Lateral Vehicle Trajectory Optimization Using Constrained Linear Time-Varying MPC

Benjamin Gutjahr, Lutz Gröll, and Moritz Werling

***Abstract*— In this paper, a trajectory optimization algorithm is proposed, which formulates the lateral vehicle guidance task along a reference curve as a constrained optimal control problem. The optimization problem is solved by means of a linear time-varying model predictive control scheme that generates trajectories for path following under consideration of various time-varying system constraints in a receding horizon fashion. Formulating the system dynamics linearly in combination with a quadratic cost function has two great advantages. First, the system constraints can be set up not only to achieve collision avoidance with both static and dynamic obstacles, but also aspects of human driving behavior can be considered. Second, the optimization problem can be solved very efficiently, such that the algorithm can be run with little computational effort. In addition, due to an elaborate problem formulation, reference curves with discontinuous, high curvatures will be effortlessly smoothed out by the algorithm. This makes the proposed algorithm applicable to different traffic scenarios, such as parking or highway driving. Experimental results are presented for different real-world scenarios to demonstrate the algorithm's abilities.**

***Index Terms*— Advanced driver assistance systems, path tracking, trajectory optimization, linear time-varying MPC.**

I. Introduction

WITHIN the last few decades a strong research effort was made to improve comfort and safety in road traffic by vehicle automation. This has already led to great achievements in supporting the driver in various monotonous and challenging driving situations, see [27], [33]. In order to raise the level of vehicle automation towards highly- and fully-automated driving, current research focuses on trajectory generation algorithms that can cope with complex traffic featuring dynamic, time-critical street scenarios, such as merging into fast traffic flow, to pass opposing traffic, or to avoid other moving vehicles [38]. In addition to that, aspects of human driving behavior get more and more into focus (see e. g. [17], [26]). For this reason these planning algorithms must not only produce safe trajectories for a preferably wide range of driving scenarios, but also user-acceptable ones.

In literature many different methods for trajectory generation for automated vehicles exist. Under certain assumptions, fairly simple heuristics in combination with path-planning strategies were designed for isolated dynamic traffic scenarios [37]. When trying to integrate these heuristics into a universal concept, these approaches quickly reach their limits resulting in poor performance or even accidents [13]. Apart from these rule-based methods, potential field methods have been studied intensively (see e. g. [7], [23]). Due to their intuitive problem formulation, potential field methods have been used for several different vehicle applications. However, as shown in [22] there are still open issues as system dynamics and constraints are difficult to integrate within this approach. On the contrary, as shown in [1], [16], [31], and [38], optimization techniques have been applied successfully and therefore became the established method for trajectory generation over the last years. Within the field of optimization, there are three methods that can be distinguished. Throughout literature, many applications for vehicle automation can be found for all of these methods.

First, *Dynamic Programming* can be applied to a broad range of universal system models. Various different terms can be easily integrated into the cost function, as well as system constraints can be added flexibly. In addition to that, even non convex optimization problems can be solved globally. Unfortunately, due to the *curse of dimensionality*, [3], *Dynamic Programming* is very limited to small order system models. All these properties, on the one hand, make this method highly attractive for planning the future vehicle motion on a tactical level [16]. On the other hand, it is by far not possible to generate trajectories online that are sufficiently smooth to be used as system inputs for vehicle applications. For this reason, as suggested for instance in [18], it is favorable to combine *Dynamic Programming* with other optimization methods. For example, the rough reference curve for the parking scenario in the result section was generated by *Dynamic Programming* [34].

Second, although *Indirect Methods* can only be applied to local optimization problems, they show great potential with respect to trajectory generation in some special cases. In [19] it has been shown that for simple system models solutions to the optimization problem can be found extremely efficiently. In these very rare cases, an analytical solution can be pre-calculated offline. However, considering system constraints explicitly within the optimization is extremely difficult. Additionally, when solving the optimization problem numerically, initial conditions for the adjoint variables must be determined. This makes the applications of the *Indirect Methods* unflexible for online applications. Nevertheless, [38] shows how these methods can be applied successfully. In this

Manuscript received August 4, 2015; revised November 26, 2015, February 22, 2016, and August 29, 2016; accepted September 17, 2016. Date of publication October 19, 2016; date of current version May 29, 2017. The Associate Editor for this paper was B. De Schutter.

