Vehicle System Dynamics

International Journal of Vehicle Mechanics and Mobility

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/nvsd20

Tube model predictive control for an autonomous race car

A. Wischnewski, M. Euler, S. Gümüs & B. Lohmann

VEHICLE SYSTEM DYNAMICS
2022, VOL. 60, NO. 9, 3151–3173
https://doi.org/10.1080/00423114.2021.1943461

Tube model predictive control for an autonomous race car

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

Chair of Automatic Control, Technical University of Munich, Munich, Germany

ABSTRACT
Nonlinear effects and external disturbances can severely impact the control of an autonomous race car at the handling limits. State-of-the-art approaches do not take 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 to guarantee constraint satisfaction. The resulting optimisation problem can be posed in the form of a standard quadratic programme by tightening the input and state constraints. 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 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.

ARTICLE HISTORY
Received 11 December 2020
Revised 22 April 2021
Accepted 2 June 2021

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 control

CONTACT A. Wischnewski ✉ alexander.wischnewski@tum.de

*The authors contributed equally.

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 Heilmeier et al. [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 programme (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

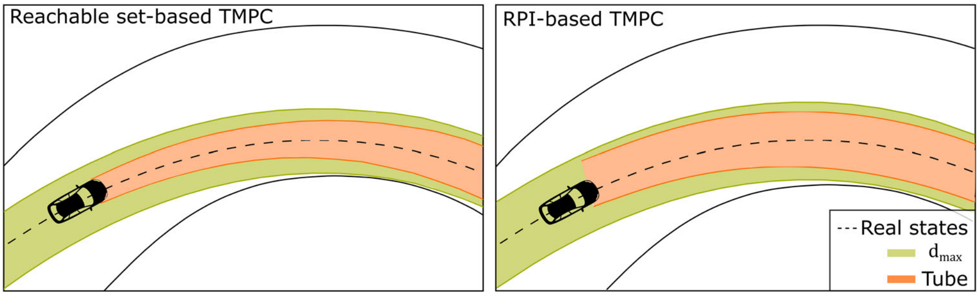

Figure 1. Comparison of the reachable set and RPI-based TMPC approaches with the lateral deviation limit $d_{\max}$.

concepts exist (see Figure 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 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

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

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

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

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

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

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

$$p_0 = x_t, \tag{8a}$$

$$p_{k+1} = Ap_k + Br_k, \tag{8b}$$

$$M_{k+1} = (1 + c_k^{-1})(A + BK)M_k(A + BK)^T + (1 + c_k)D\tilde{M}D^T, \tag{8c}$$

$$c_k = \sqrt{\mathrm{Tr}((A + BK)M_k(A + BK)^T)/\mathrm{Tr}(D\tilde{M}D^T)} \tag{8d}$$

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 Equations (8c) and (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:

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

s.t.

$$p_{k+1} = Ap_k + Br_k \tag{9b}$$

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

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

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

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

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 $\mathcal{X}_k \in \mathbb{X}_k$ (Equation (9d)) and input constraints $\mathcal{U}_k \in \mathbb{U}_k$ (Equation (9e)) have been reformulated using the approach presented by van Hessem and Bosgra [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 [28,59]. 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 programme. Furthermore, we emphasise 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 tyre models as they require more parameter identification work and lead to complex numerical optimisation problems. All these tyre 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,

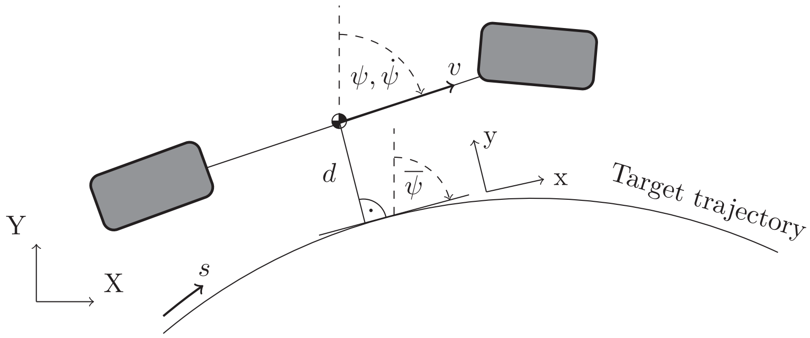

Figure 2. Vehicle point mass model in error coordinates.

