NMPC for Racing Using a Singularity-Free Path-Parametric Model with Obstacle Avoidance

Daniel Kloeser [*,***] **Tobias Schoels** [*] **Tommaso Sartor** [*]
Andrea Zanelli [*] **Gianluca Frison** [*] **Moritz Diehl** [*,**]

[*] *Department of Microsystems Engineering, University of Freiburg (e-mail: daniel.kloeser@rwth-aachen.de, {tommaso.sartor, andrea.zanelli, gianluca.frison, tobias.schoels, moritz.diehl}@imtek.de)*
[**] *Department of Mathematics, University of Freiburg*
[***] *Chair of Automatic Control, Christian-Albrechts-University Kiel*

Abstract: This work presents the real-time control of 1:43 scale autonomous race cars using nonlinear model predictive control based on a singularity-free prediction model. This model allows the car to drive at both low and high speeds and in stop-and-go maneuvers. Additional constraints are imposed in the optimal control problem to ensure the validity of the model assumptions. Moreover, the control scheme is capable of avoiding obstacles online. The experimental results show that the proposed method converges to nearly time-optimal behavior by maximizing the progress on the track and achieves competitive lap time results.

Keywords: Nonlinear and optimal automotive control, obstacle avoidance, automotive system identification and modelling, autonomous vehicles

1. INTRODUCTION

Autonomous car racing poses a particular challenge for control methods, because the vehicle is driven at its handling limits. The objective is to reach the minimal lap time while keeping the vehicle on the track and avoiding obstacles. A common approach for testing methods tailored for this purpose is the use of remote controlled miniature cars. These are a low cost and low risk alternative with similar properties and challenges as full-size vehicles.

Many methods for autonomous racing are based on nonlinear model predictive control (NMPC), because nonlinear dynamics and dynamical constraints, e.g. velocity or acceleration limits, can be imposed explicitly. As computational power increases, it allows for longer prediction horizons of the NMPC and mitigates the need for hierarchical architectures such as (Gao et al., 2010). Therefore, most recent methods use a single-layer architecture (Perantoni and Limebeer, 2014; Verschueren et al., 2014; Liniger et al., 2015; Rosolia et al., 2017) that differs in the approach taken to approximate time-optimal behavior, avoid obstacles, and model vehicle dynamics. The latter reach from simple approximate models (Lima et al., 2018) over sophisticated physical formulations (Kabzan et al., 2019) to machine learning based approaches (Hewing et al., 2018).

Related Work: Several NMPC formulations have been proposed to approximate time-optimal behavior. Liniger et al. (2015) present a model predictive contouring control

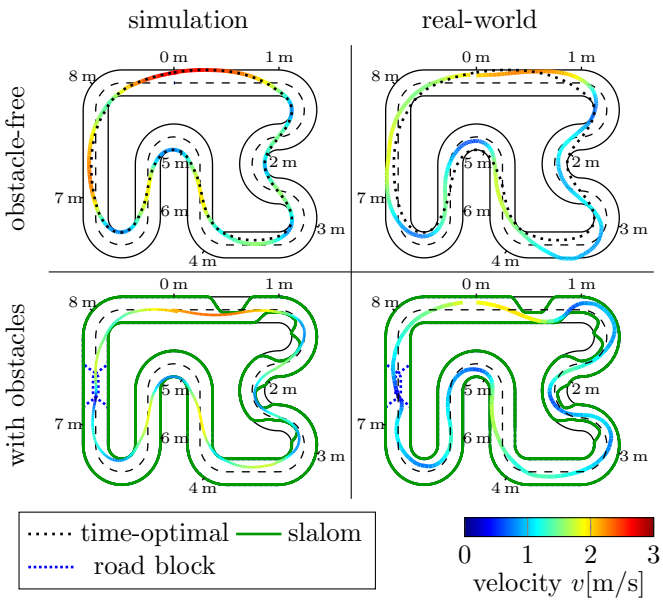

Fig. 1. Real-world and simulation results: On the left are simulation results with and without obstacles. On the right are the same scenarios on the real-world setup. The scenario without obstacles is also compared to the offline optimal trajectory illustrated by the dotted black line. We challenge the proposed approach by a slalom curve on the chicane (0.5 m to 3 m). Moreover, we place a road block at 7.2 m which is removed as soon as the vehicle comes to a complete stop.

⋆ This research was supported by the German Federal Ministry for Economic Affairs and Energy (BMWi) via eco4wind (0324125B) and DyConPV (0324166B), and by DFG via Research Unit FOR 2401.

that maximizes progress on the track. Verschueren et al. (2014) use a formulation to minimize the final time for a fixed-length spatial prediction horizon while Lima et al. (2018) optimize the longitudinal speed and the lateral displacement depending on the track's curvature in order to maximize progress.