B. Gutjahr and M. Werling are with the BMW Group Research and Technology, 80992 Munich, Germany (e-mail: benjamin.gutjahr@bmw.de; moritz.werling@bmw.de).

L. Gröll is with the Institute for Applied Computer Science, Karlsruhe Institute of Technology, 76344 Karlsruhe, Germany (e-mail: lutz.groell@kit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2016.2614705

1524-9050 © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: Yuanfei Lin. Downloaded on August 12,2021 at 13:20:26 UTC from IEEE Xplore. Restrictions apply.

case, a simple system model is used in combination with a discretized terminal manifold and sampling, such that the constrained optimization problem can be solved at least in a sub-optimal way.

Third, to circumvent the difficulties of the indirect optimization for online applications, *Direct Methods* convert the optimal control problem (OCP) into nonlinear programming (NLP) that can be solved using numerical NLP solvers. Therefore, as there is no need for initial conditions for the adjoint variables, *Direct Methods* are very adequate for finding local solutions to nonlinear problems in real-time. Recent examples can be found in [14], [15], [39], and [41]. Apart from *Interior Point Methods*, sequential quadratic programming (SQP)-solvers are well-established for solving NLPs. The problem here is, that the solution of the optimal control problem is very much dependent on an initial starting guess for the numerical solver, which is not trivial to find. In addition to that, solving NLPs is very time-consuming, as these solvers brake down the NLP into smaller linear-quadratic programs (QP) that are solved iteratively using forward simulation of the system dynamics.

In order to reduce the computational effort, [10] for instance, suggests the formulation of linear-quadratic optimal control problems for trajectory generation. This way, no iterations and no initial starting guess are necessary such that only a single QP-problem needs to be solved at each optimization step. This makes the use of linear model predictive control (LMPC) highly favorable for real-time trajectory generation with high replanning frequencies. Therefore, applications can not only be found within the field of aerial vehicles [4], [24] or marine vessels [30], but also in the field of vehicle automation [2], [8], [29]. As following a given reference curve is one of the most fundamental tasks for vehicle automation, a strong effort was made to apply LMPC to the lateral vehicle guidance problem. For example, [11], [20], and [21] use LMPC for designing a lateral vehicle controller for tracking a given collision free evasive path. As generating a collision free evasive path is very complex for dynamic, time-critical street-scenarios, [9], [14] for instance, include the collision avoidance functionality into the reference tracking concept, by separating the lateral vehicle guidance task into a high-level trajectory generation layer for collision avoidance and a low-level control layer for stabilizing the vehicle, where LMPC can be applied for both layers.

Similar to our work, there are several works, using LMPC for trajectory optimization for lateral vehicle guidance. In [1], a trajectory planning algorithm is proposed for collision avoidance in hazard scenarios. In contrast to our approach, the optimal control problem is formulated only for straight roads and constant speeds, which significantly limits the application of this algorithm. Another approach is presented in [28], where the linear system is chosen such that the vehicle heading is used as a system input resulting in kinodynamically infeasible trajectories. In [35], a problem formulation is presented that allows for path tracking of low curvature roads, which again makes the algorithm only applicable for limited use-cases. However, this work features several aspects that were used as an inspiration for the work presented in this paper.

Apart from the method for trajectory optimization, current research also focuses on generating trajectories that are similar to human driving behavior [36]. This driving behavior is characterized by a very smooth and continuous steering action minimizing the lateral vehicle jerk and limiting the lateral vehicle acceleration dependent on the current curvature value [17]. Therefore, as indicated in [25] or [26], it is favorable to allow offsets to a given reference under consideration of the vehicles environment in order to increase driving comfort. Consequently, we account for this compromise by choosing an appropriate problem formulation in combination with a suitable cost function in the optimization problem.