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:

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

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

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

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 Betz et al. [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 raceline.

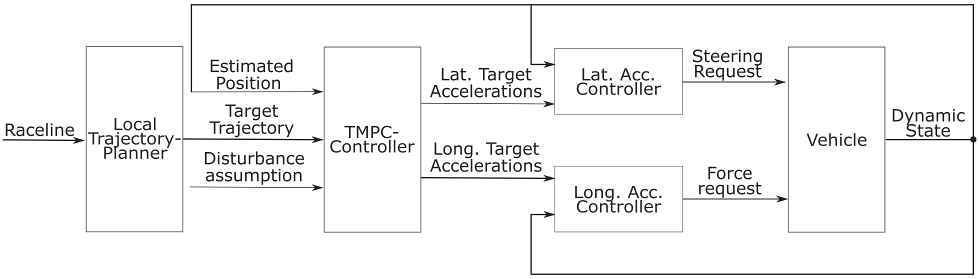

Figure 3. Autonomous racing software architecture including the TMPC controller.

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 programme

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 programme 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 Equation (13) and use Equation (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 Appendix.

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 Equation (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 $p^T Pp$ 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 accelerations, instead of the exact nonlinear transformations:

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

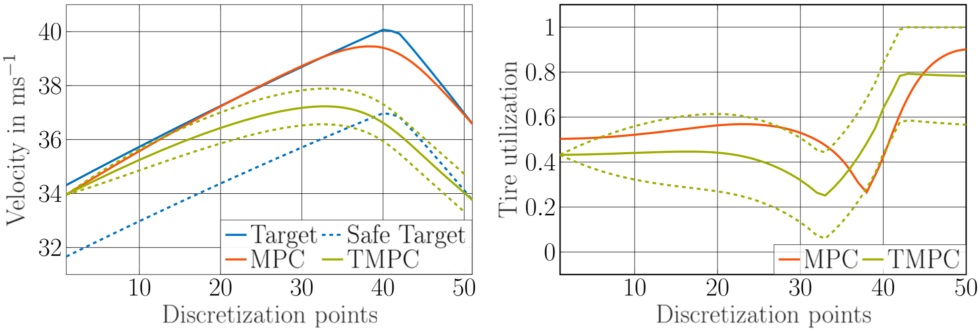

Figure 4. Terminal set design by using a safe target velocity.

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

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 Equation (9e), a suitable velocity scale factor θ_v to obtain a safe target velocity $v_\mathrm{T} = \theta_v \bar{v}$ is determined

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

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 Figure 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 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 Figure 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 sensor 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 Figure 6. Its combination of tight corners with high-speed sections make

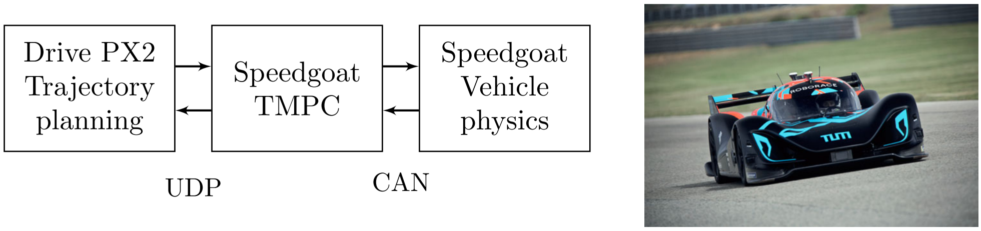

Figure 5. Hardware-in-the-loop setup and corresponding Roborace DevBot 2.0 vehicle.