The performance of the NMPC heavily relies on the prediction model's accuracy. Liniger et al. (2015) and Perantoni and Limebeer (2014) derive a detailed model using Pacejka's magic formula (Pacejka, 2006). This is a common approach for modelling tire-ground interaction and has proven to be working nicely, when operating at handling limits. Nonetheless, the tire slip angle enters the formula such that it has a singularity at zero speed. This introduces high stiffness to a system at low speed. Verschueren et al. (2014) instead use a slip-free tire model, but do not account for lateral accelerations which leads to poor performance at high speed. Kabzan et al. (2019) use a linear blend between a simple singularity-free model at low speeds and a detailed model at high speeds. The model assumptions of Lima et al. (2018) are similar to the ones in this paper as they utilize a slip-free tire model and limit the longitudinal and lateral accelerations. However, the acceleration constraints are based on the GG-diagram while we treat lateral and longitudinal accelerations independently.

Polack et al. (2017) investigate the validity of the slip-free tire model assumption based on the lateral acceleration. Therefore, they compare a detailed 9-DOF model with a 3-DOF kinematic model in simulation. They conclude that the model is accurately matched until 0.5 g assuming a large friction coefficient. Also Kong et al. (2015) compare the prediction errors of a kinematic and a dynamic model and motivate the usage of a kinematic model in the control design. However, they do not operate the vehicle at its handling limits.

Contribution: This paper presents an optimal control approach that maximizes progress on the track and thereby approximates time-optimal racing on a real-world 1:43 race car setup. For this purpose, we propose a singularity-free model and enforce its validity by limiting the lateral acceleration. Thereby, the setup is capable of start-stop maneuvers and static obstacle avoidance at low and high speeds. The obstacles are modeled as deformations in the road boundary and are mathematically described by piecewise cubic polynomials. The proposed approach is validated experimentally in both simulation and real-world.

Structure: Section 2 states the system model and validates the model assumptions based on real-world experiments. Section 3 presents the optimal control formulation including obstacle avoidance. Section 4 shows simulation as well as experimental results, while Section 5 summarizes the paper and gives a brief outlook.

2. SINGULARITY-FREE MODEL

The prediction model plays a key role in the design of NMPC methods. We use a dynamic model that combines a kinematic model with the longitudinal dynamics only.

2.1 Race Car Model

A generic dynamic bicycle model, similar to the ones used by Liniger et al. (2015) or Verschueren et al. (2016), is illustrated in Figure 2. The lateral and longitudinal forces $F_{\mathrm{y}}^{\mathrm{r}}$, $F_{\mathrm{y}}^{\mathrm{f}}$ and $F_{\mathrm{x}}^{\mathrm{d}}$ describe the forces acting on the tires. The parameters l_{r} and l_{f} denote the distance between the rear and front wheel to the center of gravity (CG), respectively. The steering angle δ quantifies the deflection of the front wheel, while the back wheel is fixed. The vehicle's position and orientation in the world frame (X-Y-frame) are given by p_{X}, p_{Y} and ψ, respectively. Utilizing Newton's second law, the dynamic equations with respect to CG can be described by

$$\dot{p}_{\mathrm{X}} = v_{\mathrm{x}}\cos(\psi) - v_{\mathrm{y}}\sin(\psi), \tag{1a}$$
$$\dot{p}_{\mathrm{Y}} = v_{\mathrm{x}}\sin(\psi) + v_{\mathrm{y}}\cos(\psi), \tag{1b}$$
$$\dot{\psi} = \omega \tag{1c}$$
$$m\dot{v}_{\mathrm{x}} = F_{\mathrm{x}}^{\mathrm{d}} - F_{\mathrm{y}}^{\mathrm{f}}\sin(\delta) + m\omega v_{\mathrm{y}}, \tag{1d}$$
$$m\dot{v}_{\mathrm{y}} = F_{\mathrm{y}}^{\mathrm{r}} + F_{\mathrm{y}}^{\mathrm{f}}\cos(\delta) - m\omega v_{\mathrm{x}}, \tag{1e}$$
$$I_{\mathrm{z}}\dot{\omega} = F_{\mathrm{y}}^{\mathrm{r}} l_{\mathrm{r}} - F_{\mathrm{y}}^{\mathrm{f}} l_{\mathrm{f}}\cos(\delta), \tag{1f}$$

where m is the mass of the vehicle, I_{z} is the inertia while v_{x} and v_{y} denote the car's velocity in the vehicle frame (x-y-frame).