Based on the idea of using LMPC, the main contribution of this work is an elaborate formulation of a constrained linear-quadratic optimal control problem for the lateral vehicle guidance task that shows essential differences to comparable works. Due to the proposed problem formulation, solving the optimization problem for a given predicted velocity profile of the future vehicle motion, this algorithm is capable of generating smooth and collision free trajectories among static obstacles and moving traffic for the vehicle's lateral movement even along reference curves that feature high and discontinuous curvature values and can run on a low-performance electronic control unit in milliseconds. This all together makes the algorithm applicable for a broad range of automated vehicle applications. Following the overall MPC scheme presented in [6], this paper is organized equivalently. Therefore, in Sec. II a linear time-varying system model is formulated approximating the vehicle movement with respect to the reference curve. In order to achieve collision avoidance both with static and dynamic obstacles, and to obtain kinematically feasible system trajectories, time-varying system constraints are defined in Sec. III. In the following Sec. IV, a thoroughly chosen cost function is stated such that aspects of a human driving behavior can be accounted for. This leads to an optimal control problem which is solved by means of a time-varying LMPC in a receding horizon fashion in Sec. V. After a brief description of the algorithms implementation and testing setup, experimental results are presented in Sec. VI. Finally, a short summary and an outlook are given in the concluding Sec. VII.

II. Vehicle Prediction Model

The essence of MPC is to optimize a forecast of the system behavior. This is accomplished by a system model, making it the most important element of a model predictive controller [32]. Hence, in order to generate optimized trajectories for the future vehicle motion, a prediction model is essential. These models differ not only in terms of the considered physical aspects, but also by the choice of the system inputs. As we aim for a universal trajectory generation concept independent of specific vehicle parameters, we only consider the vehicle's kinematic, choosing the vehicle's rear axle center as a reference point and the curvature's time derivative as a system input. This, on the one hand, leads to smooth steering actions even for discontinuous input sequences as the system input is defined as a high order system state. On the other hand, as the slip angle can be assumed to be zero at the

Authorized licensed use limited to: Yuanfei Lin. Downloaded on August 12,2021 at 13:20:26 UTC from IEEE Xplore. Restrictions apply.

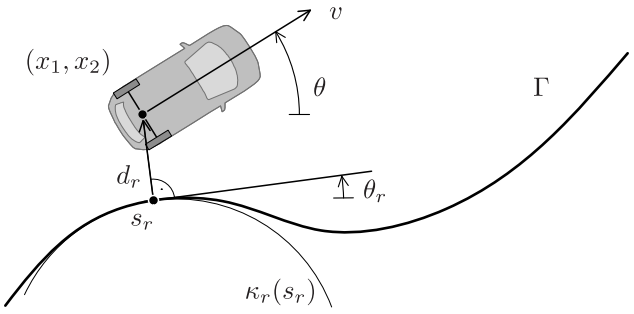

Fig. 1. Kinematic vehicle model with respect to a given reference curve Γ.

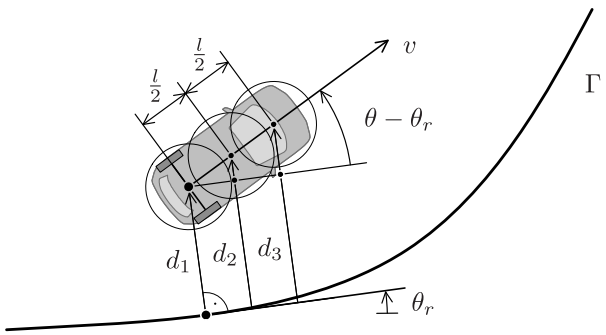

Fig. 2. Visualization of the defined system outputs.

