lienkamp, M. (2018). What can we learn from autonomous level-5 Motorsport? *Proceedings of chassis.tech 2018*.

Bonnifait, P., Bouron, P., Crubille, P., and Meizel, D. (2001). Data fusion of four ABS sensors and GPS for an enhanced localization of car-like vehicles. *Proceedings of 2001 IEEE International Conference on Robotics and Automation*, 1597–1602.

Bresson, G., Rahal, M.C., Gruyer, D., Revilloud, M., and Alsayed, Z. (2016). A cooperative fusion architecture for robust localization: Application to autonomous driving. *Proceedings of 2016 IEEE 19th Conference on Intelligent Transportation Systems*, 859–866.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Reid, I., Leonard, J.J., Neira, J., Reid, I., and Leonard, J.J. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6), 1309–1332.

Du, H. and Li, W. (2014). Kinematics-based parameter-varying observer design for sideslip angle estimation. *Proceedings of 2014 International Conference on Mechatronics and Control*, 2042–2047.

Farrelly, J. and Wellstead, P. (1996). Estimation of Vehicle Lateral Velocity. *Proceedings of the 1996 IEEE International Conference on Control Applications*, 552–557.

Gelb, A., Kasper, J.F., Nash, R.A., Price, C.F., and Sutherland, A.A. (eds.) (1974). *Applied Optimal Estimation*. MIT Press, Cambridge, MA.

Guo, H., Cao, D., Chen, H., Lv, C., Wang, H., and Yang, S. (2018). Vehicle dynamic state estimation: State of the art schemes and perspectives. *IEEE/CAA Journal of Automatica Sinica*, 5(2), 418–431.

Haiyan, Z. and Hong, C. (2006). Estimation of vehicle yaw rate and side slip angle using moving horizon strategy. *Proceedings of 2006 6th World Congress on Intelligent Control and Automation*, 1828–1832.

Jang, S., Ahn, K., Lee, J., and Kang, Y. (2015). A study on integration of particle filter and dead reckoning for efficient localization of automated guided vehicles. In *2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, 81–86.

Jeon, S. (2010). State Estimation Based on Kinematic Models Considering Characteristics of Sensors. *Proceedings of the 2010 American Control Conference*, 640–645.

Kang, C.M., Lee, S.H., and Chung, C.C. (2014). Lane estimation using a vehicle kinematic lateral motion model under clothoidal road constraints. *Proceedings of 17th IEEE International Conference on Intelligent Transportation Systems*, 1066–1071.

Kuutti, S., Fallah, S., Katsaros, K., Dianati, M., Mccullough, F., and Mouzakitis, A. (2018). A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications. *IEEE Internet of Things Journal*, 5(2), 829–846.

Milliken, W.F. and Milliken, D.L. (1996). *Race Car Vehicle Dynamics*. Society of Automotive Engineers Inc., Great Britain.

Mur-Artal, R., Montiel, J.M., and Tardos, J.D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5), 1147–1163.

Pacejka, H. (2006). *Tire and Vehicle Dynamics*. SAE International.

Simon, D. (2006). *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, New York, NY, USA.

Stahl, T., Wischnewski, A., Betz, J., and Lienkamp, M. (2018). Ros-based localization of a race vehicle at high-speed using lidar. In *7th International Conference on Mechatronics and Control Engineering 2018*.

Suhr, J.K., Jang, J., Min, D., and Jung, H.G. (2017). Sensor fusion-based low-cost vehicle localization system for complex urban environments. *IEEE Transactions on Intelligent Transportation Systems*, 18(5), 1078–1086.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.

Trehard, G., Pollard, E., Bradai, B., and Nashashibi, F. (2015). On line mapping and global positioning for autonomous driving in urban environment based on evidential SLAM. *Proceedings of 2015 IEEE Intelligent Vehicles Symposium*, 814–819.

Usenko, V., Engel, J., Stuckler, J., and Cremers, D. (2015). Reconstructing Street-Scenes in Real-Time from a Driving Car. *Proceedings of International Conference on 3D Vision 2015*, 607–614.

Wenzel, T., Burnham, K., Blundell, M., and Williams, R. (2006). Dual extended Kalman filter for vehicle state and parameter estimation. *Vehicle System Dynamics*, 44(2), 153–171.

Wielitzka, M., Dagen, M., and Ortmaier, T. (2014). State Estimation of Vehicle's Lateral Dynamics using Unscented Kalman Filter. *Proceedings of 53rd IEEE Conference on Decision and Control*, 5015–5020.

Zhao, L.H., Liu, Z.Y., and Chen, H. (2011). Design of a nonlinear observer for vehicle velocity estimation and experiments. *IEEE Transactions on Control Systems Technology*, 19(3), 664–672.

7.2 Tube Model Predictive Control for an Autonomous Racecar

Contributions: Alexander Wischnewski, the author of this dissertation, and Martin Euler designed and implemented the presented robust control algorithm. They are further responsible for the analysis of the system performance and parameter studies. Salih Gümüs is responsible for the real-time embedded MPC scheme. Boris Lohmann contributed to the critical revision of the manuscript and the conception of the research project.

ARTICLE TEMPLATE

Tube Model Predictive Control for an Autonomous Race Car

A. Wischnewski[a,*], M. Euler[a,*], S. Gümüs[a] and B. Lohmann[a]

[a]Chair of Automatic Control, Technical University of Munich, Munich, Germany
[*]The authors contributed equally.

ARTICLE HISTORY
Compiled June 9, 2021

ABSTRACT
Nonlinear effects and external disturbances can severely impact the path and velocity tracking control of an autonomous race car at the handling limits. State-of-the-art approaches do not take information about these uncertainties explicitly into account in the design process and are therefore prone to failure. To overcome this limitation, we present a robust control design based on tube model predictive control (TMPC). It is based on a simplified friction limited point-mass model and an additive disturbance for the lateral and longitudinal dynamics. Instead of nominal predictions, it leverages an approximate tube of reachable sets over the prediction horizon using an ellipsoidal set representation to guarantee constraint satisfaction. The resulting optimisation problem can be posed in the form of a standard quadratic program by tightening the input and state constraints of the nominal model predictive control problem appropriately. The computational burden is therefore the same as in the nominal case. We benchmark our controller on a Hardware-in-the-Loop testbench with a nonlinear dual-track model and a combined Pacejka tyre model. The results demonstrate that the TMPC controller reduces the number and severeness of constraint violations while achieving comparable lap-times in contrast to an MPC controller and an infinite time LQR controller. It manages to apply caution when needed while maintaining a similar level of performance and is therefore considered to be superior in practical applications.

KEYWORDS
Robust control; Model predictive control; Tube MPC; autonomous driving; race car

1. Introduction

1.1. Motivation

In recent years, there have been significant contributions to the domain of autonomous driving. Systems like lane-keeping assist and adaptive cruise control are nowadays wide-spread in state-of-the-art production vehicles. However, further advances within autonomous driving require profound and reliable vehicle control strategies to master also the most demanding scenarios.

The application of vehicle control algorithms to race vehicles turned out to be a perfect research challenge to advance the technology tackled by several research groups [1–4]. Compared to urban driving, the racetrack is much more structured and allows for several simplifications during software development. On the other hand, the vehicle

CONTACT A. Wischnewski. Email: alexander.wischnewski@tum.de

control complexity increases, as the vehicle drives at the handling limits with complex nonlinear dynamics. The high uncertainty of these models (e.g. due to the empirically estimated tyre parameters) and the sensitivity to external disturbances (e.g. wind or inclination) motivate the question whether more sophisticated design techniques from the field of robust control can lead to a reduction of the safety margins usually applied during operation of autonomous racing algorithms.

1.2. Scope & outline of the paper

This paper will investigate the possible performance gains from application of a robust model predictive controller (RMPC) based on tube model predictive control (TMPC) for autonomous racing. It is capable of tracking a target trajectory (consisting of a path and a velocity profile) and is integrated into the autonomous racing software framework developed by the team of the Technical University of Munich (TUM). The latter has been successfully applied during several real-world experiments on full-scale race cars within the Roborace competition [5], reaching lap-times close to amateur racing drivers. The planning algorithm presented in [6] is used to generate the target trajectories. Our key contributions are the efficient formulation of the dynamics model such that the resulting RMPC problem is feasible in real-time on a state-of-the-art rapid control prototyping unit and a thorough comparison of the TMPC with a nominal MPC and an LQR controller (similar to the state-feedback controller presented and applied within the Roborace competition by [1]). We demonstrate that the additional information about the model uncertainties leads to a controller which is cautious when necessary but drives aggressively in other situations in detailed Hardware-in-the-Loop simulation studies. This behaviour leads to a significant reduction in the number of tyre and lateral error constraint violations while achieving comparable lap-times in contrast to the LQR and the nominal MPC solution.

The rest of the paper is structured as follows: related work within the field of motion control of autonomous vehicles and robust model predictive control is presented in the next section. The methodology section introduces the applied TMPC concept as well as the vehicle model and the computationally efficient formulation of the moving horizon optimal control problem. The paper concludes with a thorough parameter study and a discussion on the advantages and disadvantages of the TMPC in comparison to the nominal MPC and LQR control techniques.

2. Related work

2.1. Vehicle control

Several approaches for autonomous vehicle software design split the planning and control task into three major components [1,7–11]: a *global trajectory planner* calculates a trajectory from a start to an endpoint. In the case of a race vehicle, the global trajectory is the raceline for the current track. Furthermore, a *local trajectory planner* generates a detailed trajectory for the current road section by considering road limits and (possibly dynamic) objects within the scene. Finally, a *tracking controller* is used to follow the local trajectory by calculation of actuator commands such as steering, brake, and throttle set points.

Series production vehicles widely use models based on decoupled lateral and longitudinal dynamics for trajectory tracking, like adaptive cruise control and lane-keeping assistants. Due to the more complex dynamics, lateral control receives more attention in research activities and will be reviewed in the following. For the lateral control, linear output feedback controllers [1,3,11,12], flatness-based control [13,14], sliding-mode [15–18] and optimisation-based methods, like MPC, are the most common [2,19–21]. A key disadvantage of output feedback, flatness-based, and sliding-mode controllers is that these methods do not explicitly consider state and input constraints and therefore do not guarantee constraint satisfaction. Furthermore, Calzolari et al. [21] show that sliding mode controllers can be aggressive and result in saturated tyre forces, whereas flatness-based control is sensitive to model quality. In contrast, MPC controllers use predictions of the future vehicle behaviour to calculate inputs based on the measured state and upcoming constraints. When driving at the handling limits, this approach is advantageous in reducing the risk of an accident caused by leaving the specified domain of operation described by the input and state constraints. Therefore, we will concentrate on the review of MPC-based controllers in the following.

Namely, MPC concepts predict future vehicle states based on the current state and a system model. By demanding that the predicted states and inputs satisfy the constraints, it is possible to generate input sequences that achieve the desired behaviour while maintaining optimality with respect to a (usually quadratic) cost function. Various versions of MPC controllers for autonomous vehicles have been presented in the literature [2,7,22–27]. In their approaches, Funke et al. [23] and Brown et al. [24] show that an MPC controller is capable of tracking a trajectory and avoiding obstacles while respecting the vehicle dynamics limitations. Falcone et al. [7] track a trajectory on a slippery road using a nonlinear MPC formulation. While these approaches are used for passenger vehicles, Anderson et al. [25] and Subosits et al. [26] show that MPC is suitable for racing vehicles. Complex nonlinear dynamics at the handling limits inspire the use of more profound models for racing vehicles. Hence, Liniger et al. [2] show that a nonlinear MPC can be implemented by subsequent linearisation at each sampling step in the form of a quadratic program (QP) and successfully drives 1:43 scale race cars. A similar vehicle dynamic model, based on nonlinear tyre characteristics, is applied for the control of a self-driving formula student vehicle in [28] and combined with a learning-based approach in [29]. A different approach, based on repetitive learning control and therefore exploiting the nature of circuit driving, is presented in [30]. Another possible direction for solving the nonlinear problem is presented by Alcala et al. [27] by using a linear parameter varying MPC that takes a combination of linear models to approximate a nonlinear model without calculating an explicit linearisation for the control of a 1:10 RC car.

One drawback of the above approaches and, in general, nominal MPC is that model errors and external disturbances lead to a deviation between the actual and predicted behaviour. As a result, these schemes cannot guarantee recursive feasibility [31]. Using slack variables allows feasible solutions for the numerical optimisation problem to be recovered, but the closed-loop dynamics will still violate the a priori specified constraints when driving at or close to the limits due to the more complex dynamics. Hence, a robust version of MPC, taking into account uncertainties and external disturbances, is required to satisfy the constraints. Alsterda et al. [32] suggest a robust control method based on contingency model predictive control. The controller tracks a target trajectory while simultaneously calculating a feasible trajectory for emergencies like driving on icy surfaces. Thus, robustness increases, and it is possible to react appropriately to upcoming or unpredicted critical scenarios. In contrast, Gao et al. [33] use a robust nonlinear model predictive controller for the lateral control of a vehicle, which avoids obstacles and tracks the desired trajectory. A robust invariant set tightens the constraints and guarantees the satisfaction of state and input constraints. Another tube model predictive control (TMPC) is presented by Sakhdari et al. [34] for adaptive cruise control. In their work, the controller considers radar measurement errors, aero-dynamical, and rolling drag when controlling the ego vehicle's distance to other road users. In the lateral direction, Rathai et al. [35] show that a TMPC controller can stabilise a car within a lane under the influence of disturbances.

2.2. Robust model predictive control

It is well-understood that disturbances and model errors can drive the closed-loop system to a state where the receding horizon optimal control problem becomes infeasible when using a nominal MPC controller for a task with state constraints [31,36]. One of the first approaches to address this problem for the design of an MPC controller was min-max MPC, proposed by Campo and Morari in 1987 [37]. In this case, the idea is to optimise the control problem for the worst-case disturbances. This approach increases the robustness, but there are challenges due to computational load and high conservativeness [38,39]. Scokaert and Mayne [40] reduce the computational load by using a terminal set, but the conservative behaviour remains.

In recent years, the focus of the research community has shifted towards tube model predictive control (TMPC) and stochastic model predictive control (SMPC) for controlling uncertain systems [41]. Both use a pre-defined feedback control law instead of the raw open-loop inputs for the formulation of the receding horizon optimal control problem. This allows the construction of set-valued predictions for the optimisation horizon. In the case of TMPC, it is required that all predictions fulfil the constraints; in the case of SMPC only a certain percentage of the realisations have to match the specified constraints. The numerical realisation of the problems is usually done in form of a constraint-tightening scheme, where the nominal optimal control problem is solved for a modified set of input and state constraints. We want to point out that both concepts show strong similarity in the structure of the resulting optimisation problem and their key difference is therefore in the determination of the required constraint tightening. In TMPC, disturbances only need to be upper bounded, whereas SMPC requires a full probabilistic description. In practical applications, the upper bounds can be determined more easily than the probability distribution since they can be related to data from the system under operation. Furthermore, there exist various ways to construct admissible tubes around the nominal predictions for the deterministic case [31,36,42–44]. We therefore consider TMPC to be more promising for the application to the autonomous racing control task than SMPC.

Constraint satisfaction of the closed-loop can be guaranteed if the inputs and states satisfy the tighter constraints within the optimisation horizon and suitable terminal ingredients are found [31,36,42]. The tube itself is an artificial construct around the predicted states that is used to tighten the constraints of a nominal MPC formulation. Two major concepts exist (see Fig. 1): The first uses a sequence of reachable sets $\mathcal{S}_i$ to describe the tube [36,45,46]. The tube size increases over the prediction horizon, considering the rising deviation due to cumulative model errors and disturbances. However, it stays bounded as a pre-stabilising controller is considered. The second concept uses robust positively invariant (RPI) sets $\mathcal{S}$ [31] around the nominal dynamics. A set $\mathcal{S}$ is stated as RPI if all (successor) states are inside $\mathcal{S}$ for all bounded disturbances [47]. Besides, it is allowed to reoptimise the nominal prediction of the initial state as long as the measured state lies within the RPI. Different set representations are

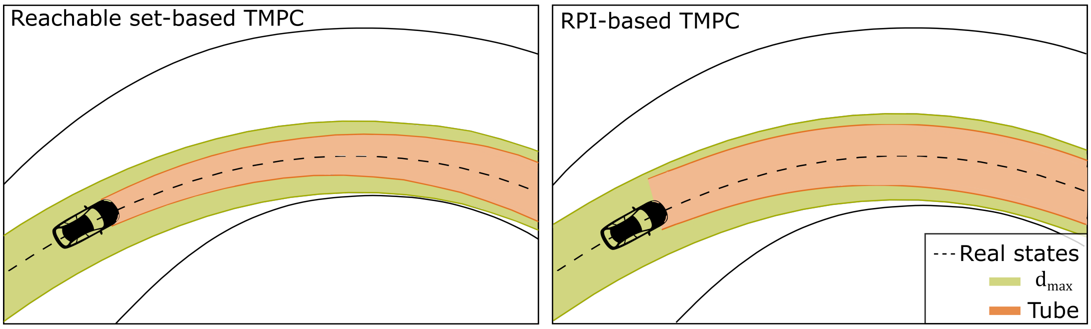


Figure 1.: Comparison of the reachable set and RPI-based TMPC approaches with the lateral deviation limit $d_{\max}$

currently used within the control community. The most common ones are polytopes [31,35,42,44,48,49], zonotopes [50–54], and ellipsoids [36,55]. Depending on the system complexity and available computational power, these types pose advantages and disadvantages in conservativeness and computational demand.

3. Methodology

3.1. Notation and preliminaries

Within this paper the sets of the tube are described as ellipsoids

$$E(p, M) := \left\{x \in \mathbb{R}^n | (x-p)^T M^{-1} (x-p) \leq 1\right\}, \tag{1}$$

with p the centre of the ellipsoid and M the shape matrix. The constraint sets are formulated as polytopes of the form

$$\mathbb{X} = \{x \in \mathbb{R}^n | H_x x \leq h_x, h_x \in \mathbb{R}^{m_x}\}, \tag{2}$$

with m_x the number of half-spaces. The Minkowski Sum for two sets $\mathcal{A}$ and $\mathcal{B}$ is defined as

$$\mathcal{A} \oplus \mathcal{B} := \{a + b | a \in \mathcal{A}, b \in \mathcal{B}\}. \tag{3}$$

The Minkowski sum of two ellipsoids is not necessarily an ellipsoid. For this reason an over-approximation by Kurzhanski [56] is used to calculate the Minkowski sum

$$E(p_1, M_1) \oplus E(p_2, M_2) \subset E\left(p_1 + p_2, \left(1 + c^{-1}\right) M_1 + (1 + c) M_2\right), \tag{4}$$

with $c = \sqrt{\mathrm{Tr}(M_1)/\mathrm{Tr}(M_2)}$ and $\mathrm{Tr}(M)$ the trace of the Matrix M. Additionally the affine transformation for ellipsoids

$$A \cdot E(p, M) + b = E(Ap + b, AMA^T) \tag{5}$$

is used to calculate the set dynamics.

3.2. Tube-based MPC

We propose a TMPC controller based on the linear dynamics reachable set approach presented by Chisci et al. [36]. We favour this approach over the RPI-based concept, as it can be solved by a standard QP solver and does not require non-linear containment constraints for the initial state [55]. Another advantage is that the nominal MPC problem can be easily extended to the robust control setting by tightening the constraints according to the calculated tube geometry. We use ellipsoids to describe the tube because the tube dynamic is straightforward to calculate and more suitable for real-time applications than polytopes or zonotopes due to the constant complexity of the set representation over the control horizon.

For a disturbed system the system dynamics are given to be

$$x_{k+1} = Ax_k + Bu_k + Dw_k, \tag{6}$$

with the state $x \in \mathbb{R}^n$, the input $u \in \mathbb{R}^m$ and the external disturbance $w \in \mathbb{R}^o$. The system matrices A, B and D are of appropriate dimensions. Instead of optimising the open-loop control input sequence u_k, TMPC employs a pre-stabilising policy

$$u_k = r_k + K(x_k - p_k), \tag{7}$$

optimises the sequence r_k, and predicts the nominal system behaviour p_k. The tube is calculated as the sequence of reachable sets $\mathcal{X}$ for the uncertain system relative to the nominal prediction. Using the set operations presented in the previous section gives the following set dynamics for the reachable sets $\mathcal{X}_k = E(p_k, M_k)$ at time-step k

$$\begin{aligned}
p_0 &= x_t, && \text{(8a)}\\
p_{k+1} &= Ap_k + Br_k, && \text{(8b)}\\
M_{k+1} &= (1 + c_k^{-1})(A + BK)M_k(A + BK)^T + (1 + c_k)D\tilde{M}D^T, && \text{(8c)}\\
c_k &= \sqrt{\mathrm{Tr}((A + BK)M_k(A + BK)^T)/\mathrm{Tr}(D\tilde{M}D^T)} && \text{(8d)}
\end{aligned}$$

with $\mathcal{D} = E(0, D\tilde{M}D^T)$ being the uncertainty ellipsoid covering the potential values of Dw_k based upon the maximum disturbance values assumed. Note that the shape matrix dynamics do not depend on the initial state nor the sequence r_k and can therefore be pre-computed ahead of the optimisation problem's formulation by application of Eq. (8c) and Eq. (8d).

The equations allow us to formulate the *receding horizon optimal control problem* using the nominal dynamics p_k and the decision variables r_k as follows:

$$\min_{r_0,\dots,r_{N_p-1}} \sum_{k=0}^{N_p-1} \left(p_k^T Q p_k + r_k^T R r_k\right) + p_{N_p}^T P p_{N_p} \tag{9a}$$

s.t.

$$p_{k+1} = A p_k + B r_k \tag{9b}$$

$$p_0 = x_t \tag{9c}$$

$$[H_x]_{i,k}\, p_k + \sqrt{[H_x]_{i,k}\, M_k\, [H_x]_{i,k}^T} \leq [h_x]_{i,k}, \forall i,\ k \tag{9d}$$

$$[H_u]_{j,k}\, r_k + \sqrt{[H_u]_{j,k}\, K M_k K^T\, [H_u]_{j,k}^T} \leq [h_u]_{j,k}\,, \forall j,\ k \tag{9e}$$

$$i \in \{1, ..., m_x\}\,,\ j \in \{1, ..., m_u\}\,,\ k \in \{1, ..., N_p\}\,, \tag{9f}$$

with x_t the measured system state. The cost function only takes the nominal dynamics into account and is similar to a standard LQR problem. Therefore, the terminal weight matrix P is chosen as the solution of the discrete time algebraic Riccati equation with the weights Q and R. The polytopic state $\mathcal{X}_k \in \mathbb{X}_k$ (Eq. (9d)) and input constraints $\mathcal{U}_k \in \mathbb{U}_k$ (Eq. (9e)) have been reformulated using the approach presented by [57]. Note that the terminal set is implicitly included for the case $k = N_p$ in the above formulation of the constraints. We also restrict our notation to the case of m_x and m_u, the number of half-spaces in the polytopic constraints, being constant for the optimisation horizon for the sake of brevity.

3.3. Vehicle model

One of the main trade-offs in modelling vehicle dynamics for control purposes is the conflict between complexity and accuracy [58]. Based on the promising results in [1, 5,26], we choose a simple friction-limited point mass model (see Figure 2) for the control design. It was shown that it is possible to achieve high-performance in racing applications, if the relation between the driven curvature and the steering angle is known and the tyre limits are not violated [1,5]. We formulate the resulting path and velocity tracking problem in a curvilinear coordinate system similar to [28] and [59]. It turns out that this formulation is close to linear and therefore leads to intuitive results when the optimal control problem is reformulated as a quadratic program. Furthermore, we emphasize that the point-mass model is representing the motion dynamics exactly as long as we assume the lateral and longitudinal acceleration are perfectly tracked by the underlying low-level controllers (see Section 3.4). This is an advantage in comparison to other vehicle models based on linear or nonlinear tire models as they require more parameter identification work and lead to complex numerical optimization problems. All these tire details are removed from the prediction model in our concept and suppressed by the low-level, fast feedback control and a good feedforward setup based on the self-steering characteristic of the vehicle. The inevitable tracking uncertainty can be tackled by the robust MPC and its assumptions on the disturbances.

The model has the following state variables: the progress along the path or the path variable s, the lateral deviation d between the path and the vehicle's centre of gravity, the orientation error $\Delta\psi = \psi - \bar{\psi}$ of the vehicle's chassis ψ towards the path $\bar{\psi}$ and the vehicle's velocity v aligned with its longitudinal axis. The system inputs are the longitudinal force F_x and the actually driven curvature κ. The latter is often expressed using the neutral steer assumption $\kappa = \frac{\delta}{l}$, with the steering wheel angle δ and the wheelbase l. However, this relation depends strongly on the tyre and suspension design and is therefore suspect to significant uncertainties. The only model parameter in the resulting dynamics is the vehicle mass m. The path curvature enters the model description as an explicit function representation $\bar{\kappa}(s)$. The following differential equations describe the model:

$$\begin{bmatrix} \dot{s} \\ \dot{v} \\ \dot{d} \\ \Delta\dot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{v\cos(\Delta\psi)}{1-d\bar{\kappa}(s)} \\ \frac{F_\mathrm{x}}{m} \\ v\sin(\Delta\psi) \\ \kappa v - \bar{\kappa}(s)\dot{s} \end{bmatrix}. \tag{10}$$

The racing vehicle's dynamic capabilities are taken into account based on the well-known combined lateral a_y and longitudinal a_x constraints (usually formulated as gg-diagram). It combines the influence of several parameters such as tyre characteristics, aerodynamic performance, or the mechanical suspension setup [1,26]. While often approximated as an ellipsoid, high-performance driving experiments [60] suggest that a diamond shape can lead to more realistic representations of the dynamic limits

$$|\frac{a_\mathrm{x}}{a_\mathrm{x,max}}| + |\frac{a_\mathrm{y}}{a_\mathrm{y,max}}| \leq 1, \tag{11}$$

with $a_\mathrm{x,max}$ and $a_\mathrm{y,max}$ being the acceleration limits of the corresponding direction. Despite its more realistic representation of the vehicle capabilities, the constraint's linear nature is beneficial for the numerical implementation. However, it would be possible to extend the approach presented in this paper towards more complex polytopic constraint shapes. Box constraints on the maximum lateral deviation d_max complete the set of considered limitations.