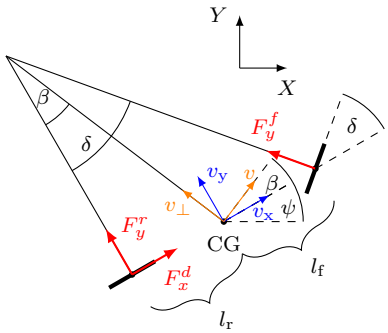

Fig. 2. Generic dynamic bicycle model.

Similar to Verschueren et al. (2014), a slip-free tire model is assumed, meaning that the rear and front wheel of the vehicle fully absorb the lateral forces $F_{\mathrm{y}}^{\mathrm{f}}$ and $F_{\mathrm{y}}^{\mathrm{r}}$. In this case, the movement of the vehicle can be described by a circular path where the CG moves in the direction $\psi + \beta$. The side-slip angle β is defined as depicted in Figure 2. Assuming that the steering angle δ is small, β can be simplified by

$$\beta = \arctan\left(\frac{l_{\mathrm{r}}}{l_{\mathrm{r}} + l_{\mathrm{f}}}\tan\delta\right) \approx \frac{l_{\mathrm{r}}}{l_{\mathrm{r}} + l_{\mathrm{f}}}\delta. \tag{2}$$

Rotating the reference frame to v and $v_\perp$, we obtain

$$\dot{p}_{\mathrm{X}} = v\cos(\psi + \beta), \tag{3a}$$
$$\dot{p}_{\mathrm{Y}} = v\sin(\psi + \beta), \tag{3b}$$
$$\dot{\psi} = \frac{v}{l_{\mathrm{r}}}\sin\beta, \tag{3c}$$
$$\dot{v} = F_{\mathrm{x}}^{\mathrm{d}}\cos\beta. \tag{3d}$$

Because we assume that there is no tire slip, the lateral force $m\dot{v}_\perp = ma_\perp = F_{\mathrm{x}}^{\mathrm{d}}\sin(\beta) + \frac{mv^2}{l_{\mathrm{r}}}\sin(\beta)$ is neglected. To ensure that this assumption is valid, the lateral force

has to be bounded by a constraint in the optimal control problem (OCP).

Spengler and Gammeter (2010) derive the longitudinal force as a function of the duty cycle D, a constant, and a second order friction term

$$F_{\mathrm{x}}^{\mathrm{d}} = (c_{\mathrm{m}_1} - c_{\mathrm{m}_2} v)D - c_{\mathrm{r}_2} v^2 - c_{\mathrm{r}_0} \tanh(c_{\mathrm{r}_3} v). \quad (4)$$

2.2 Path-Parametric Reformulation

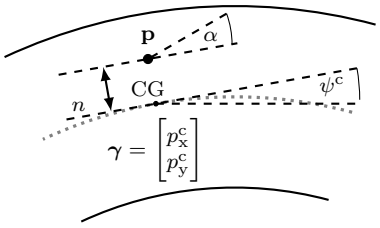

Fig. 3. Path-parametric model.

For more compact formulation, we follow the approach of Frasch et al. (2013), where the car's pose is projected onto the center line of the race track. This is illustrated in Figure 3. The path of center line $\boldsymbol{\gamma}$ is parameterized by the arc length s. The position $\mathbf{p}$ and orientation α relative to the path are given by

$$\mathbf{r}(s,t) = \mathbf{p}(\mathbf{t}) - \boldsymbol{\gamma}(s), \quad (5)$$

$$\alpha(s,t) = \psi(t) - \psi^{\mathrm{c}}(s), \quad (6)$$

where ψ^{c} is the heading of the center line. We describe the position s^* on the center line that is closest to $\mathbf{p}$ as

$$s^*(t) = \arg\min_s \frac{1}{2}||\mathbf{r}(s,t)||_2^2, \quad (7)$$

which states an optimization problem itself that can be solved under the assumption that s^* is known at the initial time $t = 0$. The speed on the center line $\dot{s}$ is then given by

$$\dot{s} = \frac{v(t)\cos(\alpha(s,t) + \beta)}{1 - n(s,t)\kappa^c(s)}, \quad (8)$$

where κ^c describes the center line's curvature and n is the minimal distance between the vehicle and the center line. The curvature κ^c is a continuous function of s, which we approximate by a third-order B-spline. To guarantee uniqueness of the projection, $n\kappa^c < 1$ must hold. The minimal distance to the center line n is given by

$$n(s,t) = \begin{bmatrix} \cos(\psi^{\mathrm{c}}(s)) \\ -\sin(\psi^{\mathrm{c}}(s)) \end{bmatrix}^T \begin{bmatrix} p_{\mathrm{y}}(t) - p_{\mathrm{y}}^{\mathrm{c}}(s) \\ p_{\mathrm{x}}(t) - p_{\mathrm{x}}^{\mathrm{c}}(s) \end{bmatrix}. \quad (9)$$