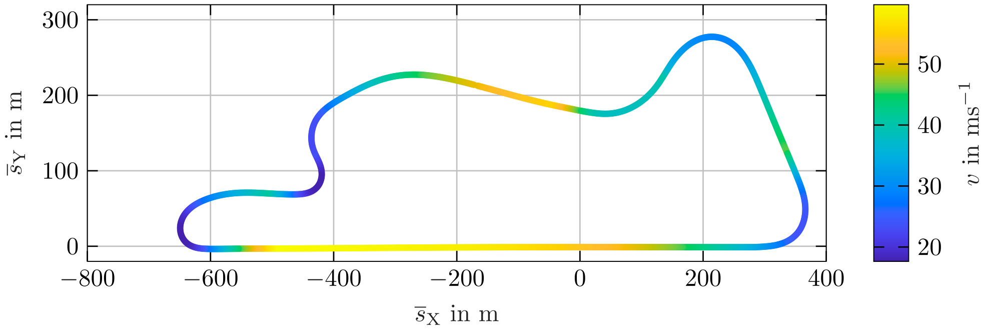

Figure 6. Monteblanco track layout.

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 limit the accelerations

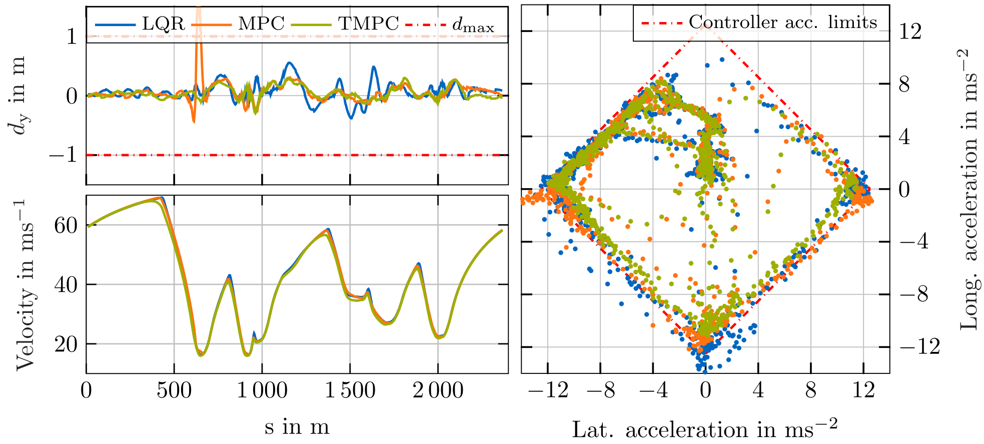

Figure 7. Lateral deviation, velocity, and gg-diagram (with controller acceleration limits) of the LQR, MPC and TMPC controller.

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.

Figure 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 tyre limits. Driving more cautiously with the TMPC controller increases the lap-time to 64.67 s but leads to a closed-loop behaviour 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 Figure 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

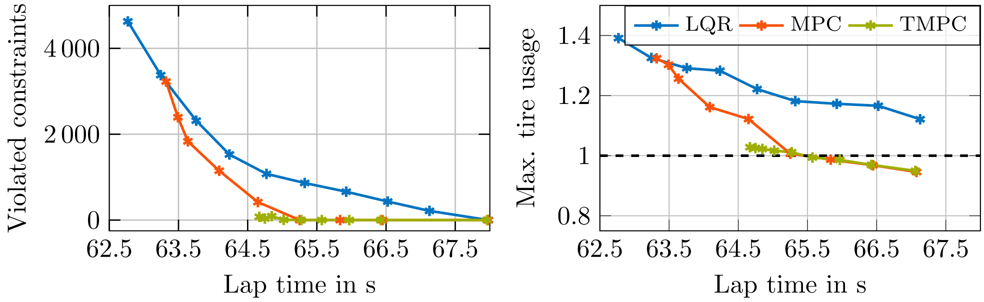

Figure 8. Number of violated constraints and maximum tyre usage in comparison to the achieved lap-time for the controllers on a flying lap.