3.4. Autonomous racing software architecture

The TMPC controller is integrated within the software stack presented by [5] for an autonomous racing car. The relevant interfaces are depicted in Figure 3. Initially, the local trajectory planner generates the target trajectory for the vehicle based on the

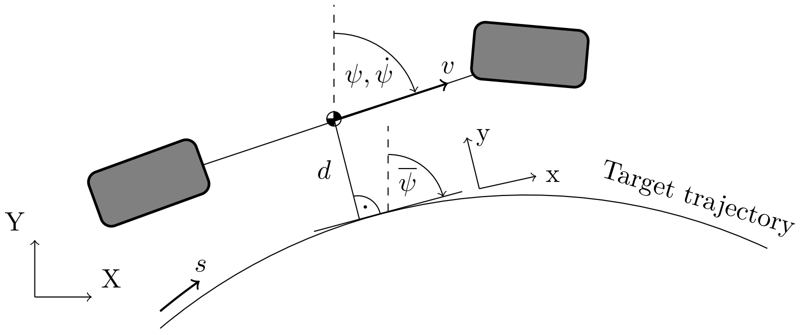


Figure 2.: Vehicle point mass model in error coordinates

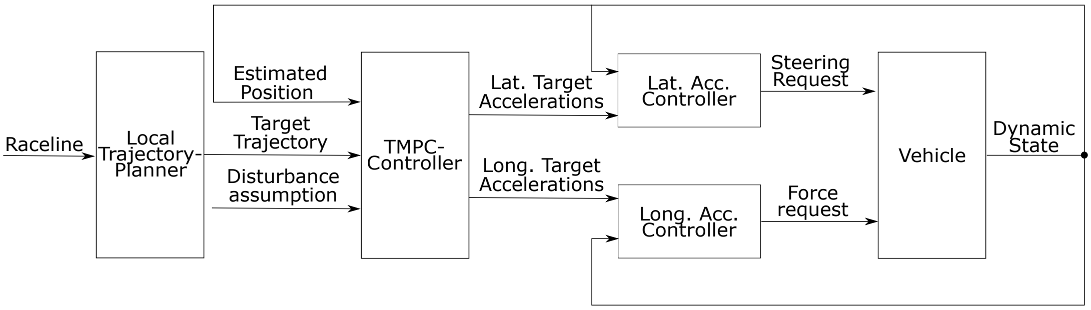


Figure 3.: Autonomous racing software architecture including the TMPC controller

raceline. The TMPC controller then receives the specified trajectory and estimated position, and calculates longitudinal and lateral target accelerations, considering the current deviation from the target trajectory (desired path and velocity profile), the acceleration constraints and the disturbance assumption. Lastly, the vehicle controller calculates a feed-forward steering angle and a force request from those target accelerations. Besides, proportional feedback controllers reduce the deviation between the requested and measured accelerations. We will not consider these low-level controllers in the rest of this paper as the closed-loop response times from both accelerations are, in comparison to the TMPC execution frequency, neglectable due to the high-performance actuators used within many autonomous race cars [1].

3.5. Formulation of the TMPC controller as a quadratic program

The following section will explain how we transform the point-mass model such that an efficient and precise approximation of the TMPC problem as a quadratic program is possible. The key idea is to introduce the corrective accelerations as manipulated variables, which allows us to reformulate the dynamics as a linear system.

Applying the state transformation $\dot{d} = v\sin(\Delta\psi)$ and the input transformations $F_x = ma_x$ and $\kappa v^2 = a_y$, the dynamics can be written as

$$\begin{bmatrix} \dot{s} \\ \dot{v} \\ \dot{d} \\ \ddot{d} \end{bmatrix} = \begin{bmatrix} \dfrac{v\cos(\Delta\psi)}{1 - d\bar{\kappa}(s)} \\ a_{\mathrm{x}} \\ v\sin(\Delta\psi) \\ a_{\mathrm{x}}\sin(\Delta\psi) + a_{\mathrm{y}}\cos(\Delta\psi) - \bar{\kappa}(s)\dot{s}v\cos(\Delta\psi) \end{bmatrix}. \tag{12}$$

Even though these transformations result in more insightful dependencies from the accelerations, the dynamic equations are still nonlinear. For this reason, we introduce a second input transformation by formulating the equations based on accelerations relative to the target path and velocity profile. Using the input definition

$$\ddot{d} = \Delta a_{\mathrm{y}} = a_{\mathrm{x}}\sin(\Delta\psi) + a_{\mathrm{y}}\cos(\Delta\psi) - \bar{\kappa}(s)\dot{s}v\cos(\Delta\psi). \tag{13}$$

the lateral dynamics become a double integrator. By expressing the longitudinal dynamics with respect to the reference velocity profile $\Delta v = \bar{v}(s) - v$, the longitudinal dynamics input can be defined as Δa_x. We assume that it is possible to obtain a reasonable approximation for the short-term behaviour of the prediction using the assumption that the target velocity profile and the path curvature can be written as purely time-dependent expressions $v(s(t))$ and $\bar{\kappa}(s(t))$. We can then neglect the state dynamics for s and use the following disturbed linear system to approximate the vehicle behaviour in the near future:

$$\dot{x} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta v \\ d \\ \dot{d} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta a_{\mathrm{x}} \\ \Delta a_{\mathrm{y}} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d_{\mathrm{a,x}} \\ d_{\mathrm{a,y}} \end{bmatrix}. \tag{14}$$

It remains to express the accelerations in terms of the states and inputs of the above dynamic model. The longitudinal acceleration is given from the definition as

$$a_{\mathrm{x}} = \frac{\partial \bar{v}(s)}{\partial s} \frac{v \cos(\Delta\psi)}{1 - d\bar{\kappa}(s)} - \Delta a_{\mathrm{x}}. \tag{15}$$

For the lateral acceleration, we rearrange Eq. (13) and use Eq. (15) to obtain

$$a_{\mathrm{y}} = \frac{\Delta a_{\mathrm{y}}}{\cos(\Delta\psi)} + \bar{\kappa}\frac{\cos(\Delta\psi)}{1 - d\bar{\kappa}(s)} v^2 - \left(\frac{\partial v(s)}{\partial s} \frac{v \cos(\Delta\psi)}{1 - d\bar{\kappa}(s)} - \Delta a_{\mathrm{x}}\right) \tan(\Delta\psi). \tag{16}$$

We could now linearise the above equations concerning the states and inputs. However, to keep the approximation simple, we apply the same assumption as previously and assume that $\bar{\kappa}(s)$ and $\bar{v}(s)$ do not depend on the state variables. The linearisation trajectory is chosen as the reference path and the predicted velocity profile from the MPC scheme's last iteration. The latter shows significantly better approximations of the lateral acceleration constraints as the target velocity even in case of small deviations as the velocity influence is of quadratic nature. Details on this procedure can be found in Section 7.

We are now ready to formulate the TMPC problem within the scheme presented in Section 3.2. We calculate the (approximate) ellipsoidal reachable sets based on the linear system Eq. (14) and neglect effects such as the implicit relationship between the lateral and longitudinal case introduced via the state uncertainty and the influence of the nonlinear state transformations. In the following, the uncertainty model will therefore purely act as a tuning parameter and can not be assumed to generate reliable over-approximations of the reachable sets. A detailed study on this topic is considered future work and neglected here for the sake of computational and algorithmic complexity. The tightening of the constraints takes the uncertainty within the lateral error d and the resulting corrective accelerations Δa_x and Δa_y into account but neglects the influences of state uncertainty on the acceleration constraints due to the approximate nature of the reachable sets. The terminal cost matrix P is taken such that $p^T P p$ reflects the cost-to-go for a certain p. It can be determined from the solution of the algebraic Riccati equation for the system dynamics and the cost matrices Q and R. Furthermore, it is beneficial to add quadratic regularisation terms for the change rate of the lateral and longitudinal accelerations to the cost function. The resulting smoothing of the closed-loop acceleration trajectories reduces the influence of the neglected acceleration dynamics and therefore the dynamic model matches the actual system behaviour better. For the sake of simplicity, we base the regularisation on the sum of the linearisation scheme's operating point accelerations and the corrective

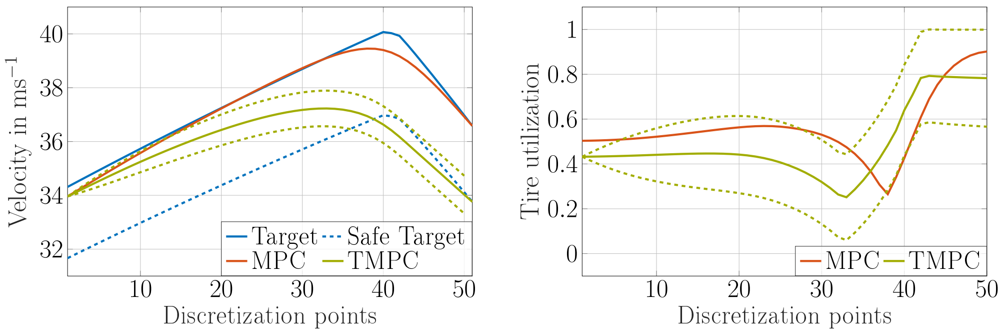


Figure 4.: Terminal set design by using a safe target velocity

accelerations, instead of the exact nonlinear transformations:

$$\tilde{a}_{\mathrm{x}} = a_{\mathrm{x,op}} + \Delta a_{\mathrm{x}}, \tag{17a}$$
$$\tilde{a}_{\mathrm{y}} = a_{\mathrm{y,op}} + \Delta a_{\mathrm{y}}. \tag{17b}$$

Besides, a terminal constraint set prevents the controller from being overly optimistic due to the short optimisation horizons. For the lateral error, we choose the same box constraints as within the optimisation horizon to be fulfilled robustly. The lateral error derivative is required to be as small as possible (the nominal prediction is constrained to zero) inspired by the discussion about appropriate terminal sets in [61]. Intuitively, this enforces the vehicle to corner on a fixed radius at the end of the optimisation horizon. The choice of the terminal set velocity is a little more involved: based on the acceleration limits used for planning the target trajectory $a_{\mathrm{x,plan}}$ and $a_{\mathrm{y,plan}}$ and the constraint tightening for the acceleration constraints from Eq. (9e), a suitable velocity scale factor θ_v to obtain a safe target velocity $v_{\mathrm{T}} = \theta_v \bar{v}$ is determined

$$\theta_v^2 \max\left(|\frac{a_{\mathrm{x,plan}}}{a_{\mathrm{x,max}}}|,|\frac{a_{\mathrm{y,plan}}}{a_{\mathrm{y,max}}}|\right)+\sqrt{\begin{bmatrix}\frac{1}{a_{\mathrm{x,max}}} & \frac{1}{a_{\mathrm{y,max}}}\end{bmatrix} K M_{N_p} K^T \begin{bmatrix}\frac{1}{a_{\mathrm{x,max}}} & \frac{1}{a_{\mathrm{y,max}}}\end{bmatrix}^T}=1. \tag{18}$$

It should be noted that this choice does not guarantee the terminal set to be robust positive invariant but has shown to be an easy to determine and reliable heuristic in our experiments. The above choice and the effects on the optimisation problem are visualised in Fig. 4. While the nominal MPC algorithm sticks closely to the target trajectory, the TMPC follows the target trajectory at the beginning of the optimisation horizon but deviates significantly towards the end. This cautious behaviour allows for exploitation of the vehicle capabilities as long as it is possible to keep a viable solution in case disturbances enter the system.

Even though we apply a robust MPC formulation, a violation of the disturbance assumptions might still lead to the infeasibility of the optimisation problem. Therefore, the constraints are softened using slack variables ϵ following the idea of exact penalty functions [62]. The one-norm for slack costs and a small regularisation term use the two-norm guarantee to obtain the hard constrained solution in case the problem is feasible. Note that the terminal constraints are not softened as it is always possible to reach them by appropriate softening of the stage constraints. As an additional safety measure, we constrain the slacks to be limited to a maximum violation of 5 % and trigger an emergency manoeuvre in case a solution can not be achieved within this

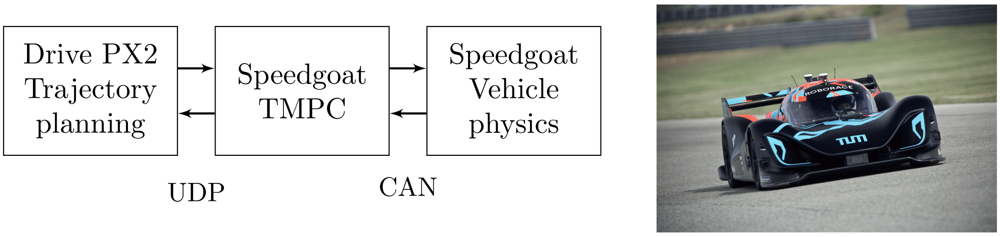


Figure 5.: Hardware-in-the-loop setup and corresponding Roborace DevBot 2.0 vehicle

tolerance. The final cost function with all modifications can now be written as

$$\sum_{k=0}^{N_p-1} \left(p_k^T Q p_k + r_k^T R r_k + \gamma_{\mathrm{x}} \tilde{a}_{\mathrm{x}}'^2 + \gamma_{\mathrm{y}} \tilde{a}_{\mathrm{y}}'^2 \right) + p_{N_p}^T P p_{N_p} + \rho_1 ||\epsilon||_1 + \rho_2 ||\epsilon||_2 \tag{19}$$

with $\tilde{a}_{\mathrm{x}}'$ and $\tilde{a}_{\mathrm{y}}'$ being the differences between two consecutive stages for the approximated lateral and longitudinal accelerations and γ_{x} and γ_{y} the corresponding cost function weights. The slack weights are written as ρ_1 and ρ_2.

There are many solvers readily available to solve quadratic optimisation problems. We chose the first-order solver OSQP [63] (based on the alternating direction method of multipliers) due to its high performance for low to medium accuracy solutions which are well suited for online optimisation with a limited computational budget. The discretisation of the dynamics was done using the Euler-forward method at a step size of 40 ms, which is also the execution frequency of the TMPC algorithm. The prediction horizon is chosen to be 2 s long and consists of $N_p = 50$ steps, leading to a medium-sized problem with 400 optimisation variables (100 control variables and 300 slack variables) and 453 constraints. The QP is solved once per execution cycle and due to the successive linearisation, the presented algorithm can be considered a real-time SQP method. The linearisation is done with respect to a weighted combination of the previous iteration linearisation and the previous iteration solution of the QP. Similar to [23], we found that only using the previous iteration solution, oscillations are likely to occur. The weighting of the previous step solution is done with a factor of $\alpha = 0.3$. The OSQP algorithm is limited to 200 iterations per execution cycle and is warm started for better convergence. Furthermore, it turned out to be beneficial to use a prediction of the measured vehicle state as the initial state by one time-step to compensate for the execution time of the algorithm.

4. Results and discussion

4.1. Hardware-in-the-Loop setup and parameters

The experiments are conducted on a Hardware-in-the-Loop testbench using the same hardware as the Roborace DevBot 2.0 vehicle (see Fig. 5). It uses a Speedgoat Mobile as a rapid prototyping controller (RPC) with an i7 2.5Ghz CPU and 4GB RAM. The physics simulation is implemented on a Speedgoat Performance and is based on a nonlinear dual-track model with a Pacejka tyre model considering lateral as well as longitudinal and combined tyre behaviour. The parameters are taken from the simulation environment used within the Roborace competition [5] and represent a vehicle with LMP2 chassis and road sport tyres. The simulation implements actuator and sen-

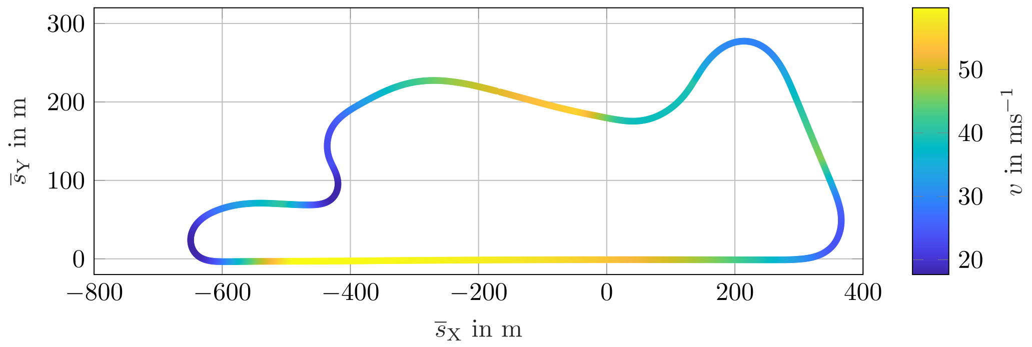


Figure 6.: Monteblanco track layout

sor models to accurately reflect response delay, low-pass characteristics and random noise. The RPC and the simulation communicate via CAN-Bus. The remaining parts of the software framework and the simulation environment are taken from the TUM Roborace Software Stack described in [5] and are largely available open-source from [64]. The trajectory planning is executed on an NVIDIA Drive PX2 and communicates to the RCP via a UDP interface. The implementation of the control algorithms took place in Simulink using a custom C-code interface to the QP-solver OSQP [63].

The following results have all been obtained from simulations for the Monteblanco racetrack, depicted in Fig. 6. Its combination of tight corners with high-speed sections make this track a good choice to benchmark different controller concepts. The controller acceleration limits are chosen to be $a_{\mathrm{x,max}}, a_{\mathrm{y,max}} = 12.5\,\mathrm{m\,s^{-2}}$ and the maximum lateral deviation is set to be $d_{\mathrm{max}} = 1\,\mathrm{m}$. The cost function matrices are chosen to

$$Q = \begin{bmatrix} 0.05 & 0 & 0 \\ 0 & 20.0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad R = \begin{bmatrix} 0.01 & 0 \\ 0 & 1 \end{bmatrix}. \tag{20}$$

The regularisation terms are set to $\gamma_x = 0.2$ and $\gamma_y = 20$ and the slack weights to $\rho_1 = 1000$ and $\rho_2 = 100$. For benchmarking purposes, we will compare the presented TMPC against a nominal MPC controller as well as the infinite time LQR solution. It should be noted that the resulting LQR controller gains are close to the ones found empirically for the controller presented during real-world experiments with the prototype [1,5]. Subsequently, we will conduct a parameter study on the TMPC controller by comparing different choices for the pre-stabilising controller K as well as different disturbance assumptions.

4.2. Comparison of control concepts

The following chapter compares the TMPC controller to a nominal MPC and an LQR controller designed similar to the concept of Heilmeier et al. [1]. The disturbance set represents the expected mismatch between the modelled dynamics and real vehicle behaviour. In the longitudinal direction, the main influences are driving resistances and powertrain and brake system uncertainties, whereas, in the lateral direction, the vehicle's self-steering behaviour and yaw dynamics influence the actual acceleration most. The disturbance bounds are chosen to be $\bar{d}_{a,x} = 1.0\,\mathrm{m\,s^{-2}}$ and $\bar{d}_{a,y} = 1.0\,\mathrm{m\,s^{-2}}$. We

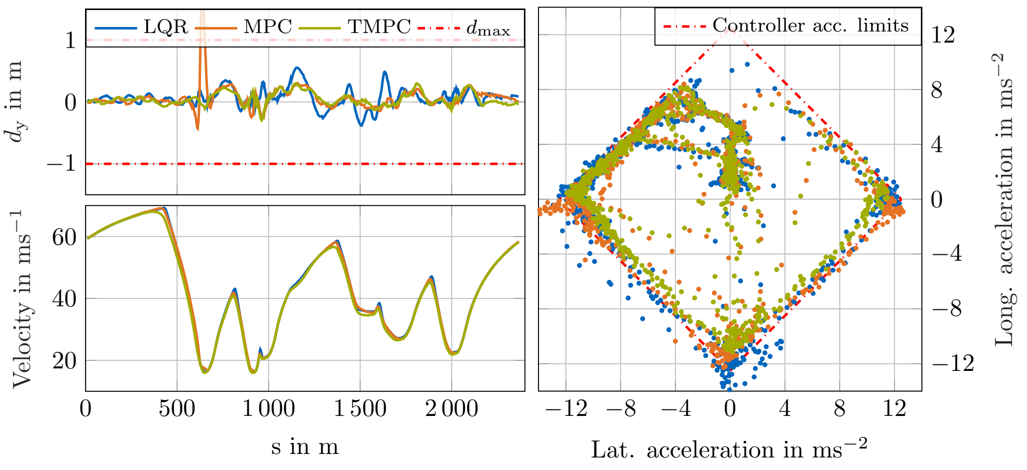


Figure 7.: Lateral deviation, velocity, and gg-diagram (with controller acceleration limits) of the LQR, MPC and TMPC controller

limit the accelerations in the trajectory planner to 12.0 m s^{-2}. The simulated vehicle physics are modified, such that the friction coefficient of the brake pads is decreased by 10 %. This parameter is highly temperature and material-dependent and is, therefore, a good choice to test the controller's robustness.

Fig. 7 shows the accelerations and lateral deviations over one flying lap for the different controllers. The acceleration plot depicts that the controller has an increasing amount of constraint violations with decreasing knowledge about the constraints and uncertainties. Note that the limitation on the positive accelerations is related to the power limit of the electric powertrain that cannot accelerate the car further. Due to no constraint consideration, the LQR controller completes the lap with the fastest time of 62.77 s. The lap-time is achieved by over-stressing the tyres and thus violating the acceleration constraints. The second-fastest time is achieved by the nominal MPC controller with 63.32 s. In addition, the MPC violates the lateral constraint in the first turn as it brakes to late but is still constrained by the tire limits. Driving more cautiously with the TMPC controller increases the lap-time to 64.67 s but leads to a closed-loop behavior which respects the constraints. These rather obvious results indicate that the TMPC sacrifices the lap-time promised by the target trajectory to increase the closed-loop robustness and safety.

However, the risk of severe accidents makes the aspect of constraint satisfaction an equally important performance indicator under race conditions. We, therefore, reformulate the performance analysis as a trade-off between the lap-time and constraint satisfaction for the presented control concepts for different settings of the local trajectory planner (see Fig. 8). The TMPC controller achieves superior results in this comparison as it demonstrates the lowest number of constraint violations and also the minimum absolute constraint violation for comparable lap-times. The reason for this is that the TMPC controller stays cautious in risky driving situations but exploits the full potential when it deems the situation safe enough. However, it can only guarantee constraint satisfaction if the used model matches all underlying assumptions which does not hold for the fastest lap-time. It shall be noted that even in this case the TMPC has the lowest amount of constraint violations and can therefore still be considered safer than the other concepts. In contrast, the MPC and LQR controller only achieve a similar safety level if the accelerations limits are globally reduced which

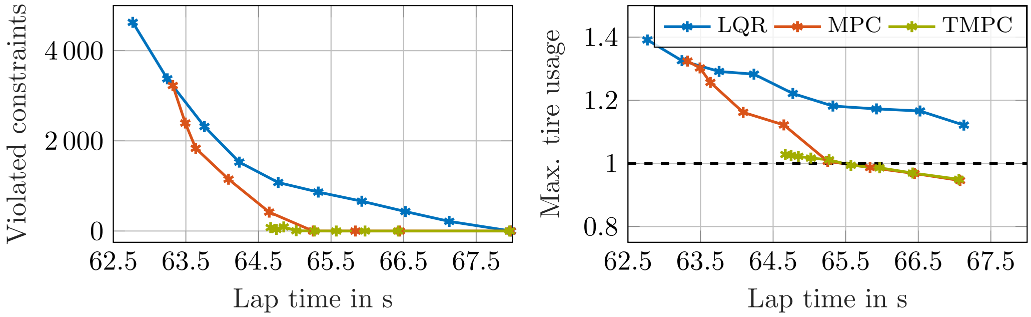


Figure 8.: Number of violated constraints and maximum tyre usage in comparison to the achieved lap-time for the controllers on a flying lap

results in a worse overall lap-time. These results indicate that intelligent use of uncertainty knowledge can increase the safety of the controller in autonomous vehicle racing while keeping a competitive level of performance.

4.3. Control parameter effects

The effect of the disturbance bounds on the control performance is analysed by varying them jointly for the longitudinal $\bar{d}_{a,x}$ and lateral $\bar{d}_{a,y}$ direction with a step size of $0.5\,\mathrm{m\,s^{-2}}$. Additionally, we will compare the effects of two pre-stabilising controllers K for the TMPC concept. First, we use the infinite-time LQR solution for the system with the weight matrices Q and R. Second, an optimised controller obtained from the procedure presented in [48]. The controllers are

$$K_{\mathrm{LQR}} = \begin{bmatrix} -2.14 & 0 & 0 \\ 0 & -4.21 & -2.99 \end{bmatrix} \quad K_{\mathrm{Opt}} = \begin{bmatrix} -5.02 & 0 & 0 \\ 0 & -28.13 & -10.44 \end{bmatrix}. \tag{21}$$

To get a realistic impression of the control system robustness, we run a multitude of simulations (conducted in Simulink for more efficient evaluation but with the same models as the Hardware-in-the-Loop setup) for different brake pad friction coefficients ranging from 80 % to 120 % of the nominal value in 5 % steps. The MPC controller (with $\bar{d}_{a,x} = \bar{d}_{a,y} = 0\,\mathrm{m\,s^{-2}}$) demonstrates the fastest lap-time, however, it also shows the largest number of tyre and lateral deviation constraint violations. While the soft constraints allow the controller to keep the vehicle on track by generating reasonable backup controls, it violates the specified envelope of safe operation and is therefore risky to transfer to the real car. With increasing disturbance set size, the TMPC controller brakes earlier when approaching a corner to account for the uncertainty within the model and therefore loses lap-time (see Fig. 9). The lateral control operation is influenced less by the uncertainties and therefore the steering angle only differs marginally for the three controllers.