The resulting dynamic model has the states $\boldsymbol{x} = [s,\ n,\ \alpha,\ v]^T$ and is given by

$$\dot{s} = \frac{v\cos(\alpha + \beta)}{1 - n\kappa^c}, \quad (10a)$$

$$\dot{n} = v\sin(\alpha + \beta), \quad (10b)$$

$$\dot{\alpha} = \dot{\psi} - \kappa^c \dot{s}, \quad (10c)$$

$$\dot{v} = \frac{F_{\mathrm{x}}^{\mathrm{d}}}{m}\cos(\beta). \quad (10d)$$

In contrast to Verschueren et al. (2014), we do not conduct a complete spatial reformulation, but discretize the model with respect to physical time t. This leads to a singularity-free model that is not restricted to $\dot{s} > 0$. A disadvantage is, however, that we cannot formulate the obstacles as simple state bounds.

2.3 Model Identification and Validation

A simple but accurate model is key to reach good performance with NMPC methods. This includes the identification of model parameters that commonly involves solving nonlinear and nonconvex optimization problems. The parameters for the kinematic model, l_r and l_f, can be measured directly. Moreover, the parameters governing longitudinal dynamics are acquired by acceleration and deceleration experiments. The identified parameters can be found in Table 1.

Table 1. Identified parameters

Parameter	m	c_{m1}	c_{m2}	c_{r0}	c_{r2}	c_{r3}
Unit	kg	kgm/s^2	kg/s	kgm/s^2	kg/m	s/m
Value	0.043	0.28	0.05	0.006	0.011	5

The use of a slip-free tire model as described above reduces the model complexity, but it has a limited validity range that needs to be enforced. In order to identify the limits of this assumption, we conduct an experimental comparison between simulation and real-world. Similar to Polack et al. (2017), we track a circle with constant radius R and different speed references leading to a certain later accelerations $a_\perp$. We then determine the discrepancy between the required steering angle δ in simulation and in the real-world.

Figure 4 compares the effective steering angles obtained in simulations to measurements from real-world experiments depending on the lateral acceleration $a_\perp$. With increasing $a_\perp$, we observe an under steering behavior of the car which is not covered by the model. Experimental validations of the closed-loop behavior show that a threshold of $4\,\mathrm{m/s^2}$ achieves best performance for our setup. Note that this threshold is imposed as nonlinear constraint in the OCP formulation in (11g).

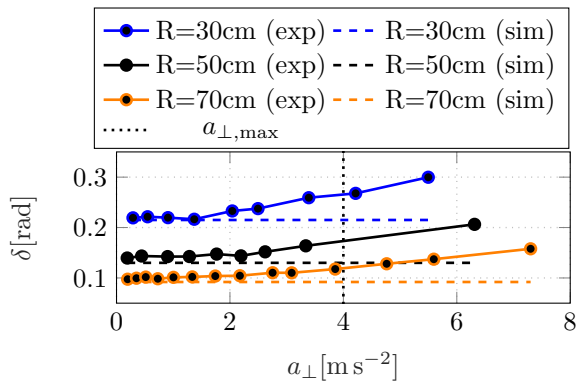

Fig. 4. Influence of the lateral acceleration $a_\perp$ on the effective steering angle δ in simulation and real-world. These results were obtained by tracking a circle of constant radius R. We report the measured average steering angle δ for a fixed lateral acceleration $a_\perp$. The results show an increasing discrepancy between simulation and real-world for increasing $a_\perp$ due to an understeering behavior of the car.

3. PROBLEM FORMULATION AND SOLUTION

We formalize the general obstacle-free periodic time-optimal racing problem that minimizes the lap time T as

$$
\begin{aligned}
\min_{\boldsymbol{x}(\cdot),\boldsymbol{u}(\cdot),T} \quad & T && (11a)\\
\text{s.t.} \quad & \boldsymbol{x}(0) = \boldsymbol{x}(T), && (11b)\\
& T \geq 0, && (11c)\\
& \dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad t \in [0,T], && (11d)\\
& \underline{\boldsymbol{u}} \leq \boldsymbol{u}(t) \leq \overline{\boldsymbol{u}}, \quad t \in [0,T], && (11e)\\
& \underline{n} \leq n(t) \leq \overline{n}, \quad t \in [0,T], && (11f)\\
& \underline{a}_\perp \leq a_\perp(\boldsymbol{x}(t)) \leq \overline{a}_\perp, \quad t \in [0,T], && (11g)\\
& \underline{a}_\parallel \leq a_\parallel(\boldsymbol{x}(t)) \leq \overline{a}_\parallel, \quad t \in [0,T], && (11h)
\end{aligned}
$$