MPC and LQR controller only achieve a similar safety level if the accelerations limits are globally reduced which 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 Figure 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 Figure 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

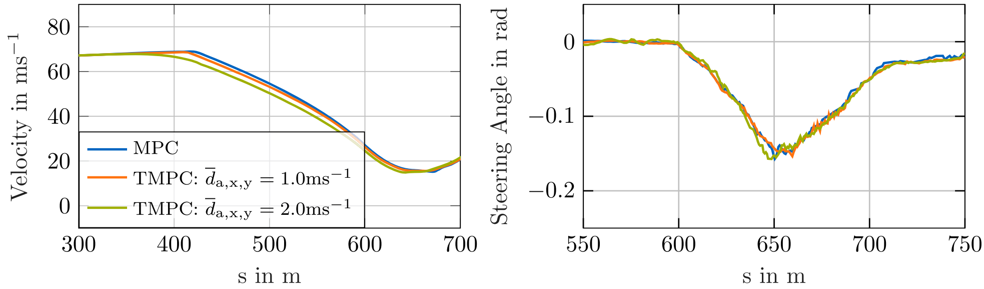

Figure 9. Comparison of the vehicles velocity profile and steering angle for the MPC and two TMPC controllers with different set sizes.

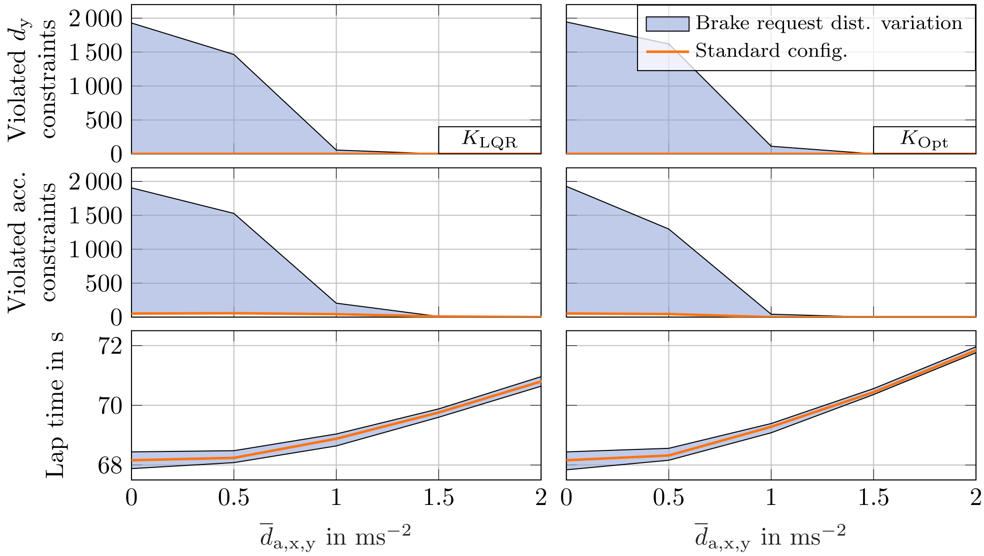

Figure 10. Comparison of violated constraints and lap-time for K_{LQR} (left) and K_{Opt} (right) with different disturbance sets for varied brake force disturbances.

behaviour and act cautiously in state-space directions where this might impact future constraint satisfaction. We, therefore, advise staying with the LQR controller due to the easy design process. The results also indicate that disturbance limits of 1 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.

Acknowledgments

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.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

A. Wischnewski http://orcid.org/0000-0001-6168-8556

References

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

[2] Liniger A, Domahidi A, Morari M. Optimization-based autonomous racing of 1:43 scale rc cars. Optim Control Appl 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, Seattle, USA; 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. Veh Syst Dyn. 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 race car at the limits of handling. In: Proceedings of the 2019 IEEE International Conference on Connected Vehicles and Expo, Graz, Austria; 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, Auckland, New Zealand; 2019. p. 3149–3154.

[7] Falcone P, Borrelli F, Asgari J, et al. Predictive active steering control for autonomous vehicle systems. IEEE Trans Control Syst Technol. 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), Los Angeles, USA; 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, Las Palmas de Gran Canaria; 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, Ypsilanti, USA; 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, Hawaii; 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, San Diego, USA; 1997. p. 4732–4737.