rear axle center when driving slowly and generally neglected when driving fast, the kinematic vehicle model is adequate for trajectory generation in both low-speed and high-speed use-cases. The classical kinematic vehicle model formulates the vehicle dynamics in reference to an initial coordinate system. However, as outlined in [38], it is favorable to plan the future vehicle motion locally along a given reference curve Γ. For this reason, we modify the classical model by introducing the new system state d_r as the normal distance between the reference curve Γ and the vehicle's rear axle center position x_1, x_2. In addition to that, θ_r and κ_r represent the reference curve's orientation and curvature at the correspondent base point, which is defined by the curve's arc length variable s_r. Equivalently, θ and κ denote the vehicle's orientation and its trajectory's curvature. These geometric relations are pictured in Fig. 1 and lead to the following nonlinear system model,

$$\dot{d}_r = v(t)\sin(\theta - \theta_r) \tag{1a}$$
$$\dot{\theta} = v(t)\kappa \tag{1b}$$
$$\dot{\kappa} = u \tag{1c}$$
$$\dot{\theta}_r = \underbrace{v(t)\tfrac{\cos(\theta-\theta_r)}{1-d_r\kappa_r}}_{v_r}\kappa_r \tag{1d}$$
$$\dot{\kappa}_r = z \tag{1e}$$

with $v(t)$ stating the vehicle's velocity as a time varying parameter and the curvature's derivative being the system input u. Furthermore, v_r denotes the projected velocity on the reference curve and z is a given external signal derived from Γ.

A. Linear System Model

When having a closer look at the system model (1), it is interesting to see, that when interpreting $v(t)$ as a given time-varying system parameter the only nonlinearities arise from the trigonometric function in (1a) and the relation between the vehicle speed $v(t)$ and the projected reference speed v_r in (1d). As the vehicle is supposed to move along the reference curve closely, the difference in orientation $\theta - \theta_r$ is typically smaller than a 20° angle at all times [35] and d_r is small with respect to the reference curve's radius r_r which is defined by $r_r = \frac{1}{\kappa_r}$. Therefore on the one hand, approximating the trigonometric functions by $\sin(\alpha) \approx \alpha$ and $\cos(\alpha) \approx 1$ is a valid assumption. On the other hand, we can assume $\frac{d_r}{r_r} = d_r\kappa_r \ll 1$, such that the difference between $v(t)$ and v_r in (1d), can be neglected even for sharp turns, leading to the following linear time-variant system model,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_C(t)\boldsymbol{x}(t) + \boldsymbol{B}_C(t)u(t) + \boldsymbol{E}_C(t)z(t),\ \boldsymbol{x}(t_j) = \boldsymbol{x}_0 \tag{2}$$

where

$$\boldsymbol{A}_C(t) = \begin{bmatrix} 0 & v(t) & 0 & -v(t) & 0 \\ 0 & 0 & v(t) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & v(t) \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$
$$\boldsymbol{B}_C(t) = \begin{bmatrix} 0, 0, 1, 0, 0 \end{bmatrix}^{\mathrm{T}}, \quad \boldsymbol{E}_C(t) = \begin{bmatrix} 0, 0, 0, 0, 1 \end{bmatrix}^{\mathrm{T}},$$

with $v(t)$ given by a desired future velocity profile and $\boldsymbol{x}^{\mathrm{T}} = [d_r, \theta, \kappa, \theta_r, \kappa_r]$ being the system's state vector.

B. Definition of the System Output

In preparation for an efficient formulation of system constraints in the following section, we define a number of system outputs. Based on the idea of approximating the vehicle shape by several circles for fast collision checking, as for instance suggested in [40], we locate three circle center points along the vehicle center line such that the vehicle shape is covered appropriately, as depicted in Fig. 2. In our specific case, we choose the first and the third circle center point to coincide with the vehicle rear axle and front axle center point, respectively. The second circle's position is selected by the distance $l/2$, in which l denotes the vehicle wheelbase. Dependent on these distant values ($l_1 = 0, l_2 = \frac{l}{2}, l_3 = l$), we define the circles' center positions with respect to the reference curve Γ as one component of the system output vector which is given by

$$d_i = d_r + l_i\sin(\theta - \theta_r) \approx d + l_i(\theta - \theta_r), \quad i = 1, 2, 3. \tag{3}$$