Comparing the stabilising controllers in Fig. 10 shows that the maximum number of violated constraints decreases slightly faster for the case of K_{Opt} in comparison to K_{LQR}. However, the results indicate that the pre-stabilising controller's choice influences the performance of the closed-loop system only marginally. This is related to the fact that it is not actually executed and only used to construct predictions for the uncertain system behaviour and act cautiously in state-space directions where this might impact future constraint satisfaction. We, therefore, advise staying with the

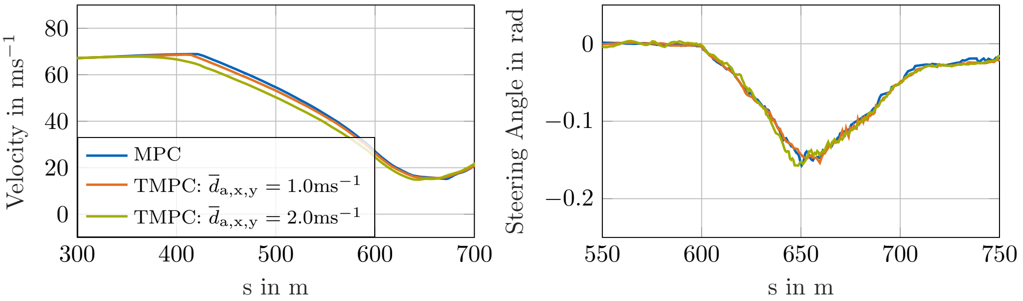


Figure 9.: Comparison of the vehicles velocity profile and steering angle for the MPC and two TMPC controllers with different set sizes

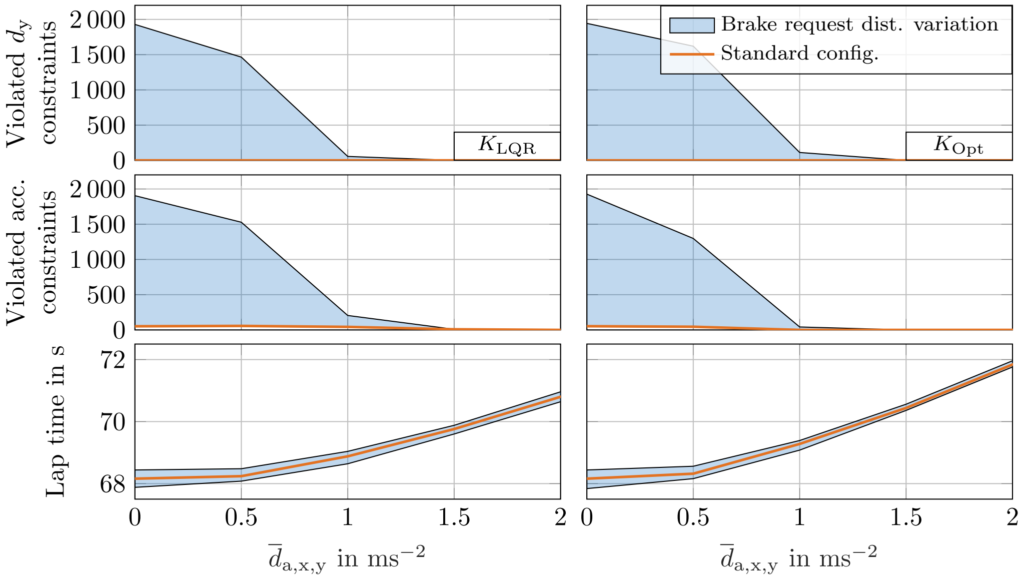


Figure 10.: Comparison of violated constraints and lap-time for K_{LQR} (left) and K_{Opt} (right) with different disturbance sets for varied brake force disturbances

LQR controller due to the easy design process. The results also indicate that disturbance limits of $1\,\mathrm{m\,s^{-2}}$ reach an appropriate coverage of the model uncertainties for the sophisticated dual-track model used within the Hardware-in-the-Loop setup.

5. Conclusion and outlook

In this paper, we presented a real-time capable Tube-MPC approach for trajectory tracking of an autonomous race car at the handling limits. In contrast to nominal MPC and a classic LQR concept, the consideration of additive disturbances shows to be an efficient way to reduce the number of constraint violations while maintaining competitive lap-times. The resulting controller tends to be cautious when it is required but can drive aggressively as the uncertainty within the system does not lead to risky situations. The nonlinear receding horizon optimisation problem is implemented on a rapid-prototyping ECU using a linearised model and the QP-solver OSQP. The results have been obtained on a Hardware-in-the-Loop test bench with a sophisticated nonlinear dual-track model for vehicle physics simulation as well as actuator and sensor models. Furthermore, a parameter study demonstrates that the upper bound of the disturbance assumption is a reasonable tuning parameter for the trade-off between constraint violations and lap-time. The choice of the pre-stabilising controller for the TMPC concept has not shown to significantly alter the closed-loop behaviour.

We want to emphasise that the results presented in this paper indicate that already simple dynamic models are able to deliver high-performance driving in autonomous racing applications when reasonable uncertainty and constraint assumptions are considered. This is especially advantageous in face of the increased computational requirements and numerical difficulties when complex nonlinear tyre models are employed for model predictive control. Furthermore, these complex models are difficult to parameterise which makes their increased accuracy questionable in real-world applications.

In the future, we plan to use the TMPC approach on a real autonomous race car and evaluate the proposed advantages under real-world conditions. Promising research directions are a more thorough analysis of the constructed tubes of reachable sets and the combination with machine learning algorithms to learn appropriate uncertainty models while driving. These improvements would allow the controller to adjust its speed and behaviour to the current environmental conditions. Furthermore, it might be an interesting research direction to allow for more deviation from the planned trajectory and allow the controller to simultaneously reoptimise the planned trajectory. This strategy could lower the quality requirements for the target trajectory and therefore improve the overall system performance in more complex racing scenarios.

6. Contributions and acknowledgements

Alexander Wischnewski and Martin Euler designed and implemented the presented robust control algorithm. They are further responsible for the analysis of the system performance and parameter studies. Salih Gümüs is responsible for the real-time embedded MPC scheme. Boris Lohmann contributed to the critical revision of the manuscript and the conception of the research project. Furthermore, we would like to thank Thomas Herrmann, Johannes Schwarz and Thomas Specker for the many discussions on the topics and comments on the manuscript.

References

[1] Heilmeier A, Wischnewski A, Hermansdorfer L, et al. Minimum curvature trajectory planning and control for an autonomous race car. Vehicle System Dynamics. 2019;58(10):1–31.

[2] Liniger A, Domahidi A, Morari M. Optimization-based autonomous racing of 1:43 scale rc cars. Optimal Control Applications and Methods. 2014;36(5):628–647.

[3] Laurense VA, Goh JY, Gerdes JC. Path-tracking for autonomous vehicles at the limit of friction. In: Proceedings of the 2017 American Control Conference; 2017. p. 5586–5591.

[4] Novi T, Liniger A, Capitani R, et al. Real-time control for at-limit handling driving on a predefined path. Vehicle System Dynamics. 2019;58(7):1007–1036.

[5] Betz J, Wischnewski A, Heilmeier A, et al. A software architecture for the dynamic path planning of an autonomous racecar at the limits of handling. In: Proceedings of the 2019 IEEE International Conference on Connected Vehicles and Expo; 2019. p. 1–8.

[6] Stahl T, Wischnewski A, Betz J, et al. Multilayer graph-based trajectory planning for race vehicles in dynamic scenarios. In: Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference; 2019. p. 3149–3154.

[7] Falcone P, Borrelli F, Asgari J, et al. Predictive active steering control for autonomous vehicle systems. IEEE Transactions on Control Systems Technology. 2007;15(3):566–580.

[8] Nolte M, Rose M, Stolte T, et al. Model predictive control based trajectory generation for autonomous vehicles — an architectural approach. In: Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV); 2017. p. 798–805.

[9] Ulbrich S, Menzel T, Reschka A, et al. Defining and substantiating the terms scene, situation, and scenario for automated driving. In: Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems; 2015. p. 982–988.

[10] Ulbrich S, Nothdurft T, Maurer M, et al. Graph-based context representation, environment modeling and information aggregation for automated driving. In: Proceedings of the 2014 IEEE Intelligent Vehicles Symposium; 2014. p. 541–547.

[11] Roselli F, Corno M, Savaresi SM, et al. H_∞ control with look-ahead for lane keeping in autonomous vehicles. In: Proceedings of the 2017 IEEE Conference on Control Technology and Applications; 2017. p. 2220–2225.

[12] Guldner J, Han-Shue T, Patwardhan S. Study of design directions for lateral vehicle control. In: Proceedings of the 36th IEEE Conference on Decision and Control; Vol. 5; 1997. p. 4732–4737 vol.5.

[13] Menhour L, d'Andréa-Novel B, Fliess M, et al. Coupled nonlinear vehicle control: Flatness-based setting with algebraic estimation techniques. Control Engineering Practice. 2014; 22:135–146.

[14] Villagra J, d'Andrea-Novel B, Mounier H, et al. Flatness-based vehicle steering control strategy with sdre feedback gains tuned via a sensitivity approach. IEEE Transactions on Control Systems Technology. 2007;15(3):554–565.

[15] Aguilar LE, Hamel T, Soueres P. Robust path following control for wheeled robots via sliding mode techniques. In: Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications; Vol. 3; 1997. p. 1389–1395.

[16] Tagne G, Talj R, Charara A. Higher-order sliding mode control for lateral dynamics of autonomous vehicles, with experimental validation. In: Proceedings of the 2013 IEEE Intelligent Vehicles Symposium; 2013. p. 678–683.

[17] Hamerlain F, Achour K, Floquet T, et al. Trajectory tracking of a car-like robot using second order sliding mode control. In: Proceedings of the 2007 European Control Conference; 2007. p. 4932–4936.

[18] Manceur M, Menhour L. Higher order sliding mode controller for driving steering vehicle wheels: Tracking trajectory problem. In: Proceedings of the 52nd IEEE Conference on Decision and Control; 2013. p. 3073–3078.

[19] Attia R, Orjuela R, Basset M. Combined longitudinal and lateral control for automated vehicle guidance. Vehicle System Dynamics. 2014;52(2):261–279.

[20] Turri V, Carvalho A, Tseng HE, et al. Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads. In: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013); 2013. p. 378–383.

[21] Calzolari D, Schürmann B, Althoff M. Comparison of trajectory tracking controllers for autonomous vehicles. In: Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems; 2017. p. 1–8.

[22] Bujarbaruah M, Zhang X, Tseng E, et al. Adaptive mpc for autonomous lane keeping. In: Proceedings of the 14th International Symposium on Advanced Vehicle Control; 2018.

[23] Funke J, Brown M, Erlien SM, et al. Collision avoidance and stabilization for autonomous vehicles in emergency scenarios. IEEE Transactions on Control Systems Technology. 2017 Jul;25(4):1204–1216.

[24] Brown M, Gerdes JC. Coordinating tire forces to avoid obstacles using nonlinear model predictive control. IEEE Transactions on Intelligent Vehicles. 2020;5(1):21–31.

[25] Anderson J, Ayalew B. Modelling minimum-time manoeuvering with global optimisation of local receding horizon control. Vehicle System Dynamics. 2018 01;56:1508–1531.

[26] Subosits JK, Gerdes JC. From the racetrack to the road: Real-time trajectory replanning

for autonomous driving. IEEE Transactions on Intelligent Vehicles. 2019;4(2):309–320.

[27] Alcalá E, V Puig, Quevedo J, et al. Autonomous racing using linear parameter varying-model predictive control. Control Engineering Practice. 2020;95:104270.

[28] Vázquez JL, Brühlmeier M, Liniger A, et al. Optimization-based hierarchical motion planning for autonomous racing. arXivorg. 2020;.

[29] Kabzan J, Hewing L, Liniger A, et al. Learning-based model predictive control for autonomous racing. IEEE Robotics and Automation Letters. 2019 Oct;4(4):3363–3370.

[30] Rosolia U, Borrelli F. Learning how to autonomously race a car: A predictive control approach. IEEE Transactions on Control Systems Technology. 2020 Nov;28(6):2713–2719.

[31] Langson W, Chryssochoos I, Rakovic SV, et al. Robust model predictive control using tubes. Automatica. 2004;40(1):125–133.

[32] Alsterda JP, Brown M, Gerdes JC. Contingency model predictive control for automated vehicles. In: Proceedings of the 2019 American Control Conference (ACC); 2019.

[33] Gao Y, Gray A, Tseng HE, et al. A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. Vehicle System Dynamics. 2014;52(6):802–823.

[34] Sakhdari B, Shahrivar EM, Azad NL. Robust tube-based MPC for automotive adaptive cruise control design. In: Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems; 2017.

[35] Rathai KMM, Amirthalingam J, Jayaraman B. Robust tube-mpc based lane keeping system for autonomous driving vehicles. 06; 2017. p. 1–6.

[36] Chisci L, Rossiter JA, Zappa G. Systems with persistent disturbances: predictive control with restricted constraints. Autom. 2001;37:1019–1028.

[37] Campo PJ, Morari M. Robust model predictive control. In: 1987 American Control Conference; 1987. p. 1021–1026.

[38] Raimondo DM, Limon D, Lazar M, et al. Min-max model predictive control of nonlinear systems: A unifying overview on stability. European Journal of Control. 2009;15(1):5–21.

[39] Bemporad A, Morari M. Robust model predictive control: A survey. In: Garulli A, Tesi A, editors. Robustness in identification and control; London. Springer London; 1999. p. 207–226.

[40] Scokaert POM, Mayne DQ. Min-max feedback model predictive control for constrained linear systems. IEEE Transactions on Automatic Control. 1998;43(8):1136–1142.

[41] Mayne D. Robust and stochastic model predictive control: Are we going in the right direction? Annual Reviews in Control. 2016;41:184–192.

[42] Mayne DQ, Seron MM, Rakovic SV. Robust model predictive control of constrained linear systems with bounded disturbances. Automatica. 2005;41(2):219–224.

[43] Mayne DQ, Kerrigan EC. Tube-based robust nonlinear model predictive control. In: Proceedings of the 7th IFAC Symposium on Nonlinear Control Systems; 2007. p. 36–41.

[44] Rakovic SV, Kouvaritakis B, Cannon M, et al. Parameterized tube model predictive control. IEEE Transactions on Automatic Control. 2012;57(11):2746–2761.

[45] Asselborn L, Groß D, Stursberg O. Control of uncertain nonlinear systems using ellipsoidal reachability calculus. In: Proceedings of the 9th IFAC Symposium on Nonlinear Control Systems; Vol. 46; 2013. p. 50–55.

[46] Koller T, Berkenkamp F, Turchetta M, et al. Learning-based model predictive control for safe exploration. In: Proceeding of the 2018 Conference on Decision and Control; 2018. p. 6059–6066.

[47] Blanchini F. Set invariance in control. Automatica. 1999;35(11):1747–1767.

[48] Limon D, Alvarado I, Alamo T, et al. On the design of robust tube-based mpc for tracking. In: Proceedings of the 17th IFAC World Congress; 2008. p. 15333–15338.

[49] Rakovic SV, Mayne DQ. Set robust control invariance for linear discrete time systems. In: Proceedings of the 44th IEEE Conference on Decision and Control. Institute of Electrical and Electronics Engineers; 2005. p. 975–980.

[50] Buckner C, Lampariello R. Tube-based model predictive control for the approach maneuver of a spacecraft to a free-tumbling target satellite. In: Proceedings of the 2018 Annual

American Control Conference; 2018. p. 5690–5697.

[51] Chen W, O'Reilly J, Ballance D. On the terminal region of model predictive control for non-linear systems with input/state constraints. Journal of Adaptive Control and Signal Processing. 2003;17(3):195–207.

[52] Le V, Alamo T, Camacho EF, et al. A new approach for guaranteed state estimation by zonotopes. IFAC Proceedings Volumes. 2011;44(1):9242–9247.

[53] Schürmann B, Althoff M. Optimal control of sets of solutions to formally guarantee constraints of disturbed linear systems. In: Proceedings of the 2017 American Control Conference. IEEE; 2017. p. 2522–2529.

[54] Yu S, Chen H, Allgöwer F. Tube mpc scheme based on robust control invariant set with application to lipschitz nonlinear systems. In: Proceedings of the 2011 50th IEEE Conference on Decision and Control; 2011. p. 2650–2655.

[55] Zeilinger MN, Jones CN, Raimondo DM, et al. Real-time mpc - stability through robust mpc design. In: Proceedings of the 48th IEEE Conference on Decision and Control. IEEE; 2009. p. 3980–3986.

[56] Kurzhanski A. Ellipsoidal calculus for estimation and feedback control. In: Byrnes CI, editor. Systems and control in the twenty-first century. (Progress in Systems and Control Theory; Vol. 277). Boston: Birkhäuser; 1997. p. 229–243.

[57] van Hessem DH, Bosgra OH. Closed-loop stochastic dynamic process optimization under input and state constraints. In: Proceedings of the 2002 American Control Conference; Vol. 3; 2002. p. 2023–2028.

[58] Polack P, Altché F, d'Andréa-Novel B, et al. The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In: Proceedings of the 2017 IEEE Intelligent Vehicles Symposium; 2017. p. 812–818.

[59] De Luca A, Oriolo G, Samson C. Feedback control of a nonholonomic car-like robot. In: Lecture notes in control and information sciences. Springer Berlin Heidelberg; 1998. p. 171–253.

[60] Betz J, Heilmeier A, Wischnewski A, et al. Autonomous driving—a crash explained in detail. Applied Sciences. 2019 Nov;9(23):5126.

[61] Liniger A, van Gool L. Safe motion planning for autonomous driving using an adversarial road model. arXivorg. 2020;.

[62] Kerrigan E, Maciejowski J. Soft constraints and exact penalty functions in model predictive control. 09; 2000.

[63] Stellato B, Banjac G, Goulart P, et al. OSQP: an operator splitting solver for quadratic programs. Mathematical Programming Computation. 2020;.

[64] Open-source software repository. Chair of Automotive Technology at Technical University of Munich. 2019;Available from: `https://github.com/TUMFTM`.

7. Appendix - Details on linearisation procedure

7.1. Longitudinal acceleration

Derivation of linearisation for Eq. (15):

$$a_{\mathrm{x}}(s, d, \dot{d}, \Delta v, \Delta a_x) = \frac{\partial \bar{v}(s)}{\partial s} \frac{(\bar{v}(s) - \Delta v) \cos(\arcsin(\frac{\dot{d}}{(\bar{v}(s) - \Delta v)}))}{1 - d\bar{\kappa}(s)} - \Delta a_{\mathrm{x}}. \tag{22}$$

We neglect the dependency of $\bar{v}(s)$ and $\bar{\kappa}(s)$ on the path coordinate s, since small modifications around the chosen linearisation trajectories are unlikely to change these values much for reasonable target trajectories due to the short horizon of the optimal control problem. Rewriting those terms as constants for each discretisation point we arrive at

$$a_{\mathrm{x}}(d, \dot{d}, \Delta v, \Delta a_x) = \bar{a}_x \frac{(\bar{v} - \Delta v)\cos(\arcsin(\frac{\dot{d}}{(\bar{v}-\Delta v)}))}{1 - d\bar{\kappa}} \; - \; \Delta a_{\mathrm{x}}. \tag{23}$$

The partial derivatives are

$$\frac{\partial a_x}{\partial d} = \bar{a}_x \bar{\kappa} \frac{(\bar{v} - \Delta v)\sqrt{1 - \frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}}}{(1 - d\bar{\kappa})^2} \tag{24a}$$

$$\frac{\partial a_x}{\partial \dot{d}} = -\bar{a}_x \frac{\dot{d}}{(\bar{v} - \Delta v)(1 - d\bar{\kappa})\sqrt{1 - \frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}}} \tag{24b}$$

$$\frac{\partial a_x}{\partial \Delta v} = -\bar{a}_x \frac{1}{(1 - d\bar{\kappa})\sqrt{1 - \frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}}} \tag{24c}$$

$$\frac{\partial a_x}{\partial \Delta a_x} = -1 \tag{24d}$$

If we now choose the path ($d = 0$, $\dot{d} = 0$), the previous iterations velocity profile ($v = \bar{v} - \Delta v = v_{\mathrm{p}}$) and zero corrective longitudinal acceleration ($\Delta a_x = 0$) as a linearisation point, we can rewrite the longitudinal acceleration in linear form as

$$a_x \approx \bar{a}_x \bar{\kappa} v_{\mathrm{p}} d - \bar{a}_x \Delta v - \Delta a_x + \bar{a}_x v_{\mathrm{p}} \tag{25}$$

7.2. Lateral acceleration

Derivation of linearisation for Eq. (16):

$$\begin{aligned} a_{\mathrm{y}}(s, d, \dot{d}, \Delta v, a_x, \Delta a_x) = & \frac{\Delta a_{\mathrm{y}}}{\cos(\arcsin(\frac{\dot{d}}{(\bar{v}(s)-\Delta v)}))} \\ & + \bar{\kappa} \frac{\cos(\arcsin(\frac{\dot{d}}{(\bar{v}(s)-\Delta v)}))}{1 - d\bar{\kappa}(s)} (\bar{v} - \Delta v)^2 \\ & - a_x \left(d, \dot{d}, \Delta v, \Delta a_x\right) \tan(\arcsin(\frac{\dot{d}}{(\bar{v}(s) - \Delta v)})) \end{aligned} \tag{26}$$

Similar to the longitudinal acceleration we neglect the dependency of $\bar{v}(s)$ and $\bar{\kappa}(s)$ and can therefore write the following

$$\begin{aligned} a_{\mathrm{y}}(d, \dot{d}, \Delta v, \Delta a_x, \Delta a_y) = & \frac{\Delta a_{\mathrm{y}}}{\cos(\arcsin(\frac{\dot{d}}{(\bar{v}-\Delta v)}))} \\ & + \bar{\kappa} \frac{\cos(\arcsin(\frac{\dot{d}}{(\bar{v}-\Delta v)}))}{1 - d\bar{\kappa}} (\bar{v} - \Delta v)^2 \\ & - a_x \left(d, \dot{d}, \Delta v, \Delta a_x\right) \tan(\arcsin(\frac{\dot{d}}{(\bar{v} - \Delta v)})) \end{aligned} \tag{27}$$

The partial derivatives are

$$
\begin{align}
\frac{\partial a_y}{\partial d} &= \bar{\kappa}^2 \frac{\cos(\arcsin(\frac{\dot{d}}{(\bar{v}-\Delta v)}))}{(1-d\bar{\kappa})^2} (\bar{v}-\Delta v)^2 - \frac{\partial a_x}{\partial d} \tan(\arcsin(\frac{\dot{d}}{(\bar{v}-\Delta v)})) \tag{28a}\\
\frac{\partial a_y}{\partial \dot{d}} &= \frac{\Delta a_y \dot{d}}{(\bar{v}-\Delta v)^2 \left(1-\frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}\right)^{3/2}} \nonumber\\
&\quad -\bar{\kappa}\frac{\dot{d}}{(1-d\bar{\kappa})\sqrt{1-\frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}}} \nonumber\\
&\quad -\frac{\partial a_x}{\partial \dot{d}}(d,\dot{d},\Delta v,\Delta a_x)\tan(\arcsin(\frac{\dot{d}}{(\bar{v}-\Delta v)})) \nonumber\\
&\quad -a_x(d,\dot{d},\Delta v,\Delta a_x)\left(\frac{\dot{d}^2}{(\bar{v}-\Delta v)^3\left(1-\frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}\right)^{3/2}} + \frac{1}{(\bar{v}-\Delta v)\sqrt{1-\frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}}}\right) \tag{28b}\\
\frac{\partial a_y}{\partial \Delta v} &= \frac{\Delta a_y \dot{d}^2}{(\bar{v}-\Delta v)^3 \left(1-\frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}\right)^{3/2}} \nonumber\\
&\quad \frac{\bar{\kappa}}{1-d\bar{\kappa}}\left(\frac{\dot{d}^2}{(\bar{v}-\Delta v)\sqrt{1-\frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}}} - 2\,(\bar{v}-\Delta v)\sqrt{1-\frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}}\right) \nonumber\\
&\quad -\frac{\partial a_x}{\partial \Delta v}(d,\dot{d},\Delta v,\Delta a_x)\tan(\arcsin(\frac{\dot{d}}{(\bar{v}-\Delta v)})) \nonumber\\
&\quad -a_x(d,\dot{d},\Delta v,\Delta a_x)\left(\frac{\dot{d}}{(\bar{v}-\Delta v)^2}\sqrt{1-\frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}} + \frac{\dot{d}^2}{(\bar{v}-\Delta v)^3\left(1-\frac{\dot{d}^3}{(\bar{v}-\Delta v)^4}\right)^{3/2}}\right) \tag{28c}\\
\frac{\partial a_y}{\partial \Delta a_x} &= \tan(\arcsin(\frac{\dot{d}}{(\bar{v}-\Delta v)})) \tag{28d}\\
\frac{\partial a_y}{\partial \Delta a_y} &= \frac{1}{\cos(\arcsin(\frac{\dot{d}}{(\bar{v}-\Delta v)}))} \tag{28e}
\end{align}
$$

If we now choose the path ($d = 0$, $\dot{d} = 0$), the previous iterations velocity profile ($v = \bar{v} - \Delta v = v_{\mathrm{p}}$) and zero corrective acceleration ($\Delta a_x = 0$, $\Delta a_y = 0$) as a linearisation point, we can rewrite the longitudinal acceleration in linear form as

$$
a_y \approx \bar{\kappa}^2 v_{\mathrm{p}}^2 d - \frac{a_x(d,\dot{d},\Delta v,\Delta a_x)}{v_{\mathrm{p}}}\dot{d} - 2\bar{\kappa} v_{\mathrm{p}} \Delta v + \Delta a_y + \bar{\kappa} v_{\mathrm{p}}^2 \tag{29}
$$

7.3. Linearisation of tyre constraints

Finally we can reformulate the tyre constraints in Eq. (11)

$$|\frac{a_{\mathrm{x}}}{a_{\mathrm{x,max}}}| + |\frac{a_{\mathrm{y}}}{a_{\mathrm{y,max}}}| \leq 1, \tag{30}$$

to be

$$\pm\, a_{\mathrm{x}} a_{\mathrm{y,max}} \pm a_{\mathrm{y}} a_{\mathrm{x,max}} \leq a_{\mathrm{x,max}} a_{\mathrm{y,max}}. \tag{31}$$

Using the above results, we can write this as a linear function of the inputs and the states

$$\begin{aligned}&\pm\left(\bar{a}_x \bar{\kappa} v_{\mathrm{p}} d - \bar{a}_x \Delta v - \Delta a_x + \bar{a}_x v_{\mathrm{p}}\right) a_{\mathrm{y,max}}\\ &\pm\left(\bar{\kappa}^2 v_{\mathrm{p}}^2 d - \frac{a_x(d, \dot{d}, \Delta v, \Delta a_x)}{v_{\mathrm{p}}} \dot{d} - 2\bar{\kappa} v_{\mathrm{p}} \Delta v + \Delta a_y + \bar{\kappa} v_{\mathrm{p}}^2\right) a_{\mathrm{x,max}} \leq a_{\mathrm{x,max}} a_{\mathrm{y,max}}.\end{aligned} \tag{32}$$

7.3 A Tube-MPC Approach to Autonomous Multi-Vehicle Oval Racing

Contributions: Alexander Wischnewski, the author of this dissertation, developed and analyzed the proposed control concept. Thomas Herrmann and Frederik Werner contributed to the real-time capable implementation and integration in the software stack of the TUM Autonomous Motorsport team. In addition, all of them have been involved in running the vehicle safely on multiple racetracks and finding the optimal setup for the competition runs. Boris Lohmann contributed to the critical revision of the manuscript and the conception of the research project.

A Tube-MPC Approach to Autonomous Multi-Vehicle Racing on High-Speed Ovals

Alexander Wischnewski, Thomas Herrmann, Frederik Werner, and Boris Lohmann

***Abstract*—Autonomous vehicle racing has emerged as vibrant and innovative technology development and demonstration platform in recent years. Universities and companies demonstrate their achievements on various vehicles - from 1:10th to full-scale prototypes. One of those platforms is the Dallara AV-21, the spec-vehicle for the Indy Autonomous Challenge. This paper outlines the robust model predictive control (MPC) concept used within the software stack of the TUM Autonomous Motorsport team. It is based on a simplified friction-limited point mass model and a set of low-level feedback controllers. The remaining model uncertainties are managed via introducing a constraint-tightening approach based on a Tube-MPC approach. In contrast to classical tracking controllers, the optimization problem is formulated to freely optimize the trajectory while staying within certain maximum deviations of the reference. This approach allows to rely on a coarse output of the trajectory planning approach while maintaining smoothness requirements in steering, throttle, and brake actuation.**

The paper highlights the advantages of the proposed robust reoptimization concept compared to pure tracking formulations. It showcases the performance compared to a classical LQR controller and an MPC, which utilizes a vehicle model with a more sophisticated tire model. The controller achieved a top speed of $265\,\mathrm{km\,h^{-1}}$ and lateral accelerations up to $21\,\mathrm{m\,s^{-2}}$ during a two-vehicle competition involving dynamic overtaking maneuvers on the Las Vegas Motor Speedway, a famous racetrack with turns banked up to 20°.