[13] Menhour L, Fliess M, et al. Coupled nonlinear vehicle control: flatness-based setting with algebraic estimation techniques. Control Eng Pract. 2014;22:135–146.

[14] Villagra J, Mounier H, et al. Flatness-based vehicle steering control strategy with sdre feedback gains tuned via a sensitivity approach. IEEE Trans Control Syst Technol. 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, Grenoble, France; 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, Gold Coast City, Australia; 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, Kos, Greece; 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, Firenze, Italy; 2013. p. 3073–3078.

[19] Attia R, Orjuela R, Basset M. Combined longitudinal and lateral control for automated vehicle guidance. Veh Syst Dyn. 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), The Hague, Netherlands; 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, Yokohama, Japan; 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, Tokyo, Japan; 2018.

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

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

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

[26] Subosits JK, Gerdes JC. From the racetrack to the road: real-time trajectory replanning for autonomous driving. IEEE Trans Intell Veh. 2019;4(2):309–320.

[27] Alcalá E. Autonomous racing using linear parameter varying-model predictive control. Control Eng Pract. 2020;95:Article ID 104270.

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

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

[30] Rosolia U, Borrelli F. Learning how to autonomously race a car: A predictive control approach. IEEE Trans Control Syst Technol. 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), Philadelphia, USA; 2019.

[33] Gao Y, Gray A, Tseng HE, et al. A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. Veh Syst Dyn. 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, Yokohama, Japan; 2017.

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

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

[37] Campo PJ, Morari M. Robust model predictive control. In: 1987 American Control Conference, Minneapolis, USA; 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. Eur J 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 Trans Automat Contr. 1998;43(8):1136–1142.

[41] Mayne D. Robust and stochastic model predictive control: are we going in the right direction? Annu Rev 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, Pretoria, South Africa; 2007. p. 36–41.

[44] Rakovic SV, Kouvaritakis B, Cannon M, et al. Parameterized tube model predictive control. IEEE Trans Automat Contr. 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, Toulouse, France; 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, Miami Beach, USA; 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, Seoul, Korea; 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, Milwaukee, USA; 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. J Adapt Control Signal Process. 2003;17(3):195–207.
[52] Le V, Alamo T, Camacho EF, et al. A new approach for guaranteed state estimation by zonotopes. IFAC Proc Vol. 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, Orlando, USA; 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. Boston: Birkhäuser; 1997. p. 229–243. (Progress in systems and control theory; vol. 277).
[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, Anchorage, USA; 2002. p. 2023–2028.
[58] Polack P, Altché F. The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In: Proceedings of the 2017 IEEE Intelligent Vehicles Symposium, Los Angeles, USA; 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. Berlin Heidelberg: Springer; 1998. p. 171–253.
[60] Betz J, Heilmeier A, Wischnewski A, et al. Autonomous driving – a crash explained in detail. Appl Sci. 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. 2000 Sep.
[63] Stellato B, Banjac G, Goulart P, et al. OSQP: an operator splitting solver for quadratic programs. Math Program Comput. 2020;12:637–672.
[64] Open-source software repository. Chair of automotive technology. Technical University of Munich; 2019. Available from: https://github.com/TUMFTM.

Appendix. Details on linearisation procedure

A.1 Longitudinal acceleration

Derivation of linearisation for Equation (15):

$$a_{\mathrm{x}}(s, d, \dot{d}, \Delta v, \Delta a_x) = \frac{\partial \bar{v}(s)}{\partial s} \frac{(\bar{v}(s) - \Delta v) \cos\left(\arcsin\left(\frac{\dot{d}}{(\bar{v}(s)-\Delta v)}\right)\right)}{1 - d\bar{\kappa}(s)} - \Delta a_{\mathrm{x}}. \tag{A1}$$

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

$$a_{\mathrm{x}}(d, \dot{d}, \Delta v, \Delta a_x) = \bar{a}_x \frac{(\bar{v} - \Delta v) \cos\left(\arcsin\left(\frac{\dot{d}}{(\bar{v}-\Delta v)}\right)\right)}{1 - d\bar{\kappa}} - \Delta a_{\mathrm{x}}. \tag{A2}$$