These relations are also illustrated in Fig. 2. Furthermore, the vehicle curvature κ is defined as an additional system output to account for physical limitations in the following section. This leads to the following compact formulation of the system output vector

$$\boldsymbol{y} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \kappa \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & \frac{1}{2}l & 0 & -\frac{1}{2}l & 0 \\ 1 & l & 0 & -l & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}}_{\boldsymbol{C}_C} \begin{bmatrix} d_r \\ \theta \\ \kappa \\ \theta_r \\ \kappa_r \end{bmatrix}. \tag{4}$$

Authorized licensed use limited to: Yuanfei Lin. Downloaded on August 12,2021 at 13:20:26 UTC from IEEE Xplore. Restrictions apply.

C. Discrete-Time Vehicle Prediction Model

In order to achieve a suitable problem formulation for efficient calculations, we transform the continuous-time vehicle prediction model (2) and (4) into its equivalent time-discrete form. Assuming the system matrix $\boldsymbol{A}_C(t)$ as well as the system input $u(t)$ and the system disturbance $z(t)$ to be constant within each discretization interval k, the system's state-transition matrix $\boldsymbol{\phi}_k(T_s) = e^{\boldsymbol{A}_C T_s} \circ\!\!-\!\!\bullet\, [s\boldsymbol{I} - \boldsymbol{A}_C]^{-1}$ is given by the Laplace transformation for each interval, with T_s being the discretization step size. This leads to the following time-discrete system matrices for each discretization interval

$$\boldsymbol{A}(k) = \boldsymbol{\phi}_k(T_s), \;\; \boldsymbol{B}(k) = \int_0^{T_s} \boldsymbol{\phi}_k(\tau)\boldsymbol{B}_C \, \mathrm{d}\tau. \tag{5}$$

Please note, that $\boldsymbol{E}(k)$ is derived equivalently to $\boldsymbol{B}(k)$ and $\boldsymbol{C}(k) = \boldsymbol{C}_C$. With these matrices (5) in hand, the time-discrete vehicle prediction model is given by

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}(k)\boldsymbol{x}(k) + \boldsymbol{B}(k)u(k) + \boldsymbol{E}(k)z(k), \\ \boldsymbol{y}(k) &= \boldsymbol{C}(k)\boldsymbol{x}(k), \quad \boldsymbol{x}(k=0) = \boldsymbol{x}(t_j) = \boldsymbol{x}_0. \end{aligned} \tag{6}$$

As we want to minimize the discretization error caused by the system disturbance $z(k)$ and to achieve $z(k)$ to be constant within each discretization interval we define

$$z(k) = \frac{\kappa_r(k+1) - \kappa_r(k)}{T_s} \tag{7}$$

approximating the reference curvature as a forward first order hold.

III. System Constraints

When generating trajectories in real-world road traffic scenarios, other traffic participants or obstacles have a huge influence on the future vehicle motion. Obviously, if two objects move towards the same place at the same time, this necessarily leads to a collision. For this reason, as indicated in Fig. 3, based on the predicted movement of other traffic participants (rectangular box) and the location of static obstacles (arbitrarily dashed lines), we constrain the future vehicle motion, such that collision avoidance can be achieved.

As pictured in Fig. 2, the system output (3) can be interpreted as a normal distance between the reference curve Γ and the center position of each approximating circle. Thus, following this interpretation, and considering the vehicle environment, an upper and a lower bound for these distance values $d_i, \; i = 1, 2, 3$ can be determined limiting the vehicle's future lateral deviation from the reference curve Γ. These limitations are shown in Fig. 3, where the approximating circles are moved to both sides of the vehicle indicating their maximum admissible deviation on the left $d_{i,\max}$ and on the right $d_{i,\min}$ for two consecutive time instants t_j and t_Δ. The second time instant t_Δ illustrates the predicted constellation of the moving object $(x(t_\Delta), y(t_\Delta))_{\text{obj}}$ and the ego-vehicle position $s_r(t_\Delta)$ along the reference curve Γ after the prediction interval Δt, in which $t_\Delta = t_j + \Delta t$. This constellation demonstrates, that due to the presence of moving traffic (rectangular box) the maximum admissible deviation $d_{i,\max}$ is not only dependent on the vehicle's position s_r along the reference curve Γ, but also on time t. Therefore, the system constraints must be formulated in a time-varying fashion such that $d_{i,\min}(s_r(t), t)$ and $d_{i,\max}(s_r(t), t)$ for $i = 1, 2, 3$, where $s_r(t)$ is given by the integration of the desired future velocity profile.