***Index Terms*—MPC, robust, control, vehicle dynamics, autonomous driving.**

I. Introduction

CONTROLLING a vehicle safely at the limits of its physical capabilities is a key requirement to enable full autonomy even in the most challenging scenarios. The main challenges from a control systems perspective are the strong coupling of longitudinal and lateral vehicle dynamics, the nonlinearities in the tire behavior, and the apparent uncertainty with respect to tire-road friction, actuator dynamics, and environmental influences. From an engineering point of view, these are accompanied by the difficulty to recreate and test scenarios that require a vehicle to operate at its limits. These are usually tried to be circumvented and therefore rare.

All of those challenges can be found in autonomous racing, which is therefore an ideal benchmark environment for control algorithms which target these problems. A team from the Technical University of Munich entered the Indy Autonomous Challenge [1], a university competition aiming at advancing the state-of-the-art in autonomous vehicle research (see Fig. 1). This paper is focused on the motion control algorithm of the software stack, which won the inaugural Indy Autonomous Challenge at the Indianapolis Motor Speedway on October 23rd 2021 and placed second in the passing competition with two vehicles at the Las Vegas Motor Speedway on January 7th 2022.

II. Related Work

A. State-of-the-Art

The field of autonomous vehicle control is well established since the successful application of multiple control strategies during the DARPA autonomous driving challenges [2], [3] as well as the availability of several series-level driver assistance systems. A survey covering the key concepts used in the field is presented in [4]. In recent years, several researchers have turned towards the application of these algorithms in racing scenarios [5]–[7]. These combine the unique opportunity to benchmark algorithms close to the limit of the vehicle's capabilities while being inherently safe at the same time.

The vehicle dynamics near the handling limits show significant nonlinearities in the tire behavior and strong coupling between the longitudinal and lateral vehicle motion. The concepts applied in [5] and [7] demonstrate successful racing on a full-size prototype using a linearized system description and adjust the linear lateral feedback controller using a gain-scheduling strategy based on the vehicle velocity. An enhancement to those approaches is the utilization of exact linearization-based controller design [8] methods; however,

Fig. 1. The race vehicles driven by the autonomous software stacks of the TUM Autonomous Motorsport and the PoliMOVE teams at the passing competition of the Indy Autonomous Challenge at January 7th 2022 with speeds above $260\,\mathrm{km\,h^{-1}}$. Credits: Indy Autonomous Challenge.

Research was supported by the basic research fund of Technical University of Munich.

A. Wischnewski and B. Lohmann are with the Chair of Automatic Control, School of Engineering & Design, Technische Universität München, München, Germany. (email: {alexander.wischnewski, lohmann}@tum.de

T. Herrmann and F. Werner are with the Institute of Automotive Technology, School of Engineering & Design, Technische Universität München, München, Germany. email: {thomas.herrmann, frederik.werner}@tum.de

those perform an implicit decoupling and linearization of the multivariable control task and can therefore not fully exploit potential advantages of the nonlinear system dynamics. Furthermore, they are usually prone to parametric as well as dynamic uncertainties. A comprehensive comparison of other nonlinear control techniques which do not require online optimization can be found in [9]. All of them share the common disadvantage, that they do not explicitly handle constraint satisfaction in the controller design. This leaves the consideration of tire and position deviation constraints to an a-priori specified worst-case safety margin for the trajectory planning algorithms.

In contrast to the already mentioned approaches, MPC promises to handle the multivariable, constrained control problem via the repeated solution of a finite-horizon optimization problem during the vehicle's operation. The first systems of this type dealt with the lateral and longitudinal control problem independently [10]. However, this approach does not allow to leverage many of the strengths of MPC concepts. The majority of the approaches therefore jointly optimize the lateral and longitudinal vehicle dynamics at the expense of larger optimization problems and higher computational burden. They can be grouped according to the class of models used: Point-mass models offer a simplistic view of the dynamics while still capturing the major dynamics in the euclidean space. If they are used at the handling limits of the vehicle, they are enhanced with friction limits to account for the physical limitation of the tires [11]. More complex models usually use single-track models with varying fidelity of the tire model [6], [12], [13]. A common choice are Pacejka [14] models with a coupling via a constraint on the maximum combined tire force. The influence of the model choice on the controller capability is analyzed by [15]. It was found that the choice of vehicle model does not significantly influence the optimal racing trajectory. Still, the more complex models can consider and stabilize the transient yaw dynamics. However, at the same time, the computational complexity grows significantly due to the additional state variables and constraints which have to be considered. In addition to handling multi-input dynamics and constraints, a key strength of MPC approaches are applications where a reoptimization around an initial reference path is allowed and required [6], [16], e.g., to consider sudden obstacles or to achieve smoother input behavior. This property will also be exploited in this paper to reduce the smoothness requirements for the output of the motion planning algorithms. Even more complex vehicle dynamics descriptions, e.g., including the banking and inclination effects of racetracks, have been used for minimum lap-time optimal control problems but have not yet shown to be real-time capable [17].

The case of autonomous racing holds a unique position in these approaches as it aims at minimizing the travel time around the circuit instead of tracking a given trajectory. This requirement can be approached using minimum-time optimization [18], by reformulating the control objective to maximize the progress along a given reference track [6], [19] or via a two-level approach where an appropriate reference trajectory generation takes care of the overall aim and a high fidelity nonlinear MPC is applied to track this trajectory [6], [20].

A challenge for all of those approaches is to generate a recursively feasible formulation of the finite-horizon optimization problem. From a practical perspective, this translates into designing the optimization problem such that it applies sufficient caution to manage effects that affect the behavior after the considered optimization horizon. A theoretically rigorous solution to this problem would be the construction of a positive invariant set [21]. However, this is a yet unsolved problem in the case of arbitrary trajectory tracking with nonlinear vehicle models. Promising directions towards this aim are given via offline computation of a discriminating kernel for a given track [22] and via the usage of simplified vehicle models [23]. A way to circumvent this issue is the utilization of sufficiently long optimization horizons and the application of slack variables to soften the state constraints [18], [19].

Similar issues arise in reality due to the inevitable model inaccuracies and external disturbances. From a control-theoretic perspective, they can be resolved by thoroughly extending the recursive feasibility analysis framework to a robust version considering those uncertainties, either in a deterministic or a stochastic modeling framework. Solutions for linear models are well-known [24], [25] and extensions for nonlinear models are coming up in recent years [26]. While there already exist several applications for autonomous driving in general [27]–[29], the application to racing vehicles has not found widespread adoption so far. This is likely due to the conservative nature of thorough theoretical solutions for high-dimensional systems. Notable exceptions have been the approach proposed by [30], which models uncertainty in a stochastic framework and manages to reduce the accident rate while keeping a similar performance level, [31], which leverages constraint tightening to handle the transient dynamics of an auxiliary vehicle dynamics learning algorithm, and [32], which uses a similar approach to introduce tuning parameters for managing caution with respect to constraint exploitation in a structured way.

B. Scope & Outline of the Paper

In this paper, we extend the Tube-MPC scheme presented in [32] to match the requirements of high-speed oval racing. First, we allow for varying acceleration limits due to the large speed range (depending on the aerodynamic downforce) and propose a strategy to handle the three-dimensional layout of the race track, i.e., the effects on the vehicle dynamics due to banked turns. Second, we propose a cost design to smoothen coarse target trajectories to improve driving stability while simultaneously adhering to the specified constraints. In contrast to a classical trajectory tracking controller, this concept can handle non-continuous curvature and longitudinal accelerations of the target trajectory without an impact on the driving performance. Third, we present results of a full-scale vehicle racing in scenarios with two vehicles (including dynamic replanning) with speeds up to $265\,\mathrm{km\,h^{-1}}$ and accelerations of $21\,\mathrm{m\,s^{-2}}$. Additionally, we perform a simulation study to compare the proposed friction limited point-mass model Tube-MPC with an MPC scheme based on a nonlinear tire model and a classical LQR controller and elaborate on the importance of the underlying acceleration controllers.

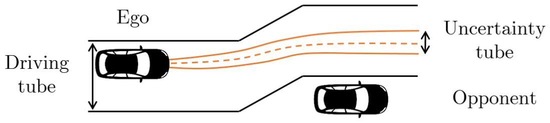


Fig. 2. Visualization of the Tube-MPC concept with reoptimization capabilities. The nominal prediction (dashed-orange) of the MPC is replaced with a set-valued prediction representing the potential uncertainty within the model (solid-orange). The controller is allowed to deviate from the middle of the driving tube but considers the constraints for all potential future outcomes of the system (black).

The paper is structured as follows: The following section presents essential preliminaries and the vehicle modeling applied in this work. Section IV explains the proposed Tube-MPC scheme. An extensive study of different algorithm configurations in simulation is presented within section V. The successful application of the proposed algorithm on a full-scale prototype at high speeds is presented in section VI. The conclusions drawn can be found in section VII.

III. Methodology

A. Software Architecture

The overall driving task within the autonomous racing software stack is split into three areas: perception, planning, and control [33]. We omit a detailed discussion for the perception and the planning part and only cover the aspects that are important to understand the following implications for the control system. The perception part of the software stack builds upon multiple independent object detection pipelines using LIDAR, Radar and Camera. These are fused via an object-tracking algorithm based on a Kalman-Filter approach. Based on those estimates, a prediction algorithm forecasts what the other vehicles on track are going to do. The planning problem is difficult to be efficiently solved due to the combinatorial nature resulting from other vehicles on the track [6]. Intuitively, this reflects the fact that there are multiple local minima [34], e.g., passing a vehicle in front either on the left or on the right side. We dedicate the task of this high-level decision problem to the trajectory planning algorithm and will use a graph-based optimization framework for this (see also the discussions on this in [6], [35]). The result of this task is an admissible driving tube depicted in black in Fig. 2.

The control algorithm generates appropriate vehicle control commands in the form of throttle, brake, and steering setpoints. It can achieve its maximum potential when it is allowed to freely optimize within the admissible driving tube. In contrast to pure tracking control, this paradigm change makes it possible to reduce the quality requirements for the trajectory planning algorithm. Instead of delivering a fine-tuned target trajectory, the planning algorithm aims to make behavioral decisions that are refined within an MPC algorithm. In addition, this allows the planner to cover longer planning horizons at the same computational budget, which is essential for high-speed operation.

The ability to exploit the admissible driving tube poses challenges when trying to guarantee that the MPC algorithm stays recursively feasible as the solutions to the optimization problem tend to be close to the constraints when the lateral deviation cost is omitted or very small. This issue arises from the fact that nominal control algorithms are prone to becoming infeasible in the presence of model uncertainties and disturbances [25]. This is countered by the explicit consideration of these in the design of the controller via the design of a Tube-MPC scheme instead of a nominal MPC. The key idea is to replace the predicted trajectory with a prediction of a set of possible trajectories resulting from the impact of uncertainties and disturbances (depicted in orange in Fig. 2). This creates a control behavior that exploits the constraints at the beginning of the optimization horizon and stays cautious towards the end.

B. Tube-MPC

We consider the discrete, linear time-invariant dynamic system

$$x(k+1) = Ax(k) + Bu(k) + Ed(k), \tag{1}$$

where $x(k) \in \mathbb{R}^n$ is the state, $u(k) \in \mathbb{R}^p$ is the input and $d(k) \in \mathbb{R}^q$ is a bounded, additive disturbance term. The index k refers to the current time instant. The system is subject to safety constraints

$$u \in \mathbb{U}, \tag{2a}$$

$$x \in \mathbb{X}. \tag{2b}$$

The nominal linear MPC controller is defined via the iterative solution of the following optimization problem

$$\min_{\bar{u}} \sum_{i=0}^{N-1} \left(x_i^{\mathrm{T}} Q x_i + u_i^{\mathrm{T}} R u_i\right) + x_N^{\mathrm{T}} P x_N \tag{3a}$$

$$\text{s.t.} \quad x_0 = x(k) \tag{3b}$$

$$x_{i+1} = Ax_i + Bu_i, \quad \forall i \in [0, N-1] \tag{3c}$$

$$x_i \in \mathbb{X}, \quad \forall i \in [0, N-1] \tag{3d}$$

$$u_i \in \mathbb{U}, \quad \forall i \in [0, N-1] \tag{3e}$$

$$x_N \in \mathbb{X}_{\mathrm{T}} \tag{3f}$$

where x_i and u_i denote the predicted state and input at the discretization point i of the optimal control problem and the matrices Q, R and P are weighting matrices of appropriate dimensions. Note that the index i is used for the dynamics on the prediction horizon, while k is the actual time index. Stability and recursive feasibility can be ensured via the choice of the terminal set $\mathbb{X}_{\mathrm{T}}$ and the terminal cost $x_N^{\mathrm{T}} P x_N$. We use the notation $\bar{u}$ whenever we refer to the vector formed by the candidate input trajectory u_0 to u_{N-1}. The control law itself is given via $u(x(k)) = u_0$.

It is well known [25] that the application of this nominal MPC scheme (see [21] for a thorough discussion of constrained nominal MPC) does not guarantee recursive feasibility of the optimization problem in the presence of the unknown disturbance $d(k)$. This can be prevented by the application of Tube-MPC, where the problem (3) is modified to ensure that the constraints are satisfied robustly. The concept utilized in this work is based upon the work of [24] and will only be introduced briefly here for the sake of brevity. First, a pre-stabilizing feedback law $u = -Kx$ is introduced which is then used to predict the reachable sets $\mathbb{X}_{R,i}$ at the discretization points i for the resulting closed-loop system with respect to the external input $d(k)$. These reachable sets are now utilized to calculate tightened state and input constraints $\bar{\mathbb{X}}_i \subset \mathbb{X}$ and $\bar{\mathbb{U}}_i \subset \mathbb{U}$ by applying the pontryagin set difference (see [24]), resulting in the new optimization problem

$$\min_{\bar{u}} \sum_{i=0}^{N-1} \left(x_i^T Q x_i + u_i^T R u_i\right) + x_N P x_N \tag{4a}$$

$$\text{s.t.} \quad x_0 = x(k) \tag{4b}$$

$$x_{i+1} = A x_i + B u_i, \quad \forall i \in [0, N-1] \tag{4c}$$

$$x_i \in \bar{\mathbb{X}}_i, \quad \forall i \in [0, N-1] \tag{4d}$$

$$u_i \in \bar{\mathbb{U}}_i, \quad \forall i \in [0, N-1] \tag{4e}$$

$$x_N \in \bar{\mathbb{X}}_T \tag{4f}$$

The implementation of this tightening will be done via calculation of the set dynamics based on ellipsoids [36] for the linearized system dynamics. Note that this approach can be further justified by the successful application of controllers obtained from the linearized dynamics for real-world autonomous racing [5]. The reachable sets around the nominal trajectory p are described via ellipsoids of the form

$$E(p, M) := \left\{x \in \mathbb{R}^n | (x-p)^T M^{-1} (x-p) \leqslant 1\right\}, \tag{5}$$

with p being the center of the ellipsoid and M the corresponding shape matrix. An over-approximation of the Minkowski sum of two ellipsoids [37] is given by

$$\begin{aligned} &E(p_1, M_1) \oplus E(p_2, M_2) \\ \subset &E\left(p_1 + p_2, \left(1 + c^{-1}\right) M_1 + (1 + c) M_2\right), \end{aligned} \tag{6}$$

with $c = \sqrt{\operatorname{Tr}(M_1)/\operatorname{Tr}(M_2)}$ and $\operatorname{Tr}(M)$ the trace of the Matrix M. Additionally, the following definition for the affine dynamics is used

$$A \cdot E(p, M) + b = E(Ap + b, AMA^T). \tag{7}$$

IV. Application to Autonomous Vehicle Racing

A. Controller Concept

The results obtained under real-world racing conditions for a point-mass-based control design in [5], [11] demonstrate that simple dynamic models enable sufficiently accurate control as long as the tire constraints are considered in the form of suitable acceleration constraints during the planning process. This idea has been leveraged for the design of a Tube-MPC control concept in [32] and has shown promising results for a classical trajectory tracking formulation. The following section will extend the controller presented in [32] by a detailed discussion of the modifications and extensions required to design a fully functional control system under real-world conditions at one of the fastest oval racetracks in the world.

The fundamental philosophy of the concept presented is a strict split into certain and uncertain dynamics. We consider the relation between the accelerations in the vehicle frame and the actual movement to be exact, as they are based on theoretically sound differential equations without parametric uncertainties. This setting is well-suited for handling the dynamics and acceleration constraints in an MPC scheme. In contrast, the effect of throttle, brake pressure, and steering requests on the actual accelerations is subject to uncertain dynamics (such as turbocharger effects, neglected actuator dynamics, etc.) as well as parametric uncertainties (steady-state errors of the steering actuator due to load on the tires, engine calibration changes compared to the setup measured on the dynamometer, etc.). We handle this situation by introducing low-level PID-feedback loops for the longitudinal and lateral acceleration and a steering feedback loop. This design is favorable as it reduces the remaining uncertainties via fast and simple-to-tune control loops without increasing the complexity of the prediction model in the MPC. The remaining uncertainties and tracking errors are assumed to be external disturbances and considered by extension of the MPC scheme to a Tube-MPC scheme. The resulting controller architecture is depicted in Fig. 3.

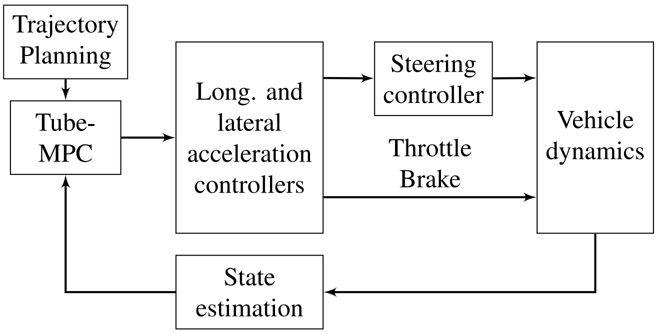


Fig. 3. Block diagram of the proposed Tube-MPC algorithm

The interfaces to the software stack are the trajectory planning and the state estimation. The former provides a path with continuous position and heading and a continuous velocity. The curvature and longitudinal acceleration do not have to be continuous, this is an advantage of the applied reoptimization concept (see Sec. III-A). The state estimation provides a localization in global coordinates and velocity and acceleration estimates. This component will be introduced briefly in Sec. IV-E and is based on the work done in [38].

B. Formulation of the Optimization Problem

The upcoming section summarizes the detailed derivation of the vehicle dynamics and its constraints presented in [32]. Using a model derivation procedure in curvilinear coordinates as in [6], [19] (also see 4) and a point-mass model to describe the behavior of the vehicle [11] we can write

$$\begin{bmatrix}\dot{s}\\ \dot{v}\\ \dot{d}\\ \ddot{d}\end{bmatrix} = \begin{bmatrix} \left(1 - d\kappa(s)\right)^{-1} v\cos(\Delta\psi) \\ \frac{F_x}{m} - \frac{c_w}{m} v^2 \\ \dot{d} \\ \left(\frac{F_x}{m} - \frac{c_w}{m} v^2\right)\sin(\Delta\psi) + \cos(\Delta\psi)\left(a_y - \kappa(s)\dot{s}v\right) \end{bmatrix}, \tag{8}$$

with

$$\Delta\psi = \arcsin\left(\frac{\dot{d}}{v}\right). \tag{9}$$

The state s represents the progress along the reference path with curvature $\kappa(s)$, while d depicts the lateral deviation and $\Delta\psi$ the orientation error of the chassis towards this path. We use the coordinate transformation $\dot{d} = v\sin(\Delta\psi)$ to replace the orientation error with the derivative of the lateral error $\dot{d}$. The longitudinal dynamics are completed via the vehicle speed v, the longitudinal force input F_x, the vehicle mass m and the lumped drag coefficient c_w. This formulation emphasizes the interpretation of the longitudinal and lateral accelerations as the key variables to influence the motion of the vehicle and will be useful during the efficient realization of the embedded optimization algorithm.

Investigations with high speed autonomous racecars have shown that circular limits for the tire forces based on Kamm's circle do not necessarily transform directly into reliable acceleration limits on the vehicle level [39]. The actual shape of the vehicle acceleration limits depends on several factors, such as the tire characteristics and the mechanical vehicle setup. We use a diamond-shaped constrained set

$$-1 \leqslant \frac{a_x}{\bar{a}_x} + \frac{a_y}{\bar{a}_y} \leqslant 1 \tag{10a}$$

$$-1 \leqslant \frac{a_x}{\bar{a}_x} - \frac{a_y}{\bar{a}_y} \leqslant 1 \tag{10b}$$

with the acceleration limits $\bar{a}_x$ and $\bar{a}_y$ in the following as it allows to exploit the pure lateral and longitudinal vehicle capability while being cautious in the combined cases. This behavior delivered a good compromise between performance and reliability, while being easy to integrate in numerical algorithms due to its polytopic form. The limits will be obtained based on the speed of the target trajectory for each discretization point, similar as the curvature of the path, from a look-up table within the trajectory planner. This table is generated based on extensive simulations as well as real-world vehicle tests. In addition to the acceleration constraints, the trajectory planner generates the lateral deviation limits dynamically from the situation. They are implemented via simple box constraints and can be different between the left and the right side of the driving tube.

$$d^- \leqslant d \leqslant d^+. \tag{11}$$

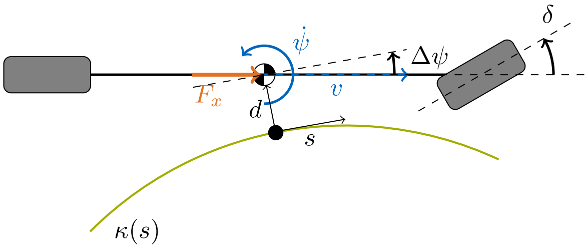


Fig. 4. Visualization of the curvilinear coordinate system and the dynamic model used to describe the vehicle behavior.

The actual MPC problem is formulated with respect to the target velocity profile v_R using the state transformation $\Delta v = v - v_\mathrm{R}$ and the input transformations

$$F_x = c_\mathrm{w}v^2 + m\frac{v\cos(\psi_\mathrm{P})}{1-d\kappa(s)}\frac{\partial v_\mathrm{R}(s)}{\partial s} + m\Delta a_x, \tag{12a}$$

$$a_y = \kappa(s)\frac{v^2\cos(\psi_\mathrm{P})}{1-d\kappa(s)} + \frac{1}{\cos(\psi_\mathrm{P})}\left(\Delta a_y - a_x\sin(\psi_\mathrm{P})\right). \tag{12b}$$

In addition, we assume the progress along the reference line $\dot{s}$ and curvature values $\kappa(s)$ to be known a-priori to the optimization problem and can therefore drop the state s from the state vector and simplify the optimization problem to use the following system dynamics:

$$\begin{bmatrix}\Delta\dot{v}\\ \dot{d}\\ \ddot{d}\end{bmatrix} = \begin{bmatrix}0&0&0\\0&0&1\\0&0&0\end{bmatrix}\begin{bmatrix}\Delta v\\ d\\ \dot{d}\end{bmatrix} + \begin{bmatrix}1&0\\0&0\\0&1\end{bmatrix}\begin{bmatrix}\Delta a_x\\ \Delta a_y\end{bmatrix}. \tag{13}$$

This linear dynamic system allows us to formulate the robust MPC problem in the form of (4). We add the constraints (10) and (11) via linearization of (12). The uncertainty tube sets for the constraint tightening are calculated using ellipsoids and a pre-stabilizing controller based on a reasonable LQR weight design [5], [32]. We assume the uncertainties to enter the dynamics as matched disturbances, i.e., via acceleration disturbances acting the same way as Δa_x and Δa_y. They will serve as a tuning parameter to blend between an aggressive driving style (small uncertainties) and a rather cautious behavior (large disturbances) and are gain-scheduled based on the vehicle speed. The linearization is done based on a velocity profile obtained from a weighted linear combination of the last steps solution and the target velocity profile. Incorporating the latter allows the MPC to adjust quickly to a change in target trajectory that frequently occurs in multi-vehicle racing scenarios.

The system is discretized using Euler forward discretization and a sample rate of 60 ms. The horizon is chosen to be 40 steps which leads to an overall prediction time of 2.4 s. The QP is implemented using OSQP [40], an alternating direction method of multipliers (ADMM)-based first-order solver. The highly constrained nature of the optimization problem suits the properties of the solver using a projection approach to handle constraint satisfaction [39]. Its efficient implementation allows to execute it at an update rate of 100 Hz.

The cost function design is based on the choice of the weighting matrices Q and R. We tune the weights for the longitudinal dynamics to reflect a very gentle tracking behavior with low convergence rates to enable smooth vehicle inputs while still keeping tracking of the target velocity profile. This is important to handle situations where two vehicles closely follow each other. The lateral dynamics weights are tuned to minimize the lateral motion in the driving tube and therefore only consider the lateral deviation very briefly. In addition, we introduce a regularization term penalizing the acceleration differences between $a_{x,i}$ and $a_{x,i+1}$ as well as $a_{y,i}$ and $a_{y,i+1}$ between the discretization points to further smooth the vehicle behavior.

C. Low Level Controllers

The lateral and longitudinal acceleration controllers serve an important task in the proposed control concept: They ensure that the assumptions used by the Tube-MPC are met by the vehicle dynamics, even in the presence of model uncertainty and external disturbances.

The lateral acceleration controller consists of three main parts: a feedforward element based on the kinematic relations of the vehicle, a proportional feedback, and an integral feedback part. The resulting control equation can be summed up to

$$\delta = \frac{1}{v^2}\left(a_y l + K_{\mathrm{p,y}}\tilde{a}_y + \delta_{\mathrm{i}}\right), \tag{14a}$$

$$\dot{\delta}_{\mathrm{i}} = K_{\mathrm{i,y}}\tilde{a}_y - \left(\frac{\delta_{\mathrm{i}}}{c_1 \bar{a}_y}\right)^3 \tag{14b}$$

with the controller gains $K_{\mathrm{p,y}}$ and $K_{\mathrm{i,y}}$ and the lateral acceleration control error $\tilde{a}_y$. The second element in the integrator equation above serves as a soft limiter for the integral controller part. It prevents the controller from winding up in case of significant understeer, which would lead to degraded control performance on turn out. This limit depends on the absolute acceleration limit $\bar{a}_y$ as well as a tuning factor c_1 to adjust the margin where the integral part can compensate for deviation from a neutral vehicle behavior. The above equations are slightly modified for low-speed scenarios (close to walking speed) to limit the resulting gain and prevent oscillations via the introduction of a lower speed limit for the evaluation of the equations. The steering controller is built based on a classical PI-controller with anti-windup mechanisms. The details are omitted for the sake of brevity. It allows to shape the behavior of the steering actuator freely as the internal control loops are not exposed on the available interface.