The partial derivatives are

$$\frac{\partial a_x}{\partial d} = \bar{a}_x \bar{\kappa} \frac{(\bar{v} - \Delta v)\sqrt{1 - \frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}}}{(1 - d\bar{\kappa})^2} \tag{A3a}$$

$$\frac{\partial a_x}{\partial \dot{d}} = -\bar{a}_x \frac{\dot{d}}{(\bar{v} - \Delta v)\,(1 - d\bar{\kappa})\sqrt{1 - \frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}}} \tag{A3b}$$

$$\frac{\partial a_x}{\partial \Delta v} = -\bar{a}_x \frac{1}{(1 - d\bar{\kappa})\sqrt{1 - \frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}}} \tag{A3c}$$

$$\frac{\partial a_x}{\partial \Delta a_x} = -1 \tag{A3d}$$

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

$$a_x \approx \bar{a}_x \bar{\kappa} v_{\mathrm{p}} d - \bar{a}_x \Delta v - \Delta a_x + \bar{a}_x v_{\mathrm{p}} \tag{A4}$$

A.2 Lateral acceleration

Derivation of linearisation for Equation (16):

$$\begin{aligned} a_{\mathrm{y}}(s, d, \dot{d}, \Delta v, a_x, \Delta a_x) = {} & \frac{\Delta a_{\mathrm{y}}}{\cos\left(\arcsin\left(\frac{\dot{d}}{(\bar{v}(s)-\Delta v)}\right)\right)} \\ & + \bar{\kappa} \frac{\cos\left(\arcsin\left(\frac{\dot{d}}{(\bar{v}(s)-\Delta v)}\right)\right)}{1 - d\bar{\kappa}(s)} (\bar{v} - \Delta v)^2 \\ & - a_x\left(d, \dot{d}, \Delta v, \Delta a_x\right) \tan\left(\arcsin\left(\frac{\dot{d}}{(\bar{v}(s) - \Delta v)}\right)\right) \end{aligned} \tag{A5}$$

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

$$\begin{aligned} a_{\mathrm{y}}(d, \dot{d}, \Delta v, \Delta a_x, \Delta a_y) = {} & \frac{\Delta a_{\mathrm{y}}}{\cos\left(\arcsin\left(\frac{\dot{d}}{(\bar{v}-\Delta v)}\right)\right)} \\ & + \bar{\kappa}\frac{\cos\left(\arcsin\left(\frac{\dot{d}}{(\bar{v}-\Delta v)}\right)\right)}{1 - d\bar{\kappa}}(\bar{v} - \Delta v)^2 \\ & - a_x\left(d, \dot{d}, \Delta v, \Delta a_x\right)\tan\left(\arcsin\left(\frac{\dot{d}}{(\bar{v}-\Delta v)}\right)\right) \end{aligned} \tag{A6}$$

The partial derivatives are

$$\frac{\partial a_y}{\partial d} = \bar{\kappa}^2\frac{\cos\left(\arcsin\left(\frac{\dot{d}}{(\bar{v}-\Delta v)}\right)\right)}{(1 - d\bar{\kappa})^2}(\bar{v} - \Delta v)^2 - \frac{\partial a_x}{\partial d}\tan\left(\arcsin\left(\frac{\dot{d}}{(\bar{v}-\Delta v)}\right)\right) \tag{A7a}$$

$$\begin{aligned} \frac{\partial a_y}{\partial \dot{d}} = {} & \frac{\Delta a_y \dot{d}}{(\bar{v}-\Delta v)^2\left(1 - \frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}\right)^{3/2}} \\ & - \bar{\kappa}\frac{\dot{d}}{(1 - d\bar{\kappa})\sqrt{1 - \frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}}} \\ & - \frac{\partial a_x}{\partial \dot{d}}(d, \dot{d}, \Delta v, \Delta a_x)\tan\left(\arcsin\left(\frac{\dot{d}}{(\bar{v}-\Delta v)}\right)\right) \\ & - a_x(d, \dot{d}, \Delta v, \Delta a_x)\left(\frac{\dot{d}^2}{(\bar{v}-\Delta v)^3\left(1 - \frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}\right)^{3/2}} + \frac{1}{(\bar{v}-\Delta v)\sqrt{1 - \frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}}}\right) \end{aligned} \tag{A7b}$$