where $\boldsymbol{x} = [s,\ n,\ \alpha,\ v]^T$ and $\boldsymbol{u} = [D,\ \delta]^T$ denote the race car's states and controls respectively and f is a model of the race car's dynamics. The controls are bounded by $\underline{\boldsymbol{u}}$ and $\overline{\boldsymbol{u}}$. The lateral displacement is limited by the left and right track boundaries that are described by $\underline{n}$ and $\overline{n}$, respectively. Additionally, we bound the lateral acceleration by $\underline{a}_\perp$ and $\overline{a}_\perp$ as well as the longitudinal acceleration by $\underline{a}_\parallel$ and $\overline{a}_\parallel$.

3.1 Progress-Maximization NMPC Formulation

In order to approximate time-optimal behavior online, we reformulate and discretiye the OCP stated in (11). We propose a quadratic objective function that tracks a reference $s_{k,\text{ref}}$ which is chosen to be slightly out of reach

$$
s_{k,\text{ref}} = s_0 + \frac{s_{N,\text{ref}}}{N} k, \qquad k = 0, \dots, N, \tag{12}
$$

where s_0 is the current track progress of the car, k indicates the step in the prediction horizon and N is the total number of steps. This formulation allows us to utilize efficient algorithms that use the generalized Gauss-Newton method. By maximizing the progress on the track we approximate time-optimal behavior. In each iteration, we then solve the following nonlinear program (NLP):

$$
\min_{\substack{\boldsymbol{x}_0,\dots,\boldsymbol{x}_N,\\ \boldsymbol{u}_0,\dots,\boldsymbol{u}_{N-1}}} \quad \sum_{k=0}^{N-1} \left\|\boldsymbol{x}_k - \boldsymbol{x}_{k,\text{ref}}\right\|_Q^2 + \|\boldsymbol{u}_k\|_R^2 + \left\|\boldsymbol{x}_N - \boldsymbol{x}_{N,\text{ref}}\right\|_{Q_\text{N}}^2 \tag{13a}
$$

$$
\begin{aligned}
\text{s.t.} \quad & \boldsymbol{x}_0 = \boldsymbol{x}_\text{c}, && (13b)\\
& \boldsymbol{x}_{k+1} = F(\boldsymbol{x}_k, \boldsymbol{u}_k, \Delta t), \quad k = 0,\dots,N-1, && (13c)\\
& \underline{\boldsymbol{u}} \leq \boldsymbol{u}_k \leq \overline{\boldsymbol{u}}, \quad k = 0,\dots,N-1, && (13d)\\
& \underline{a}_\perp \leq a_\perp(\boldsymbol{x}_k) \leq \overline{a}_\perp, \quad k = 0,\dots,N, && (13e)\\
& \underline{a}_\parallel \leq a_\parallel(\boldsymbol{x}_k) \leq \overline{a}_\parallel, \quad k = 0,\dots,N, && (13f)\\
& \underline{D} \leq D_k \leq \overline{D}, \quad k = 0,\dots,N, && (13g)\\
& \underline{\delta} \leq \delta_k \leq \overline{\delta}, \quad k = 0,\dots,N, && (13h)\\
& \underline{n}(s_k) \leq n_k \leq \overline{n}(s_k), \quad k = 0,\dots,N-1, && (13i)
\end{aligned}
$$

where $\boldsymbol{x}_\text{c}$ is the car's current state and $Q, R, Q_\text{N} \succ 0$ are positive definite weighting matrices. The constraints (13e) and (13i) are softened using slack variables that are tied to an L_1 penalty term in the cost function to ensure feasibility. To obtain smooth controls, we extend the state vector to $\boldsymbol{x} = [s,\ n,\ \alpha,\ v,\ D,\ \delta]^T$ and replace the controls by $\boldsymbol{u} = \left[\dot{D},\ \dot{\delta}\right]^T$ such that constraints (13g) and (13h) get state bounds.