The longitudinal acceleration controller is built slightly simpler: a feedforward element based on the vehicle mass and drag resistance and a proportional feedback term. This removes the necessity for an anti-windup strategy and simplifies the tuning procedure. The control equation can be written as

$$F = ma_x + c_{\mathrm{w}}v^2 + K_{\mathrm{p,y}}\tilde{a}_x \tag{15}$$

with the control gain $K_{\mathrm{p,y}}$, the longitudinal acceleration control error $\tilde{a}_x$ and the lumped drag coefficient c_{w}. The resulting force is converted to a corresponding throttle position or brake pressure requests by using empirical lookup tables.

D. Banking Influence

One of the challenges for vehicle dynamics control is the influence of the track banking on the required steering angle and the measured accelerations. While this has been handled in optimal control problems for laptime minimization several times [17], it has not yet been considered during real-time-critical implementations. As this would complicate the resulting optimal control problem significantly, we propose to handle these effects on the level of the acceleration controllers. To calculate the required transformation we use the relation of the acting forces from Fig. 5. Using the balance of forces in each of the two dimensions, we can write

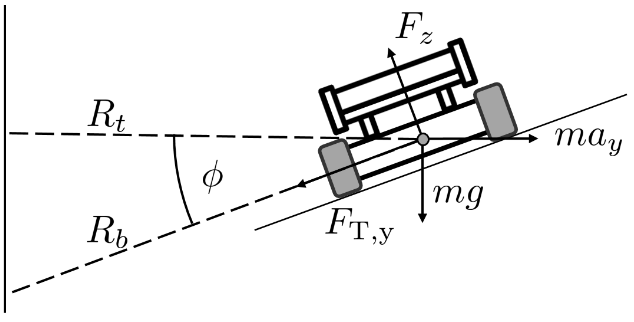


Fig. 5. Forces acting upon the vehicle in a banked turn, with R_{b} being the banked turn radius and R_{t} being the radius in the 2D-plane.

$$\cos(\phi)\,F_{\mathrm{T,y}} + \sin(\phi)\,F_z = ma_y, \tag{16a}$$

$$-\sin(\phi)\,F_{\mathrm{T,y}} + \cos(\phi)\,F_z = mg, \tag{16b}$$

with ϕ being the banking angle. The IMU which is used for lateral acceleration feedback measures the specific force acting in locally upon the vehicle which can be calculated as $a_{\mathrm{IMU}} = \frac{F_{\mathrm{T,y}}}{m}$. Rearranging the above equation, we can calculate the corresponding effective acceleration in the level plane to be

$$a_y = a_{\mathrm{IMU}}\left(\cos(\phi) + \tan(\phi)\sin(\phi)\right) + \tan(\phi)\,g. \tag{17}$$

In addition, we have to modify the feedforward portion of the lateral acceleration controller to account for the increased turn radius R_{b} in the banked plane compared to the 2D track plane R_{t}. This can be done by modifying the first term of (14) to be $a_y l\cos(\phi)$ instead of $a_y l$.

E. State Estimation

The proposed Tube-MPC requires the determination of the lateral error, the lateral error derivative and the velocity error. All of these quantities are based upon a reliable estimation of the vehicle position and motion state in the euclidean frame. To achieve this, we use a Kalman-Filter-based concept with similar model assumptions as the proposed Tube-MPC. Details on this can be found in [38]. However, as this approach is based on a two-dimensional vehicle model and relies heavily on the IMU signals, the banking has a significant influence on the estimation quality. The transformations presented above are therefore also applied to the inputs of this estimator to compensate for these effects.

The lateral error is calculated based on a matching algorithm of the current position with respect to the path in a two dimensional euclidean coordinate frame. First, the closest section of the target path is determined. Based on this, a refined estimate is calculated from a linear projection onto this path. The lateral error derivative is determined by simple numerical derivation and low-pass filtering. A heading and side slip angle based approach as proposed by [5] did not show satisfactory results, as the side slip angle could not be estimated with sufficient quality (sub-degree accuracy) on the used vehicle platform.

V. Simulation Results

A. Simulation Setup

The following section evaluates the performance of the presented algorithm in comparison to an LQR controller [5] and a wide-spread MPC formulation in path coordinates using a nonlinear Pacejka tire model (based on [19]). It further conducts a parameter study to clarify the sensitivity and generate guidelines for tuning the controller on the real vehicle.

The simulation environment is based on a nonlinear dual-track model with vehicle parameters aligned with an exemplary vehicle, including actuator and sensor models. The model and the TMPC implementation are available at [41]. All results in this section are generated using a simplified version of the trajectory planning algorithm and the full state estimation and control software required for running the real-world vehicle. The Tube-MPC is implemented according to the details provided in Sec. IV-B and achieves runtimes of approx. $1\,\mathrm{ms}$ to $4\,\mathrm{ms}$ on a standard Core i7 laptop with $2.7\,\mathrm{GHz}$ and total control software execution rate of $100\,\mathrm{Hz}$. The MPC with the nonlinear tire model is implemented using the C-Code generation framework of the library acados [42] and achieves runtimes of approx. $20\,\mathrm{ms}$ to $40\,\mathrm{ms}$ with a discretization of $40\,\mathrm{ms}$ and, similar to the Tube-MPC, $2.4\,\mathrm{s}$ prediction horizon. To account for the larger computational burden, we run this version with an update frequency of $25\,\mathrm{Hz}$.

B. Evasion Maneuver

One of the main advantages of MPC is the rigorous handling of input and state constraints in situations where feedback is necessary to ensure that the system behaves as expected. This becomes especially important for the proposed Tube-MPC concept. It does not aim at perfect lateral tracking but instead focuses on reoptimizing the target trajectory presented by the local trajectory planner. A common challenge for the proposed software architecture (see Fig. 3) is the frequent update and modification of the target trajectory generated by the local trajectory planner. Every update of the perceived opponents and their predicted behavior may lead to a behavior change in the local trajectory planning. e.g., aborting an overtaking maneuver as it might not be feasible anymore or initiating an overtaking maneuver as an obstacle was detected nearby. These scenarios are difficult to reproduce as the ego vehicle actions directly influence the behavior of other vehicles on the track during multi-vehicle simulations. We therefore propose a double-lane change maneuver to mimic the situation that a relatively slow opponent vehicle is perceived in front of the ego-vehicle. We compare two scenarios: In the first scenario, the opponent vehicle is perceived at about $50\,\mathrm{m}$ in front and an evasion maneuver has to be initiated immediately. While this might seem like an extreme scenario, the significant banking of the racetrack and the low height of the vehicles make this a realistic worst-case assumption. In the second scenario, the opponent vehicle is perceived at about $70\,\mathrm{m}$ in front, which gives the controller an additional foresight horizon to react much smoother.

The first scenario is depicted in Fig. 6 and the top plot of Fig. 8. The target trajectory switches to the lane change

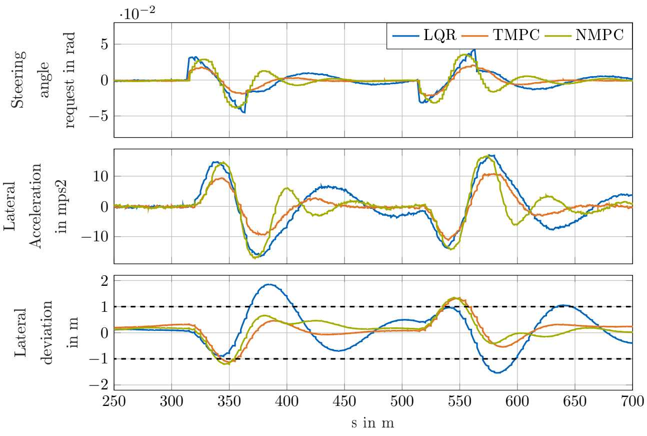


Fig. 6. Control performance during an evasive maneuver with $50\,\mathrm{m\,s^{-1}}$ with no foresight on the lane change maneuver.

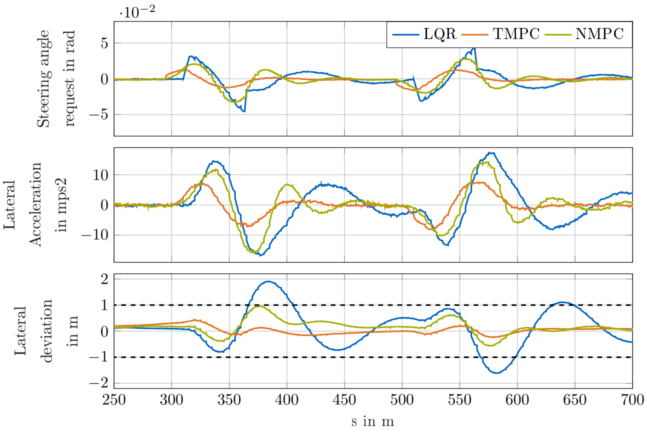


Fig. 7. Control performance during an evasive maneuver with $50\,\mathrm{m\,s^{-1}}$ with $20\,\mathrm{m}$ foresight (equivalent to $400\,\mathrm{ms}$) on the lane change maneuver.

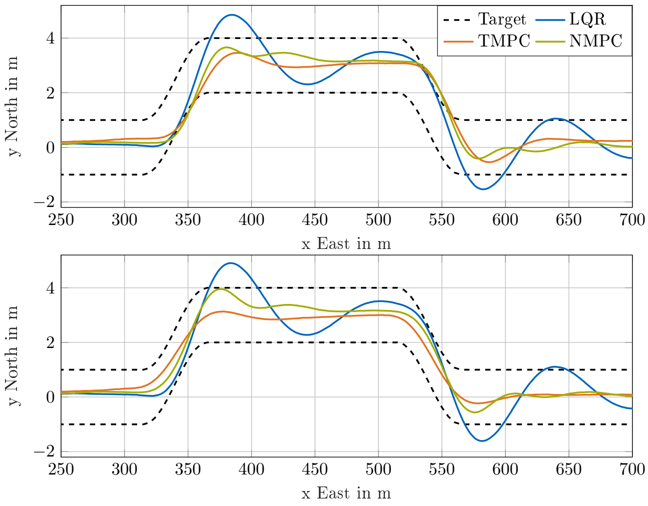


Fig. 8. Visualization of the motion control performance during an evasive maneuver with $50\,\mathrm{m\,s^{-1}}$ within the cartesian coordinate frame. The top plot shows the performance when the lane changes are not known beforehand to the software. The bottom plot depicts the improvements in performance for a foresight horizon of $20\,\mathrm{m}$ roughly equivalent to $400\,\mathrm{ms}$.

trajectory when the vehicle moves past the $320\,\mathrm{m}$ mark and it has to react immediately. The same happens at the $520\,\mathrm{m}$ mark. While all concepts have access to the exact curvature of the target path, the LQR controller fails to achieve the required control performance and to stay within the $1\,\mathrm{m}$ bounds of the target path even though it utilizes accelerations of $15\,\mathrm{m\,s^{-2}}$. The model predictive concepts (Tube-MPC and MPC with nonlinear tire model) improve upon constraint satisfaction. However, both of them still struggle to achieve the required performance. Multiple points can be observed here: First, the non-continuous curvature profile (which reflects the reduced quality requirements for the local trajectory planner) can be handled much better by the optimization-based control methods as they generate smoother steering angle behaviors. Second, the scenario itself is challenging from a system dynamics point of view. The delays and response times can only be partially overcome when the change to the trajectory is not known with sufficient foresight. Third, the Tube-MPC shows less aggressive steering interventions for longer durations than the MPC with a nonlinear tire model. This is because the MPC with a nonlinear tire model does not have an incentive to value driving stability and can therefore create the required lateral acceleration aggressively via the front axle. In conjunction with the inherent system uncertainties, this leads to an overshoot in lateral acceleration. While this might be mitigated with increased model accuracy, cost function tuning, and higher controller update frequency, the Tube-MPC controller achieves comparable lateral control performance at a fraction of those tuning efforts. It should also be noted that the Tube-MPCs incentive (stemming from the point-mass model) to focus on steady-state cornering scenarios exploits the vehicle stability properties and therefore increases the practical applicability.

This scenario is probably the most extreme - in most cases, the local trajectory planner is well aware of what's coming up, even though it still changes the target trajectory within the optimization horizon. The second scenario (see Fig. 7 and the bottom plot of Fig. 8) with an approximate foresight of $20\,\mathrm{m}$ (roughly equivalent to $400\,\mathrm{ms}$ foresight) is used here to demonstrate the advantages of the Tube-MPC concept. In this scenario, the same target path is used, but it is activated as soon as the vehicle passes the $300\,\mathrm{m}$ mark (the $500\,\mathrm{m}$ respectively for the second lane change). This foresight allows the model predictive controllers to start the lane change much more gently and therefore reduces the maximum lateral acceleration by approx. $25\,\%$ without sacrificing control performance. Again, the focus on steady-state cornering situations of the Tube-MPC leads to less aggressive steering interventions and smoother and more robust handling of the situation.

C. Importance of the Low Level Controllers

The proposed Tube-MPC concept builds around a simplified vehicle model (in contrast to other concepts in the literature, which tend to rely on more complex models for MPC [6], [31]) in conjunction with tight tracking of the target accelerations by low-level PID-like controllers. While this removes the need for sophisticated tire models, it becomes essential that the acceleration response of the vehicle matches the expectations of the model predictive controller. It should be noted that this requirement is even more pronounced as the Tube-MPC concept is heavily based on the idea of constraint satisfaction rather than target tracking. While a model/reality mismatch would only lead to slightly increased control values in a classical tracking concept, the violation of the underlying (uncertain) predictions of the vehicle motion almost certainly leads to significant violations of the state constraints and results in strong corrective actions by the Tube-MPC.

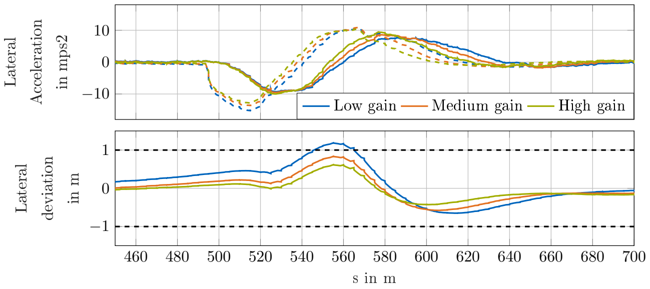


Fig. 9. Lateral acceleration controller influence during an evasive maneuver with $65\,\mathrm{m\,s^{-1}}$. The dashed lines depict target values while the solid lines depict actual measurements.

Fig. 9 depicts the influence of lateral acceleration control quality on the overall control system performance. The figure shows three different tunings from the low-level acceleration controller, ranging from a rather gentle tuning to a high-gain setup. The high-gain setup keeps the vehicle in the allowed driving tube with moderate corrections and shows better acceleration tracking. The comparison between the target accelerations calculated by the Tube-MPC controller between $500\,\mathrm{m}$ and $520\,\mathrm{m}$ shows significantly increased target accelerations for the gentle feedback setup ($15\,\mathrm{m\,s^{-2}}$ instead of $12\,\mathrm{m\,s^{-2}}$). This behavior can be explained by the fact that the vehicle approaches the lateral deviation constraint and violates it, which asks for strong corrective actions from the Tube-MPC. It should be noted that this is only possible as long as the acceleration constraints permit this increased feedback and therefore the low-level tracking becomes even more vital when the acceleration constraints are approached.

D. Influence of the Optimization Horizon

While the number of optimization variables is limited by the resulting computational load, the length of the optimization horizon can be adjusted independently from this via the discretization time. We run the optimization with an update rate of $100\,\mathrm{Hz}$ and vary the optimization horizon between $2.0\,\mathrm{s}$ and $2.8\,\mathrm{s}$. The results of this experiment are depicted in Fig. 10. As expected from a theoretical perspective, longer horizons lead to improved control quality via a decrease of the required lateral accelerations to move through the double lane change maneuver. While the lateral acceleration peaks at $10\,\mathrm{m\,s^{-2}}$ for the $2.0\,\mathrm{s}$ horizon, this is reduced to $8\,\mathrm{m\,s^{-2}}$ for the $2.8\,\mathrm{s}$ horizon. This improvement of $20\,\%$ is significant when driving at the handling limits of the vehicle. It should be kept in mind that these improvements stay only valid if the

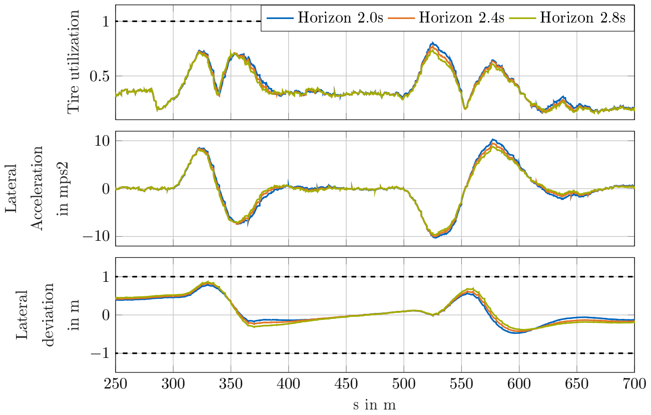


Fig. 10. Influence of the optimization horizon during an evasive maneuver with $65\,\mathrm{m\,s^{-1}}$

target trajectory does not change while the vehicle is driving the maneuver (see previous sections for a discussion of this issue) and longer horizons increase the computation time of the local trajectory planner. We choose $2.4\,\mathrm{s}$ to be a viable comprise between those targets as it still allows to cover $180\,\mathrm{m}$ of foresight horizon at top speed (approx. $75\,\mathrm{m\,s^{-1}}$)

VI. Experimental Results

A. Vehicle Platform

The autonomous race vehicle used is a modified Dallara Indy Lights vehicle, the Dallara AV-21. A drive-by-wire system actuates its steering and brakes, and its powertrain and gearbox can directly be actuated via the engine control ECU. The vehicle is equipped with two independent GPS units by Novatel, three LIDAR units by Luminar, three radar sensors by Aptiv, and six cameras by Allied Vision. In addition, standard vehicle dynamics sensors such as engine speed, wheel speeds, and brake pressures are available. The autonomous driving software runs on an ADLink x64 computer system based on an Intel Xeon with eight physical CPU cores and an NVIDIA RTX 8000 GPU.

The proposed controller requires strict real-time capability of the underlying computer platform. To enable this on the Ubuntu 20.04 based operating system, we allocate a dedicated CPU core, set appropriate scheduling properties, and move all file system interactions (e.g. data logging) from the vehicle motion control software to external, potentially asynchronous, software components. The rest of the software stack is interfaced via the standard robotics middleware ROS2 Galactic with Cyclone DDS as a transportation layer.

B. Single Vehicle Qualifying Competition

The following experiments have been conducted on the Las Vegas Motor Speedway Oval circuit. The track is $2440\,\mathrm{m}$ long and has a maximum banking of $20°$. The results of the single-vehicle qualifying run of the passing competition on January 7th are depicted in Fig. 11. It covered various speeds between $45\,\mathrm{m\,s^{-1}}$ and $70\,\mathrm{m\,s^{-1}}$, with two of those laps shown here. Interestingly, the lateral movement does not increase significantly with increasing lateral accelerations as it would be expected as a response to increasing uncertainty. In contrast, it can be seen that the controller leverages the available lateral movement to reduce the required control action to compensate for inaccuracies due to the nonlinearities in the model (see $500\,\mathrm{m}$ and $1600\,\mathrm{m}$). In classical tracking formulations, the controller would have added additional lateral acceleration when the vehicle is approaching the target of zero quickly. In the proposed concept, it allows to actually move beyond this target and starts to react only when it gets close to the actual constraint. This results in less aggressive steering corrections and therefore higher driving stability. Overall, this behavior is much closer to how a real driver would drive those turns. They focus on the overall task rather than strictly tracking artificial target trajectories.

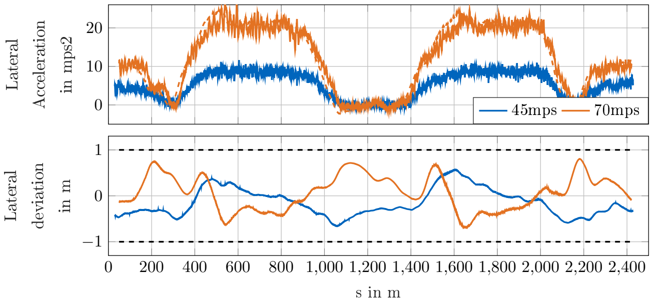


Fig. 11. Controller performance on the real world vehicle at various speeds. The dashed lines indicate target accelerations, while the solid ones indicate actual accelerations (transformed into the 2D plane).

C. Two Vehicle Passing Competition

In addition to the single-vehicle runs, the concept was proven during several experiments with two vehicles. The final competition run on January 7th has been conducted by the PoliMOVE and the TUM Autonomous Motorsport team (see Fig. 12 and [43] for a video) and includes multiple overtakes at varying speeds between $45\,\mathrm{m\,s^{-1}}$ and $74\,\mathrm{m\,s^{-1}}$. The acceleration plot shows that the trajectory planning does not consider jerk constraints or costs and is therefore prone to high gradients in lateral and longitudinal acceleration (Trajectory Target). The proposed Tube-MPC controller reduces those significantly (MPC Target in contrast to Trajectory Target), which results in a smooth vehicle behavior (Actual).

VII. Conclusions and Future Work

We presented a Tube-MPC concept for operating a full-scale autonomous racecar at speeds up to $265\,\mathrm{km\,h^{-1}}$ and lateral accelerations up to $21\,\mathrm{m\,s^{-2}}$ in single and two-vehicle scenarios. In contrast to classical tracking control concepts, we leverage the advantages of optimization-based control to allow coarse target trajectories and, therefore, increase the software stack's overall performance. This is made possible by explicitly considering the uncertainties within the prediction model, which generates a behavior that is cautious for long-term predictions but exploits the full vehicle potential on near-term predictions.

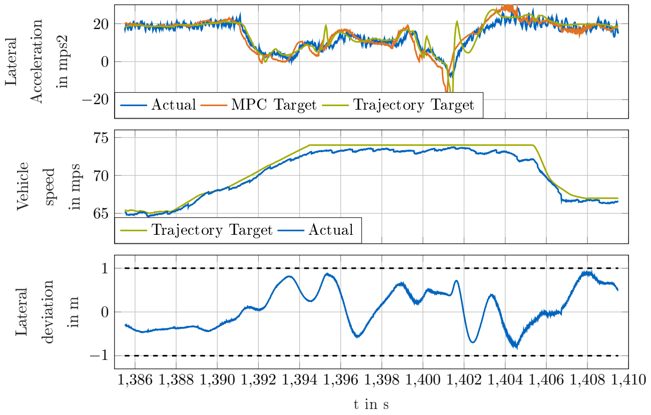


Fig. 12. Controller performance on the real world vehicle during the final competition run overtakes between PoliMOVE and TUM Autonomous Motorsport. Accelerations have been transformed into the 2D plane.

A notable feature of this strategy is that it lumps all uncertainties within a few uncertainty parameters and it's therefore sufficient to use a friction-limited point-mass model as dynamic model. This is especially helpful in the face of the significant banking influence on oval racetracks, as this is harder to handle appropriately in more sophisticated models with nonlinear tire and suspension effects. In addition, this strategy renders the concept independent from potential setup or parameter changes, e.g., a balance shift to make the vehicle more understeering.

While this paper demonstrated the huge potential of relatively simple dynamic models when combined with appropriate low-level controllers and thorough uncertainty handling, reaching super-human performance will likely require more advanced control strategies. An important improvement area is the controller's stabilization capabilities even when far away from the stable operating points of the state space (e.g., while drifting with large chassis side slip angles), as the proposed control concept relies heavily on the inherent stability of the vehicle. Even though this could be handled via more complex models [15], further work to include robustness properties and make it real-time capable has to be done to apply nonlinear tire models within an NMPC scheme for autonomous racing. Another promising direction is to leverage data-driven methods [31], [44] to refine the uncertain parts of the model while driving and exploit them more aggressively as soon as the models have converged. However, these algorithms remain challenging to implement in a real-time-capable manner and deploy to work reasonably well along all operating conditions of the vehicle.

Contributions and Acknowledgments

Alexander Wischnewski developed, analyzed and implemented the proposed control concept. Thomas Herrmann and Frederik Werner contributed to the real-time capable implementation and integration in the software stack of the TUM Autonomous Motorsport team. In addition, all of them have been involved in running the vehicle safely on multiple racetracks and finding the optimal setup for the competition runs. Boris Lohmann contributed to the critical revision of the manuscript and the conception of the research project.

We want to thank the whole TUM Autonomous Motorsport team, the Indy Autonomous Challenge organizers, Lauren and Craig Brooks, Kent Anderson, and Kris Kozak for the countless efforts to make the Indy Autonomous Challenge with multiple full-scale autonomous racing vehicles possible. Furthermore, we thank Matthias Rowold, Levent Ögretmen, and Joscha Bongard for the many discussions on the topic and comments on the manuscript.

References

[1] Energy Systems Network, "The official webpage of the indy autonomous challenge 2021 at the indianapolis motor speedway," 2021. [Online]. Available: https://www.indyautonomouschallenge.com

[2] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y.-W. Seo, S. Singh, J. Snider, A. Stentz, W. ". Whittaker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, J. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[3] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the DARPA grand challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[4] B. Paden, M. Cáp, S. Z. Yong, D. S. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, pp. 33–55, 2016.

[5] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, "Minimum curvature trajectory planning and control for an autonomous race car," *Vehicle System Dynamics*, vol. 58, no. 10, pp. 1497–1527, 2019.

[6] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2014.

[7] N. R. Kapania and J. C. Gerdes, "Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling," *Vehicle System Dynamics*, vol. 53, no. 12, pp. 1687–1704, 2015.

[8] M. Werling, "Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien," Ph.D. dissertation, 2011.

[9] D. Calzolari, B. Schurmann, and M. Althoff, "Comparison of trajectory tracking controllers for autonomous vehicles," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017.

[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*, vol. 15, no. 3, pp. 566–580, 2007.

[11] J. K. Subosits and J. C. Gerdes, "From the racetrack to the road: Real-time trajectory replanning for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 309–320, 2019.

[12] A. Katriniok, J. P. Maschuw, F. Christen, L. Eckstein, and D. Abel, "Optimal vehicle dynamics control for combined longitudinal and lateral autonomous vehicle guidance," in *2013 European Control Conference (ECC)*. IEEE, 2013.