$$\begin{aligned} \frac{\partial a_y}{\partial \Delta v} = {} & \frac{\Delta a_y \dot{d}^2}{(\bar{v}-\Delta v)^3\left(1 - \frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}\right)^{3/2}} \\ & \times \frac{\bar{\kappa}}{1 - d\bar{\kappa}}\left(\frac{\dot{d}^2}{(\bar{v}-\Delta v)\sqrt{1 - \frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}}} - 2\,(\bar{v}-\Delta v)\sqrt{1 - \frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}}\right) \\ & - \frac{\partial a_x}{\partial \Delta v}(d, \dot{d}, \Delta v, \Delta a_x)\tan\left(\arcsin\left(\frac{\dot{d}}{(\bar{v}-\Delta v)}\right)\right) \\ & - a_x(d, \dot{d}, \Delta v, \Delta a_x)\left(\frac{\dot{d}}{(\bar{v}-\Delta v)^2}\sqrt{1 - \frac{\dot{d}^2}{(\bar{v}-\Delta v)^2}} + \frac{\dot{d}^2}{(\bar{v}-\Delta v)^3\left(1 - \frac{\dot{d}^3}{(\bar{v}-\Delta v)^4}\right)^{3/2}}\right) \end{aligned} \tag{A7c}$$

$$\frac{\partial a_y}{\partial \Delta a_x} = \tan\left(\arcsin\left(\frac{\dot{d}}{(\bar{v}-\Delta v)}\right)\right) \tag{A7d}$$

$$\frac{\partial a_y}{\partial \Delta a_y} = \frac{1}{\cos\left(\arcsin\left(\frac{\dot{d}}{(\bar{v}-\Delta v)}\right)\right)} \tag{A7e}$$

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

$$a_y \approx \bar{\kappa}^2 v_{\mathrm{p}}^2 d - \frac{a_x(d, \dot{d}, \Delta v, \Delta a_x)}{v_{\mathrm{p}}}\dot{d} - 2\bar{\kappa} v_{\mathrm{p}} \Delta v + \Delta a_y + \bar{\kappa} v_{\mathrm{p}}^2 \tag{A8}$$

A.3 Linearisation of tyre constraints

Finally we can reformulate the tyre constraints in Equation (11)

$$\left|\frac{a_{\mathrm{x}}}{a_{\mathrm{x,max}}}\right| + \left|\frac{a_{\mathrm{y}}}{a_{\mathrm{y,max}}}\right| \le 1, \tag{A9}$$

to be

$$\pm\, a_{\mathrm{x}} a_{\mathrm{y,max}} \pm a_{\mathrm{y}} a_{\mathrm{x,max}} \le a_{\mathrm{x,max}} a_{\mathrm{y,max}}. \tag{A10}$$

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

$$\begin{aligned} &\pm\left(\bar{a}_x \bar{\kappa} v_{\mathrm{p}} d - \bar{a}_x \Delta v - \Delta a_x + \bar{a}_x v_{\mathrm{p}}\right) a_{\mathrm{y,max}} \\ &\pm\left(\bar{\kappa}^2 v_{\mathrm{p}}^2 d - \frac{a_x(d, \dot{d}, \Delta v, \Delta a_x)}{v_{\mathrm{p}}}\dot{d} - 2\bar{\kappa} v_{\mathrm{p}} \Delta v + \Delta a_y + \bar{\kappa} v_{\mathrm{p}}^2\right) a_{\mathrm{x,max}} \le a_{\mathrm{x,max}} a_{\mathrm{y,max}}. \end{aligned} \tag{A11}$$