In constraint (11f) the upper and lower bound $\underline{n}$ and $\overline{n}$ are constants. To account for obstacles placed on the track, we make these bounds depended on s_k (13i). Obstacles are thus modeled as a reduction of the track width. Note that we assume a supervisory control to allocate an obstacle to the left or right boundary. We implement $\underline{n}(s)$ and $\overline{n}(s)$ as piecewise cubic polynomials η_j with respect to the track progress s

$$
\eta_j(s) = p_{j,1}s^3 + p_{j,2}s^2 + p_{j,3}s + p_{j,4}, \quad j = 1,\dots,M, \tag{14}
$$

where j indicates the section of the piecewise polynomial, M the number of sections on the track and $p_{j,i}$ are the respective coefficients. We compute the coefficients based on four boundary conditions illustrated in Figure 5. Two boundary conditions correspond to the track width $\underline{w}_j$ and $\overline{w}_j$ at the beginning $\underline{s}$ and the end of the section $\overline{s}$. Those are adjusted according to the obstacles. Moreover, the first derivative $\eta_j'(\underline{s})$ is zero at the transition of the polynomials leading to a C^2 continuous inequality condition.

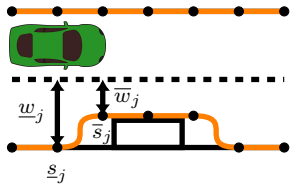

Fig. 5. Obstacles are described as a reduction of the track width. Therefore, the deformation of the track boundary is formulated by piecewise cubic polynomials.

4. RESULTS

In this section, we compare the results of the OCP formulation in (13), both in simulation and the real-world. We compare two scenarios, one related to obstacle-free racing and one to static obstacle avoidance including a stop-and-go maneuver.

4.1 Experimental Setup

The validation is conducted on the experimental setup used by Verschueren et al. (2014) which is located at the Department of Microsystems Engineering in Freiburg. The experiments were conducted using Kyosho dNaNo cars, which have rear wheel drive and use front wheel steering. They can achieve a forward velocity of more than 3 m/s. Illustrated in Figure 1, the vehicle starts at position $s = 0$ m and races clockwise. It first enters a chicane, followed by a U-turn and a longer straight section. The total length of the center line is 8.71 m.

The schematics of the experimental setup are presented in Figure 6. The position of the vehicle is detected with the RGB-camera XIMEA xiQ with 100 frames per second. The vehicles position and heading are found by image processing. Subsequently, the vehicles states $\boldsymbol{x}$ are estimated by a Kalman filter. Because the closed-loop system is affected by a considerably delay of approximately 80 ms, we compensate for it by forward prediction. The control signals $\boldsymbol{u}$ are sent to the vehicle via a 2.4 GHz remote control. The framework is embedded into the open-source software ROS (Quigley et al., 2009) and operates on a desktop computer with an Intel Xeon(R) E5-2687W at 3.1 GHz running Ubuntu 18.04.

4.2 Implementation

In order to solve the NLP in (13) online, we use the optimal control framework acados (Verschueren et al., 2019). We

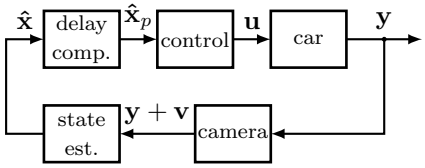

Fig. 6. Schematic of the race cars setup.

solve the NLP approximately using the well-known real-time iteration (RTI) scheme with an sequential quadratic programming (SQP) method (Diehl et al., 2002). The resulting quadratic programs (QPs) are solved by HPIPM which is based on BLASFEO (Frison et al., 2018). An efficient partial condensing method is combined with an explicit Runge-Kutta method of order 4.

We use a prediction horizon of 1 s and a sampling time of 20 ms. This results in 50 intervals with 100 controls and 300 states in total. Moreover, the piecewise polynomials in (13i) introduce 512 parameters. We choose the reference for the track progress s_N to be 3 m. The weights are selected as

$$Q_k = \mathrm{diag}\left([0.1, 1\mathrm{e}{-8}, 1\mathrm{e}{-8}, 1\mathrm{e}{-8}, 1\mathrm{e}{-3}, 5\mathrm{e}{-3}]\right) \quad (15a)$$

$$R_k = \mathrm{diag}\left([1\mathrm{e}{-3},\ 5\mathrm{e}{-3}]\right) \quad (15b)$$

$$Q_N = \mathrm{diag}\left([5,\ 100,\ 1\mathrm{e}{-8},\ 1\mathrm{e}{-8},\ 1\mathrm{e}{-3},\ 5\mathrm{e}{-3}]\right). \quad (15c)$$

4.3 Simulation

First, we present the simulation results in order to assess the theoretical limits for the real-world setup.

We consider two scenarios: an obstacle-free scenario and one with obstacles. The scenario with obstacles include a slalom parcour from 0.5 m to 3.5 m during the chicane and a temporary road blocking at the last straight line at 7.2 m which is removed when the vehicle comes to a full stop. In all scenarios, we consider a warm start of the vehicle.