[13] E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia, "Autonomous racing using linear parameter varying-model predictive control (LPV-MPC)," *Control Engineering Practice*, vol. 95, p. 104270, 2020.

[14] H. B. Pacejka, *Tire and vehicle dynamics*. Amsterdam: Elsevier, 2012.

[15] J. K. Subosits and J. C. Gerdes, "Impacts of model fidelity on trajectory optimization for autonomous vehicles in extreme maneuvers," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 3, pp. 546–558, 2021.

[16] 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*, vol. 25, no. 4, pp. 1204–1216, 2017.

[17] D. J. N. Limebeer and G. Perantoni, "Optimal control of a formula one car on a three-dimensional track—part 2: Optimal control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 5, 2015.

[18] R. Verschueren, M. Zanon, R. Quirynen, and M. Diehl, "Time-optimal race car driving using an online exact hessian based nonlinear MPC algorithm," in *2016 European Control Conference (ECC)*. IEEE, 2016.

[19] J. L. Vázquez, M. Brühlmeier, A. Liniger, A. Rupenyan, and J. Lygeros, "Optimization-based hierarchical motion planning for autonomous racing," *arXiv.org*, 2020.

[20] T. Novi, A. Liniger, R. Capitani, and C. Annicchiarico, "Real-time control for at-limit handling driving on a predefined path," *Vehicle System Dynamics*, vol. 58, no. 7, pp. 1007–1036, 2019.

[21] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[22] A. Liniger and J. Lygeros, "Real-time control for autonomous racing based on viability theory," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 464–478, 2019.

[23] A. Liniger and L. V. Gool, "Safe motion planning for autonomous driving using an adversarial road model," in *Robotics: Science and Systems XVI*. Robotics: Science and Systems Foundation, 2020.

[24] L. Chisci, J. Rossiter, and G. Zappa, "Systems with persistent disturbances: Predictive control with restricted constraints," *Automatica*, vol. 37, pp. 1019–1028, 2001.

[25] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[26] J. Kohler, R. Soloperto, M. A. Muller, and F. Allgower, "A computationally efficient robust model predictive control framework for uncertain nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 794–801, 2021.

[27] T. Brudigam, M. Olbrich, D. Wollherr, and M. Leibold, "Stochastic model predictive control with a safety guarantee for automated driving," *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2021.

[28] J. Suh, H. Chae, and K. Yi, "Stochastic model-predictive control for lane change decision of automated driving vehicles," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 4771–4782, 2018.

[29] G. Cesari, G. Schildbach, A. Carvalho, and F. Borrelli, "Scenario model predictive control for lane change assistance and autonomous driving on highways," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 23–35, 2017.

[30] A. Liniger, X. Zhang, P. Aeschbach, A. Georghiou, and J. Lygeros, "Racing miniature cars: Enhancing performance using stochastic MPC and disturbance feedback," in *2017 American Control Conference (ACC)*. IEEE, 2017.

[31] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.

[32] A. Wischnewski, M. Euler, S. Gümüs, and B. Lohmann, "Tube model predictive control for an autonomous race car," *Vehicle System Dynamics*, pp. 1–23, 2021.

[33] Z. Chai, T. Nie, and J. Becker, *Autonomous Driving Changes the Future*. Springer Singapore, 2021.

[34] L. Svensson, M. Bujarbaruah, N. Kapania, and M. Törngren, "Adaptive trajectory planning and optimization at limits of handling," 2019.

[35] T. Stahl, A. Wischnewski, J. Betz, and M. Lienkamp, "Multilayer graph-based trajectory planning for race vehicles in dynamic scenarios," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019.

[36] T. Koller, F. Berkenkamp, M. Turchetta, J. Boedecker, and A. Krause, "Learning-based model predictive control for safe exploration and reinforcement learning," *arXiv.org*, 2019.

[37] A. Kurzhanski, "Ellipsoidal calculus for estimation and feedback control," in *Systems and control in the twenty-first century*, ser. Progress in Systems and Control Theory, C. I. Byrnes, Ed. Boston: Birkhäuser, 1997, vol. 277, pp. 229–243.

[38] A. Wischnewski, T. Stahl, J. Betz, and B. Lohmann, "Vehicle dynamics state estimation and localization for high performance race cars," *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 154–161, 2019.

[39] T. Herrmann, A. Wischnewski, L. Hermansdorfer, J. Betz, and M. Lienkamp, "Real-time adaptive velocity optimization for autonomous electric cars at the limits of handling," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 665–677, 2021.

[40] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "Osqp: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, p. 637–672, 2020.

[41] TUM - Institute of Automotive Technology, "Github repository hosting the used simulation environment," 2019. [Online]. Available: https://github.com/TUMFTM

[42] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados: a modular open-source framework for fast embedded optimal control," *arXiv.org*.

[43] "Final of the two vehicle passing competition of the Autonomous Challenge @ CES." [Online]. Available: https://www.youtube.com/watch?v=df9f4Qfa0uU

[44] A. Wischnewski, J. Betz, and B. Lohmann, "Real-time learning of non-gaussian uncertainty models for autonomous racing," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020.

Alexander Wischnewski was awarded a B.Eng. in Mechatronics by the DHBW Stuttgart in 2015, and an M.Sc. in Electrical Engineering and Information Technology by the University of Duisburg-Essen, Germany, in 2017. He is currently pursuing his doctoral studies at the Chair of Automatic Control at Technical University of Munich. His research interests lie at the intersection of control engineering and machine learning, with a strong focus on autonomous driving. In addition, he was the team leader of the TUM Autonomous Motorsport team.

Thomas Herrmann was awarded a B.Sc. and an M.Sc. in Mechanical Engineering by the Technical University of Munich (TUM), Germany, in 2016 and 2018, respectively. He is currently pursuing his doctoral studies at the Institute of Automotive Technology at TUM where he is working as a Research Associate. His current research interests include optimal control in the field of trajectory planning for autonomous vehicles and the efficient incorporation of the electric powertrain behavior within these optimization problems.

Frederik Werner was awarded a B.Eng. in Industrial Engineering by the DHBW Stuttgart in 2017, and an M.Sc. in Automotive Engineering by the University of Applied Sciences of Munich, Germany, in 2021. He is currently pursuing his doctoral studies at the Institute of Automotive Technology at the School of Engineering & Design at Technical University of Munich (TUM), Germany. His research interests include vehicle dynamics and control, interaction and integration of autonomous software stack modules, and evaluation of overall vehicle performance.

Boris Lohmann received the Dipl.-Ing. degree in electrical engineering and the Ph.D. (Dr.-Ing.) degree in electrical engineering from the University of Karlsruhe, Karlsruhe, Germany, in 1987 and 1991, respectively. He is a Full Professor and the Head of the Chair of Automatic Control at the School of Engineering & Design, Technical University of Munich, Germany. His research interests include linear and nonlinear control systems design, modeling and model reduction, autonomous driving as well as applications in mechatronics and automotive.

7.4 A Model-Free Algorithm to Safely Approach the Handling Limit of an Autonomous Racecar

Contributions: Alexander Wischnewski, the author of this dissertation, initiated the idea for the paper and is responsible for the presented design and implementation. Johannes Betz and Boris Lohmann contributed equally to the conception of the overall vehicle system and the research project.

A Model-Free Algorithm to Safely Approach the Handling Limit of an Autonomous Racecar

1st Alexander Wischnewski
Chair of Automatic Control
Technical University of Munich
Munich, Germany
alexander.wischnewski@tum.de

2nd Johannes Betz
Chair of Automotive Technology
Technical University of Munich
Munich, Germany
betz@ftm.mw.tum.de

3rd Boris Lohmann
Chair of Automatic Control
Technical University of Munich
Munich, Germany
lohmann@tum.de

***Abstract*—One of the key aspects in racing is the ability of the driver to find the handling limits of the vehicle to minimize the resulting lap time. Many approaches for raceline optimization assume the tire-road friction coefficient to be known. However, this neglects the fact that the ability of the system to realize such a race trajectory depends on complex interdependencies between the online trajectory planner, the control systems and the non-modelled uncertainties. In general, a high quality control system can approach the physical limit more reliable, as it applies less corrective actions. We present a model-free learning method to find the minimum achievable lap-time for a given controller using online adaption of a scale factor for the maximum longitudinal and lateral accelerations in the online trajectory planner. In contrast to existing concepts, our approach can be applied as an extension to already available planning and control algorithms instead of replacing them. We demonstrate reliable and safe operation for different vehicle setups in simulation and demonstrate that the algorithm works successfully on a full-size racecar.**

***Index Terms*—Autonomous Racing, Learning Control, Model-Free**

I. INTRODUCTION

Autonomous driving has received great interest recently in both research and public discussions. A widely acknowledged approach is the separation of the task into perception, planning and control [1]. Recently, planning and control at the handling limits is discussed by several authors [2, 3, 4, 5]. While the split between planning and control is beneficial in terms of the development and setup process, it poses challenges when the lap time shall be minimized and the vehicle is driven at the handling limits. We identified three main difficulties while operating such a system:

1) The trajectory planner does not consider the additional lateral and longitudinal accelerations applied by the feedback-controller.
2) Deviations in the velocity tracking influence the required lateral acceleration quadratically. Therefore, its control performance has a severe impact on path tracking quality.
3) The trajectory planner applies a certain safety distance to obstacles to account for tracking errors. It is not clear a-priori how to choose this, as the tracking accuracy might decrease significantly at the handling limits.

To solve those challenges, different concepts have been presented in the literature recently: Laurense et al. [4] described a control strategy utilizing the longitudinal velocity which aims at tracking a desired front axle side slip angle calculated from a friction estimate. Due to the strong correlation of side slip with tire utilization, they could reduce the lateral error significantly in comparison to a classic path tracking controller. A completely different approach is proposed in [6], using stochastic model-predictive-control (SMPC) to account for the system uncertainties in a combined planning and tracking control setup. The work was extended in [7] using a data-driven model identification method. The approach uses constraint-tightening to guarantee that the safety constraints hold in the presence of uncertainty observed according to the identified system model. With increasing model accuracy, the driving becomes less conservative and more precise. A similar approach was combined with a Safe-MPC algorithm that can guarantee the recursive feasibility of the planning and control problem in [8]. Again, the vehicle becomes faster as more data is gathered.

In contrast to the presented references, our approach is capable of minimizing the lap time in the presence of disturbances and uncertainties for a given control and planning system with only minor modifications. Furthermore, it is not necessary to specify or identify a vehicle dynamics model as the algorithm works purely on the observations of safety constraints. The mathematical background on the algorithm applied in this paper is based on the work presented in [9]. It utilizes a Bayesian Optimization strategy with Gaussian Processes for learning the cost-function and safety constraints. Our work focusses on the choice of optimization variables, safety constraints and derivation of suitable hyperparameters for the autonomous racing application. Furthermore we present applications of the algorithm in simulation and real-world testing scenarios.

The mathematical background on Bayesian Optimization and the applied algorithm is presented in Section II. The application of the algorithm to the autonomous racing task and its results are discussed in Section III and IV. Section V reviews the achievements and presents further research directions.

Research was supported by the basic research fund of TU Munich.

II. METHODOLOGY

A. Gaussian Processes

Gaussian Processes are a machine learning technique for non-parametric function approximation. It allows one to model a non-linear function $y = f(x)\colon \mathbb{R}^d \mapsto \mathbb{R}$ using a finite set of n observations $\{(x_1, y_1), ...(x_n, y_n)\}$. The following presentation of the mathematical background is based on [10].

The Gaussian Process (GP) itself is modelled as the joint probability distribution over a finite set of Gaussian random variables. The predictions y_* can be drawn by conditioning the prediction point x_* on the observation set. The observations are stochastically related to each other based on a kernel function $k(x_i, x_j)$ which yields the covariance between x_i and x_j. This allows one to specify the joint normal probability distribution of measurement samples and predictions as

$$\begin{bmatrix} Y \\ y_* \end{bmatrix} \sim \mathcal{N}\left(m(x), \begin{bmatrix} K(X,X)+\sigma_{\mathrm{M}}^2 I & K(X,x_*) \\ K(x_*,X) & K(x_*,x_*) \end{bmatrix}\right), \quad (1)$$

where Y is the vector with measurements $\begin{bmatrix} y_0 & y_1 & \dots & y_N \end{bmatrix}^{\mathrm{T}}$, $m(x)$ the prior mean function, $K(X, X)$ the kernel matrix evaluating the kernel function $k(x_i, x_j)$ for every combination of the observation points and σ_{M}^2 the measurement variance. Note, that the input domain x might be vector valued, but the covariance function $k(\cdot)$ returns a scalar. The posterior prediction mean can be written as:

$$\begin{aligned} \mu(x_*) =& m(X) \\ & + K(x_*,X)\left(K(X,X)+\sigma_{\mathrm{M}} I\right)^{-1}(Y - m(X)) \end{aligned} \quad (2)$$

and covariance

$$\begin{aligned} \sigma(x_*) =& K(x_*,x_*) \\ & - K(x_*,X)\left(K(X,X)+\sigma_{\mathrm{M}} I\right)^{-1} K(X,x_*) \end{aligned} \quad (3)$$

based on the observations X and Y. In the following, we will denote the mean of a GP which approximates a function $f(x)$ with $\mu_{\mathrm{f}}(x)$ and its variance with $\sigma_{\mathrm{f}}(x)$.

One of the main reasons for the great interest in GPs for optimization derives from the fact that they specify the uncertainty of their predictions. This can be used to control the exploration strategy during the optimization process.

B. Safe Bayesian Optimization

Our optimization approach utilizes the SafeOpt algorithm [9]. In the following, we will only sketch the key steps and discuss their implications with respect to the optimization process and practical application. The aim of the optimization is to solve

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \le b_i,\ i = 1, \dots, m \\ & x \in \mathcal{X}, \end{aligned} \quad (4)$$

where $f(x)$ is the scalar-valued cost function, $g_i(x)$ are the safety constraints and $\mathcal{X}$ is the set of admissible parameters.

In contrast to classic optimization approaches, the cost and constraint functions are unknown before the system starts its operation. A GP is used to model them and generate samples for each function from the measurement data. Using the confidence interval β, the upper (UCB)

$$g_i^+(x) = \mu_{g,i}(x) + \beta\sigma_{g,i}(x) \quad (5)$$

and lower confidence bounds (LCB)

$$g_i^-(x) = \mu_{g,i}(x) - \beta\sigma_{g,i}(x) \quad (6)$$

can be defined. They are used to derived three important sets for the optimization. First, the set of points x which are likely to be safe after the n-th iteration, can be defined using the UCB as

$$\mathcal{S}_n = \left\{x \in \mathcal{X} | g_i^+(x) \le b_i, i = 1, \dots, m\right\}. \quad (7)$$

Within this set, the set of potential minimizers can be formulated as

$$\mathcal{M}_n = \left\{x \in \mathcal{S}_n | f^-(x) \le \min_{x' \in \mathcal{S}_n} f^+(x')\right\}. \quad (8)$$

It specifies the points that have a chance to further minimize the current, conservative best value of the cost function. Finally, the set of potential expanders is defined as the set of points that have the chance to enlarge the safe set. This is formalized as follows: Let

$$h(x_{n+1}) = |\left\{x' \in \mathcal{X} \setminus \mathcal{S}_n | x' \in \mathcal{S}_{n+1}\right\}| \quad (9)$$

be the number of points that are not in the safe set for the available observations but that become safe after adding an optimistic estimate for the constraints $y_{n+1} = g_i^-(x_{n+1})$ to the GPs. The set of potential expanders can now be written as

$$\mathcal{G}_n = \{x \in \mathcal{S}_n | h(x) > 0\}. \quad (10)$$

It remains to specify an acquisition function for how to determine the next query point for a current set of measurements. Berkenkamp et al. propose to use

$$x_{n+1} = \underset{x \in \mathcal{M}_n \cup \mathcal{G}_n}{\operatorname{argmax}} \ \sigma_{\mathrm{f}}(x), \quad (11)$$

as the algorithm can be shown to converge safely to the global safe optimal value for this choice [9].

C. Trajectory Planning and Control

The trajectory planning for the racecar is split into an online and an offline part [11]. The offline part uses a sophisticated track and vehicle dynamics model to generate the time-optimal trajectory based on an optimal-control formulation. The online phase is divided into a local path generation and a velocity profile planning problem. This allows to incorporate dynamic objects and readjustments of the velocity profile according to changing acceleration limits.

The control system consists of independent lateral path and velocity tracking controllers. The latter is a P-controller combined with disturbance estimation and a feed-forward term. A gain-scheduled PD-controller accompanied with a feed-forward term is used for path tracking. More information on the controller and overall software structure can be found in [11, 12].

III. ALGORITHM

A. Choice of Optimization Variables

The overall aim in racing is to minimize one's lap time without putting the vehicle into unstable driving situations. To achieve this, we choose to apply a scale factor θ to the acceleration limits calculated from the friction settings. This can be interpreted as a safety margin with respect to the available friction level. The acceleration limits are leveraged by the trajectory planner to adjust the velocity profile while driving. It was decided not to readjust the raceline itself in order to obtain reproducible results with respect to the safety margins. One of the key advantages of this variable choice is, that the optimization problem can be formulated to maximize the scale factor instead of minimizing the lap-time. It is known from the racing literature, that both targets are equivalent [13]. This increases the applicability of the algorithm for autonomous racing, as a data driven minimization of lap time poses difficulties due to external disturbances such as other cars or overtaking scenarios. Those would prevent the algorithm from establishing a clear relation between the optimization variable and the target variable. The problem itself can be written as:

$$\begin{aligned} \min_{\theta} \quad & -\theta \\ \text{subject to} \quad & g_i(\theta) \leq b_i,\ i = 1, \dots, m \\ & \theta \in \mathcal{O}, \end{aligned} \tag{12}$$

with the additional safety constraints g_i. The overall software architecture of the algorithm is depicted in Fig. 1.

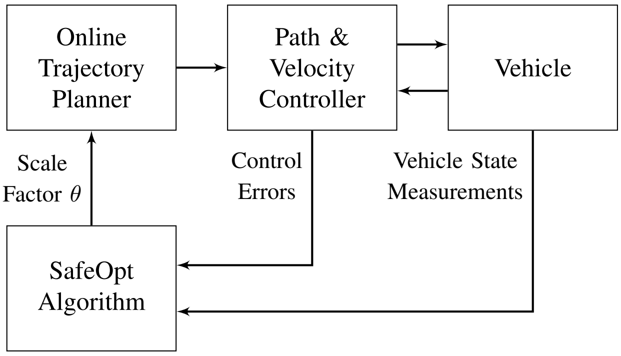

Online Trajectory Planner
Path & Velocity Controller
Vehicle
Scale Factor θ
Control Errors
Vehicle State Measurements
SafeOpt Algorithm

Fig. 1. Safe learning system architecture to minimize the resulting lap-time. The optimization algorithm is driven by the control errors and vehicle sensor data. It can influence the trajectory planner via a scale factor θ for the maximum longitudinal and lateral accelerations.

B. Choice of Constraints

The applied safe optimization strategy interprets safety as a set of constraints. To be safe, it has to guarantee that the statistical assumptions made during the prediction of the GPs are conservative, in the sense that at no time a point is predicted to be safe that is not safe with respect to the constraints. We will discuss how to achieve this by a proper selection of kernel parameters in the upcoming section. Furthermore, it is required that all parameters within the actual safe set lead to a valid execution of the task by the agent. In the case of autonomous racing, this means that all parameters that are predicted to be safe must lead to a completed lap without incidents. This translates to a strict stability requirement, as the vehicle would spin off the track otherwise. It is difficult to guarantee this analytically, since the dynamics model is highly nonlinear and uncertain in the racing case. While some authors apply system identification techniques to overcome this issue [7, 8], the aim of this paper is to set a baseline for model-free approaches. Therefore, we apply two different heuristic measures for vehicle stability from vehicle dynamics science [13]:

- The difference between the front and rear axle side-slip angle $\alpha_\mathrm{F} - \alpha_\mathrm{R}$. It relates the remaining potential of the front axle to the potential of the rear axle, being a key indicator for the amount of friction utilization. As the car has a tendency to understeer at the limits of friction, this value becomes large.
- The maximum wheel slip λ at one of the four wheels (denoted by the indices FL, FR, RL and RR). It indicates the tendency to lock or spin the wheel and that the vehicle operates near the friction limit.

It is usually difficult to judge how close the vehicle is to the limit or instability only based on data. While these criteria provide good indicators, they might fail to give a reliable estimate. This significantly depends on the vehicle dynamics design. Passenger cars are designed to have a broad transition region until the tire nonlinearity becomes relevant. In contrast, racecars are usually setup for maximum lateral acceleration instead of providing a cautious transition from stable to unstable regions of vehicle dynamics. We therefore have to rely on an underlying vehicle stability control system to handle these cases and prevent that the vehicle spins of the raceline at the cost of increased tracking error in these situations. In addition, we have to enforce a performance constraint on the absolute lateral tracking error e. It must stay within the safety margin specified by the planning algorithm, otherwise the vehicle will crash into the track boundaries.

All safety constraints must be fulfilled during the complete lap. The set of measurement points that belong to the lap with number i is denoted via $\mathcal{L}_i$. To decrease noise sensitivity, a moving average filter with a time window of $50\,\mathrm{ms}$ is applied to the sample time series before calculating the corresponding safety value. The time constant must be kept as small as possible, since too aggressive filtering could lead to violations of the safety specifications. The constraints can now be formalized:

$$g_1(\theta_i) = \max_{k \in \mathcal{L}_i} \quad |\alpha_\mathrm{F}(k) - \alpha_\mathrm{R}(k)| \tag{13a}$$

$$g_2(\theta_i) = \max_{k \in \mathcal{L}_i} \quad \max_{j \in \{\mathrm{FL,FR,RL,RR}\}} \quad |\lambda_j(k)| \tag{13b}$$

$$g_3(\theta_i) = \max_{k \in \mathcal{L}_i} \quad |e(k)| \tag{13c}$$

While the safety threshold for the last constraint is given by the specification of the trajectory planner, suitable thresholds for the heuristic stability criteria will be derived below.

C. Choice of Priors, Kernel Functions and Hyperparameters

The failure probability of the algorithm depends essentially on the accuracy of the statistical representation of the GP functions [14]. There are three main influence factors for this: The prior mean function $m(x)$ of the GP, the kernel $k(x_i, x_j)$ and the hyperparameters associated with the kernel function. While the algorithm provides a theoretical safety guarantee, this requires the knowledge of an upper bound of the norm of the final function in the Reproducing Kernel Hilbert Space (RKHS) in advance. This can be interpreted as a measure of function complexity [10]. As this is usually unknown in a practical setting, choosing the hyperparameters in a conservative manner is advised [14].

Following the arguments in [14], we apply a Matèrn Kernel with $\nu = 3/2$ so that the predictions are likely to have continuous first derivatives [10]. In contrast to the squared exponential kernel, it is less smooth, which is beneficial for the approximation of the uncertain safety constraints. This can be interpreted as a conservative choice. The variances are chosen to reflect the actual uncertainty about the outcome of the experiments based on previous test runs and are scaled to the physical units of the respective measurement variable. Finally, the kernel length scale parameter is chosen so that the GPs can generalize for a range of approximately 0.02-0.04. This resembles cautious steps near the friction limit based on past experience.

The priors' main effect is that they speed up the exploration process as they encode domain specific knowledge about the safe set and the cost function. At the same time, this can cause risk in case of misspecification. We choose to be conservative and use exponential functions that tend to have large values for an increase of the friction scale factor. In general, we know from manual driving experiments that realistic maximum scale factors range around 1.0. This leads to the conclusion, that factors below 1.0 are not safety critical. We therefore design the exponentials to cross the safety threshold at that value. At the same time, we know that scale factors above 1.2 are not realistic. We encode this in the priors so that they have roughly three times the value of the safety threshold at 1.2. The resulting priors are depicted in Table I. Note, that the cost function holds a different prior. This is due to the unusual construction of the optimization problem. In fact, it is not required to model this function as a GP at all, however, this is done to formulate the problem easily in the software framework at hand.

TABLE I
GP SETUP

Function	Prior	Variance	Length Scale	Unit
Cost	$-\theta$	1e−4	0.15	
Understeer	$b_1 e^{5.5(\theta-1)}$	1e−5	0.04	radians
Wheelslips	$b_2 e^{5.5(\theta-1)}$	1	0.04	percent
Control Error	$b_3 e^{5.5(\theta-1)}$	1e−2	0.04	meter

D. Choice of measurement uncertainty and safety thresholds

The standard GP framework allows one to model measurement uncertainty through the introduction of an additive measurement covariance matrix $\sigma_{\mathrm{M}}^2 I$ to the upper left block of the covariance matrix in (1). In this setting, the variance of the posterior distribution will become small once the number of samples in a specific region of the input domain becomes large. Eventually, it will converge to zero if the number of samples goes to infinity.

This is problematic for the application in a racing scenario. The controllers' main purpose is to mitigate external disturbances and uncertainties. While the latter will be comparable from lap to lap, the former can vary significantly. This could be caused e.g. by wind or dirt on the track surface. We will refer to this phenomenon as the repetition uncertainty u_{R} with variance σ_{R}^2, specified similar to the measurement uncertainty in the form of a Gaussian random variable. The GP framework would allow one to account for this using a white noise kernel, which adds uncertainty to the posterior that cannot be removed by conditioning it on the measurement data. However, this leads to difficulties in the application of the SafeOpt algorithm. They are caused by the fact that the potential minimizer set is calculated based on an optimistic prediction. If the uncertainty about the process does not converge to zero, this set will fail to shrink to size one. This results in a random choice of parameters within this set as also the acquisition function will converge to the same values for every sample.

Instead, we show that its possible to handle the repetition uncertainty by a conservative choice of the safety thresholds b_i. The actual constraint can be written as:

$$g_i(x) + u_{\mathrm{R}} \leq \bar{b}_i. \tag{14}$$

This shall hold with a probability of at least $1 - \delta$, where δ is the tolerated failure probability. Assuming that the SafeOpt algorithm can guarantee that $g_i(x) \leq b_i$ holds with the same probability, we can guarantee that the above holds if we set $b_i = \bar{b}_i - p\sigma_{\mathrm{R}}$. The safety factor p is obtained from the Gaussian normal distribution such that it covers $1 - \delta$ of the probability distribution.