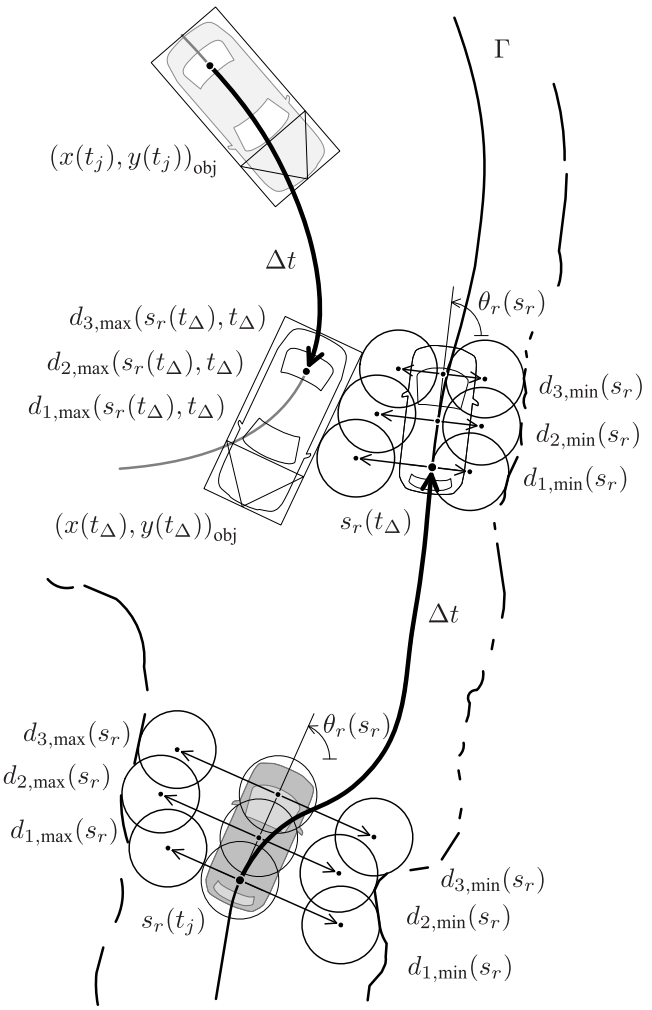

Fig. 3. Representation of the time-varying system constraints for collision avoidance for the current ego position $s_r(t_j)$ and a single arbitrarily chosen future time instance t_Δ. The ego-vehicle is pictured in dark gray, whereas other traffic is colored in light gray and approximated by a rectangular box. The ego-vehicle outline marks the predicted vehicle position on the reference curve Γ after the time interval Δt, in which $t_\Delta = t_j + \Delta t$. The predicted position of the other traffic is marked similarly.

With this illustration in mind we are now able to define linear time-varying constraints for the system output given by (3). Taking the time-discrete vehicle prediction model (6) into account, collision avoidance can be achieved if

$$d_{i,\min}(s_r(k), k) \leq d_i(k) \leq d_{i,\max}(s_r(k), k), \quad i = 1, 2, 3 \tag{8}$$

holds for each future discrete time instant k. Please note, that for a lane keeping use-case, not only other obstacles but also the lane markings can be used for constraining the vehicle's lateral motion.

In addition to these constraints for collision avoidance, we also constrain the vehicle's lateral dynamics, such that kinematically feasible system trajectories can be generated.