The tracked path for both scenarios are presented in the left column of Figure 1. The time-optimal path from (11) in the obstacle-free scenario is illustrated by the black dotted line. The time-optimal and the progress-maximization solutions are almost identical with lap times of 4.85 s and 5.02 s, respectively. In the scenario with obstacles, the vehicle successfully drives through the slalom parcour. Thereby, it has to slow down compared to the obstacle-free scenario. It also comes to a full stop in front of the road blocking and continues as soon as the obstacle is removed.

The velocity v and the lateral acceleration $a_\perp$ of the vehicle for both scenarios are illustrated in Figure 7. The velocity profile of the obstacle-free scenario is similar to the optimal one. In the scenario with obstacles, the vehicle slows down during the chicane. Afterwards, the velocity converges to the obstacle-free scenario before coming to a full stop at $s = 7.2$ m. The lateral acceleration $a_\perp$ shows a bang-bang behavior within the given constraints. In the scenario with obstacles, the constraint is not active during the start-stop maneuver and before entering the slalom parcour.

The average computation time for the NMPC is 12 ms while the maximum during the test run is 15 ms such that the sampling time of 20 ms is never missed. The computation time does not depend on the chosen scenario

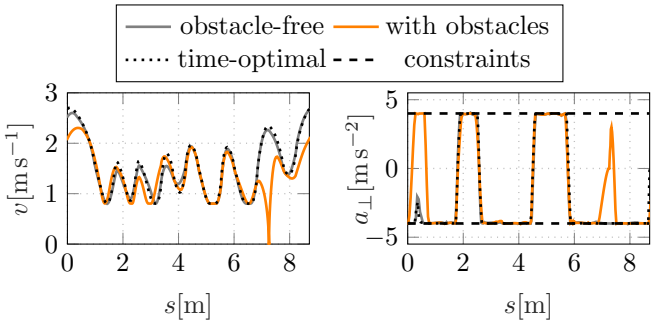

Fig. 7. Velocity and lateral acceleration profiles in simulation: The proposed approach has a velocity and lateral acceleration profile similar to the time-optimal solution for the obstacle-free scenario. In the scenario with obstacles, the vehicle slows down during the slalom parcour (0.5 m to 3 m) and comes to a full stop at 7.2 m.

as the OCP structure is does not change with added obstacles and we use the RTI scheme.

4.4 Real-World Experiments

In this section, we discuss the results on the race car setup and compare them to the solution achieved in simulation.

The driven paths for both scenarios are presented in the right column of Figure 1. For the obstacle-free scenario, the driven trajectory only slightly differs from the optimal one. The achieved lap time is 6.5 s which results in an average speed of 1.22 m/s. This is 1.91 s faster than the lap time of (Verschueren et al., 2014) and corresponds to a lap time reduction of 23.2 %. The largest offset from the optimal curve occurs at $s = 3$ m. In this case, the car approaches the curve too much on the inside. Moreover, an understeering behavior of the vehicle can be observed when the vehicle operates at the maximum lateral acceleration. This confirms the investigations in Section 2.3. In the scenario with obstacles, the vehicle successfully steers through the slalom parcour and also completes the start-stop maneuver. The most challenging situation is in the curve at the top right corner in Figure 1. In this case, the vehicle approaches the slalom at high speed and needs to conduct a S-curve. As the vehicle gets close to the right boundary, it nearly drives into an infeasible area.

Figure 8 presents a comparison of the velocity as well as the lateral acceleration profile both in the obstacle-free scenario (a) and the scenario with obstacles (b). In both cases, the average velocity is a bit slower than in simulation. Moreover, it can be seen that the lateral acceleration is not at its handling limits on the longer straight lines at $s \approx 3.5$ m and $s \approx 7$ m. Otherwise, the lateral acceleration also usually operates at its minimum or maximum in the real-world. In the scenario with obstacles, the vehicle comes to a full stop close the simulated position at $s = 7.2$ m.

5. CONCLUSION

In this paper, we illustrated how to implement a NMPC control with a tracking formulation that achieves near