Proof: The combined variance σ_{F}^2 of $g_i(x) + u_{\mathrm{R}}$ is equal to $\sigma_{g,i}^2(x) + \sigma_{\mathrm{R}}^2$, since both are Gaussian and uncorrelated. It follows that $\Pr\left(g_i(x) + u_{\mathrm{R}} \leq \bar{b}_i\right) \geq 1 - \delta$ if $\mathbb{E}\left[g_i(x) + u_{\mathrm{R}}\right] + p\sigma_{\mathrm{F}} \leq \bar{b}_i$. Since $\sigma_{\mathrm{F}} = \sqrt{\sigma_{g,i}^2(x) + \sigma_{\mathrm{R}}^2} \leq \sigma_{g,i}(x) + \sigma_{\mathrm{R}}$ for $\sigma_{\mathrm{R}}, \sigma_{g,i}(x) > 0$ and $\mathbb{E}\left[u_{\mathrm{R}}\right] = 0$, we can approximate the previous inequality conservatively as $\mu_{g,i}(x) + p\left(\sigma_{g,i}(x) + \sigma_{\mathrm{R}}\right) \leq \bar{b}_i$. After algebraic modifications, we have $\mu_{g,i}(x) + p\sigma_{g,i}(x) \leq \bar{b}_i - p\sigma_{\mathrm{R}}$. Setting $b_i = \bar{b}_i - p\sigma_{\mathrm{R}}$, we have shown the required result. ■

Note that the approximation approaches equality if the GP variance $\sigma_{g,i}^2(x)$ becomes small. This means the approach is conservative for the beginning of the experiment, however, it eventually converges to the exact result. The final thresholds b_i are set based on simulation and track data to $b_1 = 0.06$, $b_2 = 10$ and $b_3 = 0.7$.

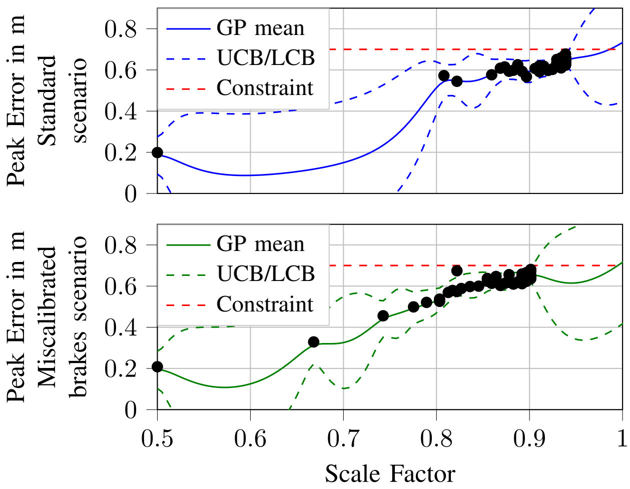


Fig. 2. The plots show the learned control error GP for the standard setup (top) and a miscalibrated brake actuator (bottom) using mean values and three sigma confidence intervals. The dots resemble measurement samples for a single lap. As the lateral error is the limiting constraint in this case, the other constraints are not visualized.

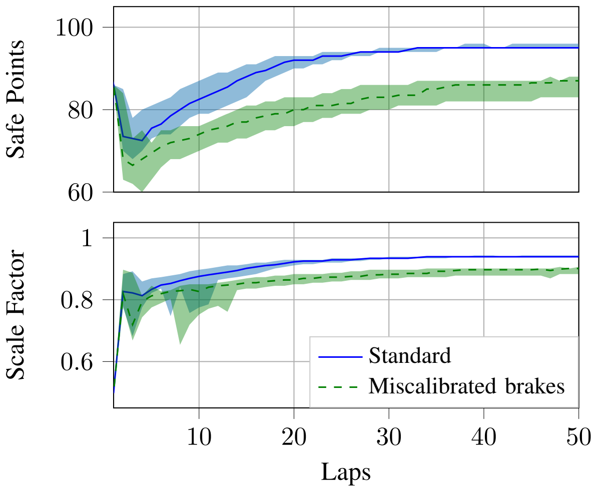


Fig. 3. The algorithm has been simulated ten times for both scenarios to analyse the variance between different runs. The uncertainty is introduced to the non-deterministic timing of many of the trajectory planner and the sensor models. The plots show the median value as bold line for each scenario. The shaded area reflects the maximum and minimum value via all iterations.

The actual choice of the measurement variance σ_{M}^2 for each GP remains to be discussed. In contrast to the repetition uncertainty σ_{R}^2, it reflects the uncertainty within the measurement, which can be reduced by consecutive measurements for the same sample points. Its proportion to the prior variance determines how strong the samples are weighted with respect to the priors. The variances are set to $\sigma_{\mathrm{M,g,1}}^2 = 1\mathrm{e}{-6}$, $\sigma_{\mathrm{M,g,2}}^2 = 4\mathrm{e}{-2}$ and $\sigma_{\mathrm{M,g,3}}^2 = 1\mathrm{e}{-3}$, which resembles a strong emphasis on the measurements.

IV. RESULTS

The algorithm is first applied to a detailed vehicle dynamics simulation. It is equipped with sensor and actuator models and a nonlinear single-track model with a Pacejka Tire Model. Since the trajectory planner, the SafeOpt algorithm and the vehicle physics run in separate processes, the outcome of the simulation is not deterministic. However, this reflects the behaviour of the car, as we cannot guarantee the timing of the trajectory planner as it is not implemented as a hard real-time application.

We use a standard simulation parametrization as a baseline and discuss the performance of the algorithm for a miscalibrated brake controller as a benchmark. It is set to a 30 % error in the brake requests and will therefore lead to a degraded velocity tracking. Fig. 2 depicts the learned GP for the peak control error function, which limits the exploration in both scenarios.

As pointed out before, the behaviour of the algorithm can not be considered deterministic, due to timing differences and different initial conditions for the sensor noise. The results of 10 experiments with 50 laps for each scenario are depicted in Fig. 3. The median is a scale factor of 0.90 for the misspecified brake actuator setting, which is below the final value of 0.94 in the standard setting. Two things should be noticed: First, the algorithm converges very slowly once it is near the friction limit, which is due to its conservative parametrization. Second, outliers in the constraints tend to force the algorithm to go back to a more conservative value for a few iterations even if it is near its final result. Again, this reflects the conservative nature of the algorithm, being focussed more on safety than on pure performance maximization.

Finally, the algorithm is applied to a full-size racecar called *DevBot 2.0*. It has a two-wheel electric drivetrain, steer- and brake-by-wire systems and is used within the Roborace Championship for autonomous vehicles. Details on the vehicle setup can be found in [15, 11]. The trials took place at the Monteblanco Circuit, Spain. The algorithm converged to an acceleration limit scale factor of 0.83, which was slightly below the factor found during manual setup of the autonomous driving system, which was 0.9. This deviation can be attributed to the conservative wheelspeed constraint value, where 10 % seems too low for the actual vehicle setup since there has not been any sign of instability noted by the safety driver or within the measurement data. The limiting constraint GP is depicted in Fig. 4.

V. CONCLUSIONS AND OUTLOOK

We have presented an algorithm that is capable of mitigating the limitations of the vehicle in the sense, that it scales the allowed accelerations in trajectory planning so that the vehicle does not violate the safety constraints. It has been proven to come reasonably close to the result achieved by manual tuning of the scale factor. It does not depend on detail knowledge about the underlying tracking controller. This facilitates easy integration in already available software stacks. Furthermore, it has been shown to be robust with respect to moderate constraint prior misspecification, which is an important feature for algorithms applied within real-world scenarios.

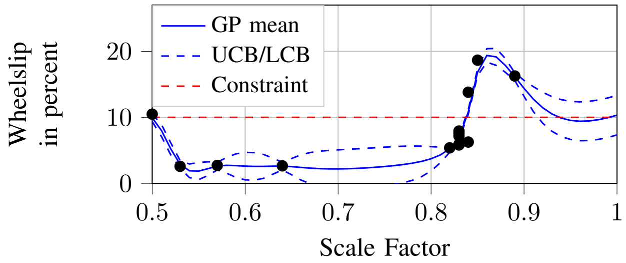


Fig. 4. At the Monteblanco Racetrack the vehicle was stopped from further minimizing the lap time by a conservative setting of the wheelslip constraint. Only the limiting constraint is depicted for a run of 14 laps. Non-safe samples were drawn due to a mismatch of the prior function and the real-world behaviour of the system. The high wheelslip sample for scale factor 0.5 is related to starting to drive from standstill. Since the prior specifies other values to be safe, this does not stop the algorithm from exploring.

One drawback of the algorithm is that the vehicle is not capable of adjusting its accelerations limitations to different circumstances on different areas on the track. Furthermore, it could leverage this information to learn faster than on a per lap basis. The difficulty with this lies in the fact that the control error depends significantly on the initial conditions present at the entry of a subsection of the lap. While this is already a problem for the algorithm running on a per lap basis, lap switching usually takes place at the start-straight where the control errors are rather low and therefore neglectable. Another difficulty lies in the fact that it is conservative with respect to outliers that exceed the constraint values. Since the algorithm does not actively retry them, they might block the exploration in case the GP length scales are set rather small, which is usually desirable to ensure safe exploration at the limits.

Future work will be dedicated to the utilization of information on a finer grid rather than a per lap basis and will leverage additional model information in the learning process. The latter promises to result in more robust guarantees in terms of control system stability and could decrease the dependence upon heuristic vehicle stability criteria.

CONTRIBUTIONS AND ACKNOWLEDGMENTS

Alexander Wischnewski initiated the idea for the paper and is responsible for the presented design and implementation. Johannes Betz and Boris Lohmann contributed equally to the conception of the overall vehicle system and the research project.

We would like to thank Roborace for the opportunity to evaluate our algorithms on their prototype as well as for their support during the testing sessions. Furthermore, we thank Johannes Strohm, Matthias Rowold, Leon Sievers, Lukas Schäfer and Tim Stahl for the many discussions on the topic and the critical revision of the manuscript.

REFERENCES

[1] S. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. Eng, D. Rus, and M. Ang, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, 2017.

[2] M. Werling, "Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien," *KIT Scientific Publishing, Karlsruhe*, vol. 60, 2011.

[3] A. Liniger, "Path Planning and Control for Autonomous Racing," *ETH Zürich Research Collection*, 2018.

[4] V. A. Laurense, J. Y. Goh, and J. C. Gerdes, "Path-tracking for autonomous vehicles at the limit of friction," *Proceedings of the American Control Conference*, pp. 5586–5591, 2017.

[5] N. R. Kapania, "Trajectory Planning and Control for an Autonomous Race Vehicle," *Stanford Digital Repository*, 2016.

[6] A. Liniger, X. Zhang, P. Aeschbach, A. Georghiou, and J. Lygeros, "Racing miniature cars: Enhancing performance using stochastic mpc and disturbance feedback," *Proceedings of the 2017 American Control Conference*, pp. 5642–5647, 2017.

[7] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious nmpc with gaussian process dynamics for autonomous miniature race cars," *Proceedings of the 2018 European Control Conference*, pp. pp. 1341–1348, 2018.

[8] U. Rosolia, A. Carvalho, and F. Borrelli, "Autonomous Racing using Learning Model Predictive Control," *arxiv.org*, 2016.

[9] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with Gaussian processes," *Proceedings of the 2016 IEEE International Conference on Robotics and Automation*, pp. 493–496, 2016.

[10] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[11] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer, and M. Lienkamp, "A software architecture for an autonomous racecar," in *2019 IEEE 89th Vehicular Technology Conference*. IEEE, Apr. 2019.

[12] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, "Minimum curvature trajectory planning and control for an autonomous race car," *Vehicle System Dynamics*, pp. 1–31, Jun. 2019.

[13] W. F. Milliken and D. L. Milliken, *Race Car Vehicle Dynamics*. Great Britain: Society of Automotive Engineers Inc., 1996.

[14] F. Berkenkamp, A. Krause, and A. P. Schoellig, "Bayesian optimization with safety constraints: Safe and automatic parameter tuning in robotics," *arxiv.org*, 2016.

[15] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer, and M. Lienkamp, "What can we learn from autonomous level-5 motorsport?" in *Proceedings*. Springer Fachmedien Wiesbaden, Sep. 2018.

7.5 Real-Time Learning of Non-Gaussian Uncertainty Models for Autonomous Racing

Contributions: Alexander Wischnewski, the author of this dissertation, initiated the idea for the paper and is responsible for the presented analysis and the implementation of the uncertainty learning algorithm. Johannes Betz and Boris Lohmann contributed equally to the conception of the research project.

Real-Time Learning of Non-Gaussian Uncertainty Models for Autonomous Racing

Alexander Wischnewski, Johannes Betz and Boris Lohmann

Abstract— **Performance and robustness targets have been considered for controller design for decades. However, robust controllers usually suffer from performance limitations due to conservative uncertainty assumptions made a priori to system operation. The increased number of systems (e.g. autonomous vehicles) which require high-performance operation in safety-critical environments is motivating research in novel design methods. Recently, machine learning methods have emerged as a promising way to reduce conservatism, based on data gathered during system operation. We propose a combination of a recursive least squares estimator with a recursive quantile estimator to identify feature-dependent upper and lower uncertainty bounds. We give conditions under which the estimator converges to a robust invariant set, such that the resulting bounds cover a target proportion of the samples up to small error. In contrast to widely applied Gaussian process regression or Bayesian linear regression approaches, we do not imply any assumptions about the probability distribution of the samples. We demonstrate that the estimated bounds achieve the desired data coverage in contrast to state-of-the-art approaches on academic examples, as well as a motion control example for an autonomous race car. Furthermore, the approach exhibits very low computational requirements and is therefore suitable for application on embedded systems.**

I. Introduction

A. Motivation & Outline

The conflict between stability, robustness, and performance in controller design is well known and analyzed within the control community [1]. An important aspect is that of robust performance, especially in constrained, safety-critical systems such as chemical plants or autonomous vehicles. One difficulty with achieving acceptable performance while maintaining the robustness requirements is that the uncertainty model often has to be chosen conservatively due to unknown and complex operating conditions. The combination of learning methods with robust control [2], [3] is a promising way of overcoming this problem. We research such methods by application to an autonomous race car with the aim of achieving human-like performance under a wide range of environmental conditions [4], [5]. A key challenge for learning robust control based on Tube-MPC [6] is their computational complexity. To mitigate this, we present an uncertainty learning method with low computational requirements capable of handling non-Gaussian uncertainty.

The algorithm is based on a novel combination of the well-known normalized least mean squares filter [7] and a quantile estimation method inspired by [8]. Stability and performance of the algorithm are analyzed within a purely deterministic setting, using a robustness definition based on Lyapunov functions and set theory. In particular, we reformulate the above algorithms in batch form, which allows us to drop probabilistic assumptions about the data such as being Gaussian-distributed or uncorrelated over time. These are usually not satisfied when learning a possibly state-dependent uncertainty model. We demonstrate superior performance compared to state-of-the-art approaches for academic examples, as well as an autonomous race car motion controller. The algorithms recursive nature and small computational requirements enable the application in conjunction with an computationally-intense robust control scheme, such as Tube-MPC [9].

The paper is structured as follows: The remainder of this section presents related work on uncertainty model identification. Section II analyzes the proposed approach, while Section III presents a benchmark of the algorithm on academic examples and a dataset obtained from a motion controller simulation of an autonomous race car. Section IV concludes the paper and points out future research directions.

B. Related Work

Robust control design requires the uncertainties in the dynamics and external disturbances to be specified. While these were mainly determined by expert knowledge in the past, a variety of control design methods utilizing a learned additive uncertainty representation (possibly state dependent) have been recently proposed [2], [3], [6], [10]–[12]. Robust constraint satisfaction is ensured by learning upper and lower uncertainty bounds which can be used to determine reachset predictions. In contrast to the above methods, the reachset conformant synthesis algorithm proposed by [13] adjusts the uncertainty model such that the resulting reachsets are conformant to the data for a given set of training trajectories. While this allows the exploitation of the potential correlation of the uncertainties over time, it requires a large set of trajectories to form a reliable dataset. This makes it difficult to use in online methods. We will therefore focus our further presentation on algorithms of the first type.

A common approach for identifying a suitable uncertainty model is to apply Gaussian process regression (GPR) [10], [11], [14]. The interest of the control community in this method is motivated by excellent performance in applications as well as the availability of a sound theoretical background.

Research was supported by the basic research fund of TU München.

A. Wischnewski and B. Lohmann are with the Chair of Automatic Control, Department of Mechanical Engineering, Technische Universität München, München, Germany. (email: {alexander.wischnewski, lohmann}@tum.de

J. Betz is with the Chair of Automotive Technology, Department of Mechanical Engineering, Technische Universität München, München, Germany. email: betz@ftm.mw.tum.de

The key idea behind its prediction mechanism is to utilize the similarity of the prediction points to the training samples via a kernel function [15]. The fully Gaussian framework permits the derivation of a confidence interval for the posterior distribution as an estimate of the upper and lower bounds of the uncertainty. The high computational complexity has been tackled by various sparse approaches, e.g., by restricting the training to a subset of well-distributed, relevant samples [6], [16] or only utilizing samples from the neighborhood of the prediction space [17], [18]. Besides the high computational requirements, the accuracy of the posterior distribution depends on the assumption of Gaussian noise and the correct choice of hyperparameters [15]. This becomes evident if the model should also capture the aleatoric uncertainty of the model under control. A realistic model, therefore, requires online hyperparameter optimization, which further increases the computational requirements. On the downside, this also leads to difficulties in modeling the epistemic uncertainty at the beginning of the identification process and breaks the corresponding theoretical bounds used to guarantee cautious exploration [19]. Another nonparametric regression technique is Kinky-Inference [20], a concept exploiting Lipschitz properties of the identified function. Its downside is a disadvantage of all nonparametric approaches: Their predictions are inferred from the data points, which makes it challenging to implement a computationally efficient adaption mechanism for slowly time-varying uncertainties [29].

An alternative to GPR is Bayesian linear regression (BLR) [21], [22]. It is usually implemented in a recursive manner using Monte-Carlo techniques and deliver a joint estimate of the posterior mean and variance. A compelling special case of BLR using fixed basis functions and noise variance estimation is applied to the uncertainty learning problem in [12] following an approach presented in [23]. It differs from the previously discussed approaches in the fact that it adjusts to the actual variance observed in the data and therefore learns the aleatoric uncertainty in the samples. Furthermore, it can be implemented by the use of simple matrix operations and shows to outperform standard GPR in terms of posterior accuracy. However, the key assumption of this algorithm, independent, identically distributed (i.i.d.) Gaussian noise, does not hold in many applications.

The assumption of Gaussian noise could be removed by the calculation of the full posterior over the parameters of Bayesian neural networks. Since this task is computationally intractable for nearly all models, approximations based on gradient descent have been proposed [24]. Another approach capable of dealing with non-gaussian uncertainty is conditional quantile regression [25]. It generalizes the concept of a quantile to a feature-dependent function, splitting the data into given percentiles. A similar idea is the identification of an Interval Predictor Model [26]. It constructs feature-dependent upper and lower bounds for a given dataset and derives probabilistic precision guarantees based on the number of training samples. Both approaches require the solution of a computationally-intense optimization problem which makes them unsuitable for real-time control.

II. Methodology

A. *Problem Formulation*

We consider the discrete-time system

$$x(k+1) = f(x(k), u(k)) + Ed(x(k), u(k), k), \qquad (1)$$

where $x(k) \in \mathbb{R}^n$ is the state, $u(k) \in \mathbb{R}^p$ the control input and $f(x(k), u(k))$ the system dynamics. The uncertainty is modeled as an additive signal $d(x(k), u(k), k) \in \mathbb{R}$. The approach can be extended to incorporate dependencies on external signals, however, we omit this for the sake of brevity. We also drop the explicit dependency on the time k in the following and restrict our analysis to the scalar case by requiring that E has rank one. This allows suitable estimates to be generated for the uncertainty if the state and the control signal are available. If the disturbance is vector valued but its components are independent of each other, the extension to the multi-dimensional case can be done by multiple application of the algorithm presented below. The case of dependent disturbances appears to be more difficult due to the non-defined ordering for vector-valued signals and is considered future work.

We aim to identify upper and lower bounds such that

$$d^-(x, u) \leq d(x, u, k) \leq d^+(x, u) \qquad (2)$$

holds for Nq sample, where N represents the number of samples which form a batch and q the target coverage of the data. This aim is reformulated such that

$$d(x, u, k) \leq d^+(x, u) \qquad (3a)$$

$$d(x, u, k) \leq d^-(x, u) \qquad (3b)$$

hold for Nq^+ respectively Nq^- samples and q^+ and q^- are the proportions of the data to be covered. This can be seen as a deterministic equivalent to the stochastic interpretation of an upper and lower quantile. The values q^+ and q^- are usually chosen to cover a large range of the data, e.g. to be $q^+ = 0.99$ and $q^- = 0.01$. The target coverage can be calculated to be $q = q^+ - q^-$.

Given the above description, sets that are likely to include the closed-loop system state can be predicted for (1) using set-theoretic methods [27]. They can be used for guaranteeing the satisfaction of state and input constraints in robust control schemes such as Tube-MPC [9]. However, our interest lies in the real-time identification of accurate and tight uncertainty bounds, to enable non-conservative, robust control and we will therefore not discuss the construction of the reachable sets.

B. *Recursive Uncertainty Model Identification (RUMI)*

We propose to solve the above problem by combination of a batch-version of the recursive Least Mean Squares (LMS) algorithm presented in [7] with a batch version of a recursive quantile estimation (QE) algorithm inspired by [8]. This allows us to model the feature dependency of the mean value; however, the uncertainty itself is assumed to be independent of these features. The general structure of the resulting RUMI algorithm is depicted in Fig. 1. We assume throughout the

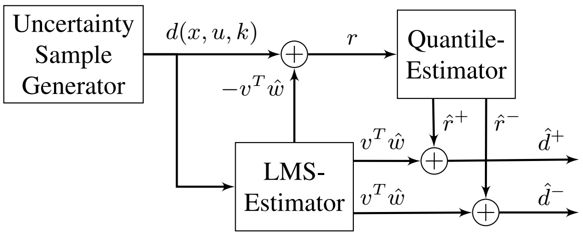


Fig. 1. Uncertainty model identification with RUMI. The explicit dependency on the time-step k is dropped in the scheme.

paper that a suitable method for measuring or estimating the additive uncertainty acting upon the system, e.g. full state knowledge and solving (1) for d, is available. The presented method is applicable independently from this choice.

The LMS algorithm delivers an L2-optimal estimate $\hat{w} \in \mathbb{R}^f$ for the weights of the linear regression model [23]

$$d(x,u,k) = v^T(k)\hat{w}(k) + r(k), \tag{4}$$

$$v^T(k) = \begin{bmatrix}\phi_1(x(k),u(k)) & \dots & \phi_f(x(k),u(k))\end{bmatrix}, \tag{5}$$

where $v(k) \in \mathbb{R}^f$ is the vector of basis functions $\phi(\cdot)$ and $r(k)$ the remaining unstructured uncertainty. The basis functions are allowed to depend on the states as well as the inputs. This decomposition is applicable to all bounded uncertainties described in (1). However, the basis functions should be chosen (e.g. by domain knowledge) such that $r(k)$ is small to reduce conservatism.

The unstructured uncertainty $r(k) = d(x,u,k) - v^T(k)\hat{w}(k)$ is used to estimate the upper and lower bounds $\hat{r}^+(k)$ and $\hat{r}^-(k)$. Combining these with the result of the LMS algorithm, the bounds can be written as

$$\hat{d}^+(v(k)) = v^T(k)\hat{w}(k) + \hat{r}^+(k) \tag{6a}$$

$$\hat{d}^-(v(k)) = v^T(k)\hat{w}(k) + \hat{r}^-(k). \tag{6b}$$

This concept allows to construct a reliable uncertainty model and to ensure robust convergence with only deterministic assumptions. The key change in contrast to the available algorithms is that we conduct the analysis in a lifted time setting utilizing batches of data for updating the estimates. This permits us to bound the effects of correlated data and noise over the time-horizon spanned by the batch, instead of imposing probabilistic assumptions. All computations can be implemented recursively in every time step and the resulting computational load is independent of the batch size.

C. Preliminaries

The analysis will be conducted in a set-theoretic setting based on Lyapunov theory. This section outlines the required definitions and theorems following the presentation in [27].

Lyapunov function outside a set (based on Definition 2.40 [27]): The continuous, positive definite function $V(x)$ is said to be a Lyapunov function outside the set $\mathcal{N} = \{x \in \mathbb{R}^n | V(x) < \nu^2\}$ if there exists a $\nu > 0$ such that for all $x \notin \mathcal{N}$ the Lyapunov difference $\Delta V(x)$ can be bounded from above by $\Delta V(x) \leq -\Phi(\|x\|)$ for some class $\mathcal{K}$-function Φ and $\mathcal{N}$ is a robust positive invariant set.

Robust convergence to a set (based on Theorem 2.42 and Definition 2.20 [27]): Assume the system admits a Lyapunov function $V(x)$ outside the set $\mathcal{N}$. Then it is uniformly ultimately bounded in the robust invariant set $\mathcal{N}$. The latter implies that for all $\delta > 0$ there exists $T(\delta) > 0$ such that for every $\|x\| \leq \delta$ it holds that $x(k) \in \mathcal{N}$ for all $k > T(\delta)$ under all uncertainties.

Robust exponential convergence (based on Theorem 2.27 and Theorem 2.46 [27]): Assume that the previous statement holds for some class $\mathcal{K}$-function $-\Phi(\|x\|) \leq -\beta V(x)$ with $V(x) = x^T x$ and $0 < \beta < 1$. Then we say that the system converges exponentially to the robust invariant set $\mathcal{N}$ and it holds that

$$\|x(k)\|^2 \leq \max\left\{(1-\beta)^k \|x(0)\|^2,\, \nu^2\right\} \quad \forall k. \tag{7}$$

D. Least Mean Squares Algorithm (Main Result I)

There are two widely known algorithms solving the linear least squares regression problem recursively. The first, Recursive Least Squares with forgetting factor (RLS), is derived from an H_2-optimal solution for a regularized least squares optimization problem for every single time-step [7]. The Least Mean Squares (LMS) algorithm is based on a stochastic-gradient method. It proved to be an H_∞-optimal solution to the parameter identification problem [7] and is computationally cheaper since it does not have to keep track of the estimation-error covariance matrix.

We will now show, that a modified version of the normalized LMS-update rule [7] for the estimate $\hat{w}(k+N) = \Delta\hat{w}(k) + \hat{w}(k)$ using batch updates every N steps

$$\Delta\hat{w}(k) = \gamma \sum_{i=k}^{k+N-1} \frac{\mu v(i)}{1 + \mu v^T(i)v(i)} \left(d(i) - v^T(i)\hat{w}(k)\right) \tag{8}$$

converges to the correct estimate. The parameter $\gamma > 0$ depicts the learning rate and the parameter $\mu > 0$ is used to mitigate ill-posedness of the problem.