Authorized licensed use limited to: Yuanfei Lin. Downloaded on August 12,2021 at 13:20:26 UTC from IEEE Xplore. Restrictions apply.

This restriction arises from the physical limitations of the vehicle's steering actuator and wheel traction. As the steering actuator can only supply a certain maximum power, the vehicle's steering rate is limited. Considering the vehicle's general dynamics and the prediction model (6), this limitation can be accounted for by constraining the curvature's rate of change by box constraints on the system input:

$$u_{\min} \leq u(k) \leq u_{\max}. \tag{9}$$

Apart from that, the vehicle's steering angle δ is restricted, too. Therefore, similarly to (9), a fixed time-invariant upper $\kappa_{\max,\delta}$ and lower $\kappa_{\min,\delta}$ bound on the system output κ (see (4)) is defined. Additionally, in order to account for the limited wheel traction, we extend these bounds by a time-varying term which is dependent on the desired future velocity profile $v(k)$ and the estimated friction parameter μ. The combination of both constrains can be expressed as

$$\underbrace{\max\left(\kappa_{\min,\delta}, \kappa_{\min,\mu}(k)\right)}_{\kappa_{\min}} \leq \kappa(k) \leq \min\underbrace{\left(\kappa_{\max,\delta}, \kappa_{\max,\mu}(k)\right)}_{\kappa_{\max}},$$

where $\kappa_{\min,\mu}(k) = \kappa_{\min}(v(k), \mu)$ and $\kappa_{\max,\mu}(k) = \kappa_{\max}(v(k), \mu)$ are derived using the relations stated by the so-called *Kamm's circle*.

IV. Cost Function

Following the basic idea of LMPC [32], the optimization objective is formulated as the minimization of a quadratic running cost function $l(\boldsymbol{x}(k), \boldsymbol{u}(k))$. According to our goal of generating user-acceptable system trajectories, this cost function is motivated by human driving behavior. It is characterized by a trade-off between lane keeping and the minimization of the vehicle's lateral acceleration and jerk. In other words, human drivers deviate from the lane center if this leads to a smooth and continuous steering action. Therefore, based on the states of the vehicle prediction model (6), we propose the running cost function as

$$l(\boldsymbol{x}(k), u(k)) = w_d d_r^2 + w_\theta[\theta - \theta_r]^2 + w_\kappa \kappa^2 + w_u u^2 \tag{10}$$

with $w_d, w_\theta, w_\kappa, w_u > 0$. Here, the first two terms penalize deviations from the reference curve Γ to account for a path tracking behavior. Moreover, the second term is chosen to ensure that the linear approximation in Sec. II-A holds. Furthermore, the last two terms restrain the vehicle's lateral dynamics by penalizing any curvature values and change rates resulting in deviations from the reference when turning. With these opposing objectives we account for the trade-off mentioned above, where the function's weighting factors w_d, w_θ, w_κ can be parametrized specifically for different use-cases. As these use-cases mainly differ by speed, it is advantageous to define these parameters dependent on the desired future velocity, such that $w_{x_i}(v(k))$, $x_i = d, \theta, \kappa, u$. Although tuning these parameters is not a secondary task with respect to a stable closed-loop behavior, an adequate parametrization was determined by test-drives at different speeds where w_κ is continuously increased over speed while w_d is degraded. With these time-varying parameters, the running cost function (10) can be expressed in its general matrix form as

$$l(\boldsymbol{x}(k), u(k)) = \boldsymbol{x}(k)^{\mathrm{T}} \boldsymbol{Q}(k) \boldsymbol{x}(k) + u(k) \boldsymbol{R}(k) u(k), \tag{11}$$

where

$$\boldsymbol{Q}(k) = \begin{bmatrix} w_d(k) & 0 & 0 & 0 & 0 \\ 0 & w_\theta(k) & 0 & -w_\theta(k) & 0 \\ 0 & 0 & w_\kappa(k) & 0 & 0 \\ 0 & -w_\theta(k) & 0 & w_\theta(k) & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and $\boldsymbol{R}(k) = [w_u(k)]$, with $\boldsymbol{Q}(k)$ being a *positive-semidefinite* matrix and $\boldsymbol{R}(k)$ being *positive-definite* one for all times.