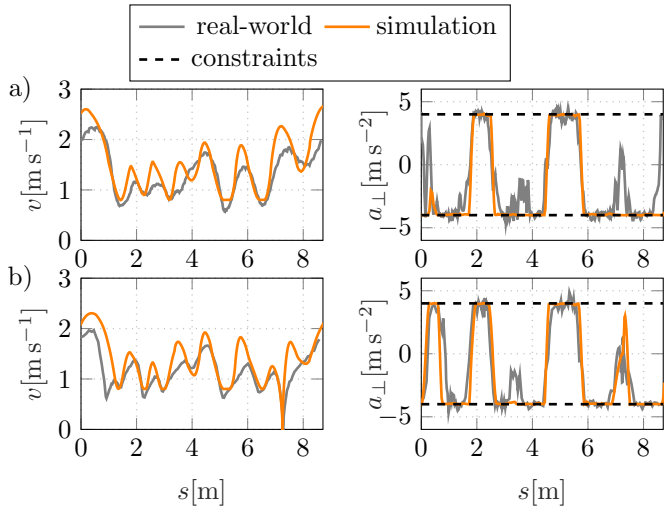

Fig. 8. Velocity and lateral acceleration comparison of simulation and real-world experiment for the obstacle-free scenario (a) and the scenario with obstacles (b). In general, the average speed on the real-world setup is a bit slower. In the scenario with obstacles, the vehicle comes to a full stop close to the simulated position. Similar to the simulation, the lateral acceleration $a_\perp$ also shows a bang-bang behavior in the real-world.

time-optimal results in simulation as well as in the real-world. We used a singularity-free formulation that can operate at low and high speeds and realize start-stop maneuvers. The singularity-free prediction model which only includes the forward dynamics enforces the model assumptions by constraints. The implementation can be found as part of the examples of acados on GitHub. Future research will focus on a singularity-free compensation for the model mismatch as the lateral acceleration increases in order to operate closer to its handling limits.

ACKNOWLEDGEMENTS

The authors thank Robin Verschueren for his advice and for setting up the race track. Further, the authors acknowledge the fruitful discussions with Alexander Liniger concerning the model identification.

REFERENCES

Diehl, M., Bock, H.G., Schlöder, J.P., Findeisen, R., Nagy, Z., and Allgöwer, F. (2002). Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4), 577–585.

Frasch, J.V., Gray, A.J., Zanon, M., Ferreau, H.J., Sager, S., Borrelli, F., and Diehl, M. (2013). An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles. In *ECC*, 4136–4141.

Frison, G., Kouzoupis, D., Sartor, T., Zanelli, A., and Diehl, M. (2018). BLASFEO: Basic linear algebra subroutines for embedded optimization. *ACM Transactions on Mathematical Software (TOMS)*, 44(4), 42:1–42:30.

Gao, Y., Lin, T., Borrelli, F., Tseng, E., and Hrovat, D. (2010). Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads. In *ASME 2010 dynamic systems and control conference*. American Society of Mechanical Engineers Digital Collection.

Hewing, L., Liniger, A., and Zeilinger, M. (2018). Cautious nmpc with gaussian process dynamics for autonomous miniature race cars. In *2018 European Control Conference (ECC)*. IEEE.

Kabzan, J., Valls, M., Reijgwart, V., et al. (2019). Amz driverless: The full autonomous racing system. *arXiv preprint: 1905.05150*.

Kong, J., Pfeiffer, M., Schildbach, G., and Borrelli, F. (2015). Kinematic and dynamic vehicle models for autonomous driving control design. *IEEE Intelligent Vehicles Symposium*.

Lima, P., Pereira, G.C., Martensson, J., and Wahlberg, B. (2018). Progress maximization model predictive controller. *International Conference on Intelligent Transportation Systems*.

Liniger, A., Domahidi, A., and Morari, M. (2015). Optimization-based autonomous racing of 1:43 scale RC cars. *Optimal Control Applications and Methods*, 36(5), 628–647.

Pacejka, H.B. (2006). *Tyre and Vehicle Dynamics*. Elsevier.

Perantoni, G. and Limebeer, D.J. (2014). Optimal control for a formula one car with variable parameters. *Vehicle System Dynamics*, 52(5).

Polack, P., Altche, F., d'Andrea Novel, B., and de La Fortelle, A. (2017). The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? *IEEE Intelligent Vehicles Symposium*.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A.Y. (2009). ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*.

Rosolia, U., Carvalho, A., and Borrelli, F. (2017). Autonomous racing using learning model predictive control. In *2017 American Control Conference (ACC)*.

Spengler, P. and Gammeter, C. (2010). *Modeling of 1:43 scale race cars*. Master's thesis, ETH Zürich.

Verschueren, R., Bruyne, S.D., Zanon, M., Frasch, J.V., and Diehl, M. (2014). Towards time-optimal race car driving using nonlinear MPC in real-time. In *CDC*, 2505–2510.

Verschueren, R., Frison, G., Kouzoupis, D., van Duijkeren, N., Zanelli, A., Novoselnik, B., Frey, J., Albin, T., Quirynen, R., and Diehl, M. (2019). acados: a modular open-source framework for fast embedded optimal control. *arXiv preprint: 1910.13753*.

Verschueren, R., Zanon, M., Quirynen, R., and Diehl, M. (2016). Time-optimal race car driving using an online exact hessian based nonlinear MPC algorithm. In *ECC*.