To establish the required guarantees on the upper and lower uncertainty bounds, we need to bound the absolute value of the model error $|v^T(k)\tilde{w}(k)|$ with the parameter error $\tilde{w}(k) = w_\text{o}(k) - \hat{w}(k)$ robustly for all realizations $d(k)$. We define $w_\text{o}(k)$ as the weighted least-squares approximations for the model (4) in the time interval k to $k+N-1$, which can be obtained via the weighted pseudo-inverse [28] by $w_\text{o}(k) = \left(V_k M_k V_k^T\right)^{-1} V_k M_k D_k$, where each column of $V_k \in \mathbb{R}^{f\times N}$ corresponds to a sample $v(k)$ to $v(k+N-1)$, $M_k \in \mathbb{R}^{N\times N}$ is a matrix with entries obtained from the scalar factors in the sum of (8) on its diagonal and $D_k \in \mathbb{R}^N$ holds the corresponding measurement samples $d(k)$. Introducing this change of variables and rewriting (8) the batch difference for the parameter error becomes

$$\Delta\tilde{w}(k) = \Delta w_\text{o}(k) - \gamma V_k M_k (R_k + V_k^T w_\text{o}(k) - V_k^T \hat{w}(k)), \tag{9}$$

where $\Delta w_\text{o}(k)$ is the batch difference between the optimal solutions and R_k is the column vector holding the residuals with respect to the model $v^T(k)w_\text{o}$. By the definition of w_o, it follows that R_k is orthogonal to $V_k^T w_\text{o}$ and lies within the nullspace of $V_k M_k$. Therefore the influence of the residual vector R_k vanishes:

$$\Delta\tilde{w}(k) = \Delta w_\text{o}(k) - \gamma V_k M_k V_k^T \tilde{w}(k). \tag{10}$$

We will denote $V_k M_k V_k^T$ by the symmetric positive semi-definite matrix P_k and rewrite it as a sum

$$P_k = \sum_{i=k}^{k+N-1} \frac{\mu v(i) v^T(i)}{1 + \mu v^T(i) v(i)} \tag{11}$$

with minimum and maximum singular values σ_k^- and σ_k^+ respectively. By taking the maximum and the minimum over all k, it is ensured that

$$\sigma^- \|\tilde{w}\|^2 \leq w^T P_k w \leq \sigma^+ \|\tilde{w}\|^2 \quad \forall k \tag{12}$$

holds. If $\sigma^- > 0$, P_k is positive definite and the equation is similar to the persistent excitation condition in [7]. Defining the Lyapunov function $V(\tilde{w}) = \tilde{w}^T\tilde{w}$, the difference $\Delta V(\tilde{w}(k)) = V(\tilde{w}(k+N)) - V(\tilde{w}(k))$ for (10) is

$$\begin{aligned} \Delta V =& \tilde{w}^T \left(\gamma^2 P_k^2 - 2P_k\gamma\right) \tilde{w} + \Delta w_\text{o}^T \Delta w_\text{o} \\ &+ 2\Delta w_\text{o}^T \left(I - \gamma P_k\right) \tilde{w}. \end{aligned} \tag{13}$$

Rewriting P_k with the orthonormal basis Q_k we have:

$$\begin{aligned} \Delta V =& -\tilde{w}^T Q_k \left(2\gamma\Sigma_k - \gamma^2\Sigma_k^2\right) Q_k^T \tilde{w} + \Delta w_\text{o}^T \Delta w_\text{o} \\ &+ 2\Delta w_\text{o}^T Q_k \left(I - \gamma\Sigma_k\right) Q_k^T \tilde{w}. \end{aligned} \tag{14}$$

If we choose γ such that $\gamma < \frac{1}{\sigma^+}$ (a reasonable upper bound on σ^+ could be obtained e.g. from extensive simulations as it only depends on the features v) it follows from $2a - a^2$ being strictly monotone on $[0, 1]$ that the minimum singular value of $\left(2\gamma\Sigma_k - \gamma^2\Sigma_k^2\right)$ is $2\gamma\sigma_k^- - \gamma^2{\sigma_k^-}^2$. We can then apply (12) and the Cauchy-Schwarz inequality to obtain

$$\begin{aligned} \Delta V \leq& -\left(2\gamma\sigma^- - \gamma^2{\sigma^-}^2\right) \|\tilde{w}\|^2 + \|\Delta w_\text{o}\|^2 \\ &+ 2\left(1 - \gamma\sigma^-\right) \|\Delta w_\text{o}\| \|\tilde{w}\|. \end{aligned} \tag{15}$$

The first term is always negative for $0 < \gamma < \frac{1}{\sigma^+}$ which allows us to define the positive root of the polynomial $\Delta V^+(\|\tilde{w}\|)$ on the right-hand-side by $\|\tilde{w}_\text{R}\|$ and the set $\mathcal{N} = \left\{\tilde{w} \in \mathbb{R}^n | V(\tilde{w}) \leq \nu^2 = \alpha^2 \|\tilde{w}_\text{R}\|^2\right\}$ where $\alpha > 1$. It follows from comparison with $\Delta V^+(\|\tilde{w}\|)$ that $-\beta V(\tilde{w})$ bounds the Lyapunov difference from above outside $\mathcal{N}$ for

$$\beta \leq -\frac{\Delta V^+(\nu)}{\nu^2}. \tag{16}$$

It remains to show that $\mathcal{N}$ is a robust invariant set. By using the fact that all coefficients of the polynomial

$$\begin{aligned} V(\tilde{w}(k+1)) \leq& \|\tilde{w}\|^2 - \left(2\gamma\sigma^- - \gamma^2{\sigma^-}^2\right) \|\tilde{w}\|^2 \\ &+ 2\left(1 - \gamma\sigma^-\right) \|\Delta w_\text{o}\| \|\tilde{w}\| + \|\Delta w_\text{o}\|^2 \end{aligned} \tag{17}$$

are positive, the right hand side is strictly monotonic increasing for $\|\tilde{w}\| > 0$. Together with $\Delta V^+(\nu) < 0$, it follows that the right hand side is bounded from above by ν^2. This completes the set of conditions required for robust exponential convergence of the update rule (8) according to the theorem presented in section II-C.

E. Recursive Quantile Estimation (Main Result II)

In the next step, we will show how to determine the estimate $\hat{r}^+$ reflecting that a certain proportion q^+ of the residuals r are smaller than or equal to this value. The results hold equally for $\hat{r}^-$. A naive approach would be to store all data points and order them, however, this is not suitable for continuous data streams, due to limited memory and computation power. Our approach is based on a batch version of the recursive algorithm presented in [8].

We define the empiric estimator

$$\hat{\alpha}(\hat{r}^+, k) = \frac{1}{N} \sum_{i=k}^{k+N-1} \mathbb{1}\left(r(i) \leq \hat{r}^+\right) \tag{18}$$

for the proportion of the data which is smaller than or equal to the current quantile estimate and $\mathbb{1}\left(r(i) \leq \hat{r}^+\right)$ is equal to one if the condition in brackets is true, and zero otherwise. Rewriting the algorithm of [8] in batch form, we obtain the following update equation

$$\begin{aligned} \hat{r}^+(k+N) = \hat{r}^+(k) &+ \lambda q^+ \left(1 - \hat{\alpha}(\hat{r}^+(k), k)\right) \\ &- \lambda(1 - q^+)\hat{\alpha}(\hat{r}^+(k), k), \end{aligned} \tag{19}$$

where $\lambda > 0$ denotes the adaption speed. Note, that we dropped the multiplicative dependence of the update equation on the estimate $\hat{r}^+(k)$ compared to [8] to ensure that the convergence properties are independent of the final value. Reformulating $\tilde{\alpha}(\hat{r}^+, k) = \hat{\alpha}(\hat{r}^+, k) - q^+$, (19) reduces to

$$\hat{r}^+(k+N) = \hat{r}^+(k) - \lambda\tilde{\alpha}(\hat{r}^+(k), k). \tag{20}$$

Due to the data-induced variation of (18), the best result achievable is to show that the batch estimator (20) converges to (and stays within) a ball $\mathcal{S}$ in $\mathbb{R}$ centered around r_c^+ and radius $\rho > 0$. We will show these properties using the Lyapunov function $V(\tilde{r}) = \tilde{r}^2$ with the deviation $\tilde{r} = \hat{r}^+ - r_\text{c}^+$. Dropping the explicit time dependency, the Lyapunov difference $\Delta V(\tilde{r}(k)) = V(\tilde{r}(k+N)) - V(\tilde{r}(k))$ is given by

$$\Delta V = \Delta\tilde{r}\left(\Delta\tilde{r} + 2\tilde{r}\right), \tag{21}$$

where $\Delta\tilde{r} = \tilde{r}(k+N) - \tilde{r}(k)$. If we show that $\Delta V < -\eta^2$ holds outside $\mathcal{S}$ for some positive constant $\eta > 0$, the existence of a class $\mathcal{K}$-function Φ bounding ΔV from above outside of $\mathcal{S}$ follows directly. This holds under the following conditions

$$-2\tilde{r} + \eta < \Delta\tilde{r} < -\eta \quad \forall\tilde{r} > \rho \tag{22a}$$

$$\eta < \Delta\tilde{r} < -2\tilde{r} - \eta \quad \forall\tilde{r} < -\rho \tag{22b}$$

If we can further show that

$$-\tilde{r} - \rho \leq \Delta\tilde{r} \leq -\tilde{r} + \rho \quad \forall|\tilde{r}| \leq \rho \tag{23}$$

holds, the set $\mathcal{S}$ is robust positively invariant, and therefore the estimate converges robustly to $\mathcal{S}$ according to the theorem presented in section II-C. If the left-hand side holds additionally for $\tilde{r} \geq \rho$, it converges without overshooting, which leads to desirable conservative behavior. By inserting (20) into the definition of $\Delta\tilde{r}$, we get

$$\Delta\tilde{r}(k) = \Delta r_\text{c}^+(k) - \lambda\tilde{\alpha}(r_\text{c}^+(k) + \tilde{r}(k), k). \tag{24}$$

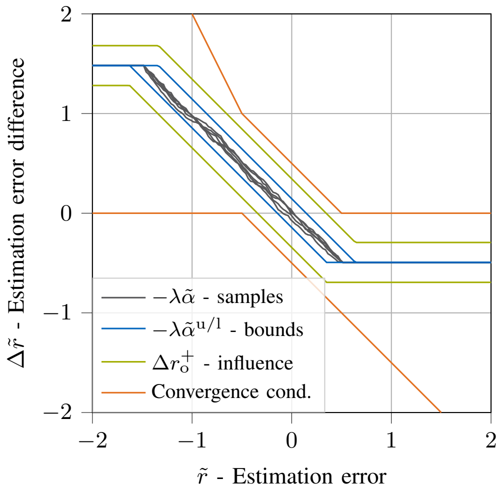


Fig. 2. Visualization of (24) and the convergence conditions (22) and (23) for a robust invariant set with size $\rho = 0.5$. The adaption speed was set to $\lambda = \frac{1}{m}$ and the movement of the invariant set is bounded by $|\Delta r_{\mathrm{c}}^{+}| < 0.2$. The constant η is neglected as it can be chosen arbitrary small.

To establish the desired properties, it is required that the variation of the empiric estimator (18) is bounded. This can be ensured if time-invariant upper and lower bounds

$$\hat{\alpha}^{\mathrm{l}}(r_{\mathrm{c}}^{+}(k) + \tilde{r}) \leq \hat{\alpha}(\hat{r}^{+}, k) \leq \hat{\alpha}^{\mathrm{u}}(r_{\mathrm{c}}^{+}(k) + \tilde{r}). \tag{25}$$

exist and are solely shifted by the (possibly time-dependent) center of the invariant set r_{c}^{+}. This can be interpreted such that the shape-variation of $\hat{\alpha}(\hat{r}^{+}, k)$ is limited over time. Analyzing the convergence requirements for the general case is a difficult task, due to the data-dependent nature of the estimator $\tilde{\alpha}(\hat{r}^{+}(k), k)$. We will therefore restrict our analysis to parametrizations of upper and lower bounds in the form

$$\hat{\alpha}^{\mathrm{u/l}}(\hat{r}^{+}(k)) = \mathrm{sat}\left(m\left(\hat{r}^{+}(k) - r_{\mathrm{s}}^{\mathrm{u/l}}\right)\right), \tag{26}$$

where sat is the saturation function limiting between 0 and 1, m the linear gain and $r_{\mathrm{s}}^{\mathrm{u/l}}$ the corresponding shift of bounds. It is always possible to find such a parametrization, although less conservative results might be obtained with other choices. A natural choice for the center $\hat{r}_{\mathrm{c}}^{+}$ of the set $\mathcal{S}$ is the center between the upper and lower bounds at the target value q^{+}. Fig. 2 visualizes (24) as well as the convergence conditions (22) and (23). It shows samples of $-\lambda\tilde{\alpha}$ for different batches drawn from a uniform random probability distribution. The size ρ of the robust invariant set is bounded from below by the variation of $\tilde{\alpha}(\tilde{r})$ and the time-variation of the invariant set center $\Delta r_{\mathrm{c}}^{+}$.

F. *Interconnection of LMS and QE (Main Result III)*

The previous section demonstrated that the quantile estimation algorithm converges towards a robust invariant set around r_{c}^{+} for an arbitrary input signal $r(k)$ as long as the upper and lower bounds as in (25) exist. Taking a closer look onto the signal $r(k)$, it can be decomposed into the residual $r_{\mathrm{o}}(k)$ with respect to the local L2 optimal solution w_{o} and the model error as in (9):

$$r(k) = r_{\mathrm{o}}(k) + v^{T}(k)\tilde{w}(k). \tag{27}$$

The latter term in this equation can be interpreted as an additional uncertainty with respect to the quantile identification of $r_{\mathrm{o}}(k)$. However, it converges to the robust invariant set $\mathcal{N}$ in finite time according to the result established in section II-D. In this set, it is possible to bound the influence by

$$-\|v\|^{+}\nu \leq v^{T}(k)\tilde{w}(k) \leq \|v\|^{+}\nu \tag{28}$$

if the norm of the feature vector remains bounded by $\|v\|^{+}$. If the LMS algorithm is tuned such that ν is small, the influence of the LMS term on the quantile estimation becomes small and can be incorporated into the bounds (25). The convergence result obtained in the previous section is therefore not affected by the LMS algorithm as soon as the invariant set $\mathcal{N}$ is reached. Due to the exponential convergence of the LMS algorithm, the quantile estimator adaption speed can be chosen slow enough such that it does not reach the relevant value region as long as the LMS did not converge to the invariant set. Faster convergence results might be achieved without violating the desired properties and will be investigated in future work.

It remains to show that the correct coverage is obtained for the disturbance $d(k)$ and the upper bound estimate $\hat{d}^{+}(v(k))$. Using (18) and the definition of $r(k)$ we obtain

$$\hat{\alpha}(\hat{r}^{+}, k) = \frac{1}{N}\sum_{i=k}^{k+N-1} \mathbb{1}\left(d(i) - v^{T}(i)\hat{w}(k) \leq \hat{r}^{+}\right). \tag{29}$$

From the definition of $\tilde{\alpha}$ we can deduce that $\hat{\alpha}(\hat{r}^{+}, k)$ converges to q^{+} despite a small error. Using this and (6) we obtain

$$q^{+} \approx \frac{1}{N}\sum_{i=k}^{k+N-1} \mathbb{1}\left(d(i) \leq \hat{d}^{+}(v(i))\right), \tag{30}$$

which demonstrates the desired property and completes the analysis of the algorithm. The same result holds for d^{-} and q^{-} and therefore the joint consideration of d^{+} and d^{-} leads to a data coverage of $q = q^{+} - q^{-}$.

III. Learning Benchmarks

The following section compares the proposed algorithm to the GPR and BLR approaches, since they are widely used in learning control engineering applications [11], [12], [29]. The source code is available online [30].

A. *Academic Examples*

We will use a datastream generated from a benchmark model with radial basis functions

$$d(k) = v^{T}(k)w + \epsilon(k), \tag{31}$$

where ϵ is generated to be zero mean, white and Gaussian. The benchmark BLR implementation is similar to [23] and [12]. The benchmark GPR uses the standard implementation of scikit-learn with hyperparameter training based on [15] and random sampling of the training samples. Both generate

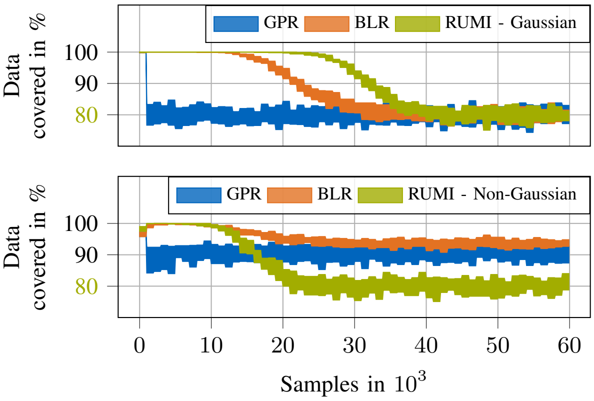


Fig. 3. Comparison of the uncertainty models obtained via GPR, BLR and RUMI on 20 datasets with Gaussian (top) and non-Gaussian (bottom) noise. The graphs depict the maximum and minimum proportion of the data covered by the upper and lower bounds. The target coverage was set to 80 %. In the Gaussian case, all estimators identified correct bounds. In the non-Gaussian case, only RUMI succeeded in identifying correct bounds.

the upper and lower bounds using multiples of posterior standard deviations to cover the desired proportion of data. The results, rendered in the upper section of Fig. 3, indicate that all estimators converge to bounds covering the desired proportion of data. The situation is different for non-Gaussian noise (modeled via two shifted Gaussian distributions). The bottom plot in Fig. 3 shows that RUMI still converges to correct bounds, but that the other approaches fail to do so. It should be noted, that depending on the actual distribution, this can lead to under- as well as overestimation. These findings indicate, that RUMI is more robust, as probabilistic assumptions can often not be met in application. This holds especially if the uncertainty is state dependent and occurs from modeling error. The major disadvantage of RUMI is its potentially slow transient response.

B. *Vehicle Motion Control Example*

This research was motivated by the requirement to estimate a reliable uncertainty model on an embedded real-time system for an autonomous race car. We benchmark the algorithms on the path-tracking controller from the TUM Roborace Team [4]. The dataset used was gathered from a sophisticated vehicle-dynamics simulation. From domain knowledge and data analysis, we deduce that the most significant influence on the uncertainty is the lateral acceleration target set by the controller. This can be explained via the uncertainty of the steady-state gain between the steering angle and the resulting lateral acceleration in the nonlinear operating region of the tire. We choose a single linear basis function depending on this feature. The uncertainty $d_{a,y}$ enters the error dynamics as follows

$$\begin{bmatrix} \dot{e} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2D\omega_0 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d_{a,y}, \qquad (32)$$

ω_0 and D are control parameters. The results for the dataset are depicted in Fig. 4 and demonstrate that the presented algorithm is capable of identifying appropriate bounds without exact knowledge about the structure of the underlying data generation process.

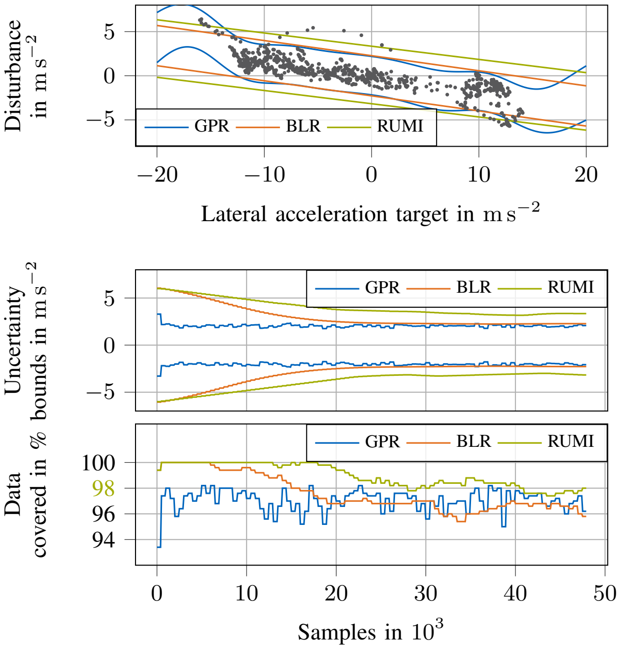


Fig. 4. Comparison of the uncertainty models obtained via GPR, BLR and RUMI on a vehicle motion controller dataset. The plots depict upper and lower bounds, the unstructured uncertainty and the proportion of the data covered by the model. The target coverage was set to 98 %. Only the RUMI approach succeeds in identifying an uncertainty model covering the desired proportion of the data. It is notable, that the uncertainty bounds differ significantly while the data coverage shows only minor differences.

TABLE I
AVERAGE COMPUTATION TIMES PER BATCH

GPR	BLR	RUMI
1349.6 ms	14.0 ms	8.9 ms

C. *Qualitative Comparison of the Approaches*

The key difficulty with the GPR approach is, that the posterior variance does not reflect the actual data variance for fixed hyperparameters. Although this problem can be solved via the application of hyperparameter optimization, this leads to a loss of the ability to tune the transient response and is computational intense. In contrast, BLR and RUMI obtain reliable variance estimates and their transient response is tunable (see Fig. 4). The drawback of both is that they require an a priori choice of basis functions. While this is an opportunity to incorporate domain knowledge, uncertainties might be difficult to predict. GPR can adapt more flexibly to the data and therefore model a wider range of functions.

From a computational point of view, the main difference is that the GPR estimator is a lazy learning algorithm in contrast to the eager learning algorithms BLR and RUMI. It stores all samples to generate the predictions, and therefore needs retraining every time the dataset is modified which requires the inversion of an $\mathbb{R}^{N_s \times N_s}$ matrix. In contrast, BLR requires only an inversion of a matrix of size $\mathbb{R}^{f \times f}$. As the number of features f is usually much smaller than the number of samples N_s, BLR has lower computational requirements compared to GPR. RUMI does not require any matrix inversion and the computational load of (8) and (18) is small. A comparison of the average computation times (taken from one dataset of the academic example on an i7-CPU) per data batch processed is depicted in Table I. Furthermore, neither BLR nor RUMI require a policy to add and remove points from a training dataset, which can be a computationally demanding task itself [29].

We wish to point out that RUMI shows similarities to BLR, which was shown to outperform GPR in [12]. In contrast, RUMI does not rely on any uncertainty assumptions, despite the existence of upper and lower bounds and sufficiently rich data. In broad terms, RUMI drops the fully Bayesian framework of the BLR, so as to achieve correct estimates in the presence of more general uncertainty.

IV. Conclusion and Future Work

We have proposed an approach for the identification of feature-dependent uncertainty models based on the concatenation of a recursive Least-Mean-Squares algorithm and a recursive quantile estimation algorithm. In contrast to other approaches, it does not rely on probabilistic assumptions and is derived in a deterministic setting. This enables correct estimation of the bounds even for the case of non-Gaussian uncertainty. A comparison to state-of-the-art approaches in learning control applications demonstrates, that they tend to over- or underestimate the bounds in this case. The algorithm relies solely on low-dimensional vector operations which favor the application on real-time systems. Future work will be the integration of into a Tube-MPC scheme and evaluation on an autonomous race car.

V. Contributions and Acknowledgments

Alexander Wischnewski initiated the idea for the paper and is responsible for the presented analysis and the implementation of the uncertainty learning algorithm. Johannes Betz and Boris Lohmann contributed equally to the conception of the research project. Furthermore, we would like to thank Thomas Herrmann, Tim Moser, Micha Obergfell, Julio Pérez, Leon Sievers, Christian Fiedler and Thomas Specker for discussions on the topic and comments on the manuscript.

References

[1] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*. Macmillan Publishing Co., 1990.

[2] A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin, "Reachability-based safe learning with gaussian processes," *Proceedings of the 53rd IEEE Conference on Decision and Control*, pp. 1424–1431, 2014.

[3] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, 2020.

[4] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, L. Hermansdorfer, T. Stahl, T. Herrmann, and M. Lienkamp, "A software architecture for the dynamic path planning of an autonomous racecar at the limits of handling," *2019 IEEE International Conference on Connected Vehicles and Expo*, 2019.

[5] A. Wischnewski, J. Betz, and B. Lohmann, "A model-free algorithm to safely approach the handling limit of an autonomous racecar," *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, 2019.

[6] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using gaussian process regression," *IEEE Transactions on Control Systems Technology*, 2019.

[7] B. Hassibi, A. H. Sayed, and T. Kailath, "H-infinity optimality of the LMS algorithm," *IEEE Transactions on Signal Processing*, vol. 44, no. 2, pp. 267–280, 1996.

[8] A. Yazidi and H. Hammer, "Multiplicative update methods for incremental quantile estimation," *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 746–756, 2019.

[9] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[10] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2019.

[11] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," *Proceedings of the 57th IEEE Conference on Decision and Control*, pp. 6059–6066, 2018.

[12] C. D. McKinnon and A. P. Schoellig, "Learning probabilistic models for safe predictive control in unknown environments," *Proceedings of the 18th European Control Conference*, pp. 2472–2479, 2019.

[13] S. B. Liu and M. Althoff, "Reachset conformance of forward dynamic models for the formal analysis of robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.

[14] J. Umlauft, T. Beckers, M. Kimmel, and S. Hirche, "Feedback linearization using gaussian processes," *Proceedings of the IEEE 56th Conference on Decision and Control*, pp. 5249–5255, 2017.

[15] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[16] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," *Proceedings of the 18th International Conference on Neural Information Processing Systems*, 1257–1264, 2005.

[17] F. Meier and S. Schaal, "Drifting gaussian processes with varying neighborhood sizes for online model learning," *Proceedings of the 2016 IEEE International Conference on Robotics and Automation*, pp. 264–269, 2016.

[18] D. Nguyen-Tuong, M. Seeger, J. Peters, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, "Local gaussian process regression for real time online model learning and control," *Advances in Neural Information Processing Systems 21: Proceedings of the 2008 Conference*, pp. 1193–1200, 2008.

[19] F. Berkenkamp, "Safe exploration in reinforcement learning: Theory and applications in robotics," Ph.D. dissertation, ETH Zurich, 2019.

[20] J.-P. Calliess, "Lazily adapted constant kinky inference for nonparametric regression and model-reference adaptive control," *arXiv.org*, 2016.

[21] M. F. Huber, "Recursive gaussian process: On-line regression and learning," *Pattern Recognition Letters*, pp. 85–91, 2014.

[22] A. Svensson, A. Solin, S. Särkkä, and T. B. Schön, "Computationally efficient Bayesian learning of Gaussian process state space models," *Proceedings of 19th International Conference on Artificial Intelligence and Statistics*, pp. 213–221, 2016.

[23] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT Press, 2013.

[24] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," *arXiv.org*, 2015.

[25] R. Koenker, *Quantile Regression*. Cambridge University Press, 2005.

[26] M. Campi, G. Calafiore, and S. Garatti, "Interval predictor models: Identification and reliability," *Automatica*, vol. 45, no. 2, pp. 382–392, 2009.

[27] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Springer International Publishing, 2015, pp. 1–26.

[28] G. Strang, *Linear algebra and its applications*. Brooks Cole, 2006.

[29] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.

[30] A. Wischnewski, 2020. [Online]. Available: `https://github.com/TUMFTM/RUMI`.