V. Constrained Optimal Control Problem

With the definition of the linear prediction model (6) in Sec. II, the linear system constrains in Sec. III, and the quadratic running cost function (11) a constrained linear quadratic optimal control problem is formulated over a prediction horizon N, with $k = 0, \ldots, N$, in this section. This is done by means of the so-called *batch-approach* as introduced in [6]. By doing so, the resulting formulation allows us to solve the optimization problem very efficiently using well-established quadratic programming solvers (QP-solvers). Solving the optimal control problem for the system input u in a receding horizon fashion, a linear model-predictive controller is established.

According to this approach, for each point in time t_j when the optimization problem is solved, the future system behavior is formulated in terms of a given initial state vector $\boldsymbol{x}_0$, a given disturbance-vector sequence $\mathbf{z}$, and a unknown input-vector sequence $\mathbf{u}$, that is

$$\mathbf{u} = \left[u_0, u_1, \ldots, u_{N-1}\right]^{\mathrm{T}}, \quad \mathbf{u} \in \mathbb{R}^N, \tag{12}$$

$$\mathbf{z} = \left[z_0, z_1, \ldots, z_{N-1}\right]^{\mathrm{T}}, \quad \mathbf{z} \in \mathbb{R}^N. \tag{13}$$

Note, that for the sake of a compact representation the optimization step time k is now represented as an index. Based on these vector sequences, $\boldsymbol{x}_0$, and the time-varying prediction model (6), the future state and output vector sequences

$$\mathbf{x} = \left[\boldsymbol{x}_1^{\mathrm{T}}, \ldots, \boldsymbol{x}_N^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \mathbf{x} \in \mathbb{R}^{nN} \tag{14}$$

$$\mathbf{y} = \left[\boldsymbol{y}_1^{\mathrm{T}}, \ldots, \boldsymbol{y}_N^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \mathbf{y} \in \mathbb{R}^{pN} \tag{15}$$

with $n = 5$ and $p = 4$ being the number of system states and outputs, are defined by

$$\mathbf{x} = \boldsymbol{\mathcal{A}} \boldsymbol{x}_0 + \boldsymbol{\mathcal{B}} \mathbf{u} + \boldsymbol{\mathcal{E}} \mathbf{z} \tag{16}$$

$$\mathbf{y} = \boldsymbol{\mathcal{C}} \mathbf{x} \tag{17}$$

where

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} (\boldsymbol{A}_0)^{\mathrm{T}} & \left(\prod_{q=0}^{1} \boldsymbol{A}_{1-q}\right)^{\mathrm{T}} & \cdots & \left(\prod_{q=0}^{N-1} \boldsymbol{A}_{N-1-q}\right)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}},$$

$$\boldsymbol{\mathcal{B}} = \begin{bmatrix} \boldsymbol{B}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \boldsymbol{A}_1 \boldsymbol{B}_0 & \boldsymbol{B}_1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \left(\prod_{q=1}^{N-1} \boldsymbol{A}_{N+0-q}\right) \boldsymbol{B}_0 & \cdots & \boldsymbol{A}_{N-1} \boldsymbol{B}_{N-2} & \boldsymbol{B}_{N-1} \end{bmatrix},$$

and $\boldsymbol{\mathcal{C}} = \mathrm{diag}(\boldsymbol{C}_1, \ldots, \boldsymbol{C}_N)$. Furthermore, as $\boldsymbol{\mathcal{E}}$ is derived similarly to $\boldsymbol{\mathcal{B}}$, its specific definition is not stated here.

Authorized licensed use limited to: Yuanfei Lin. Downloaded on August 12,2021 at 13:20:26 UTC from IEEE Xplore. Restrictions apply.

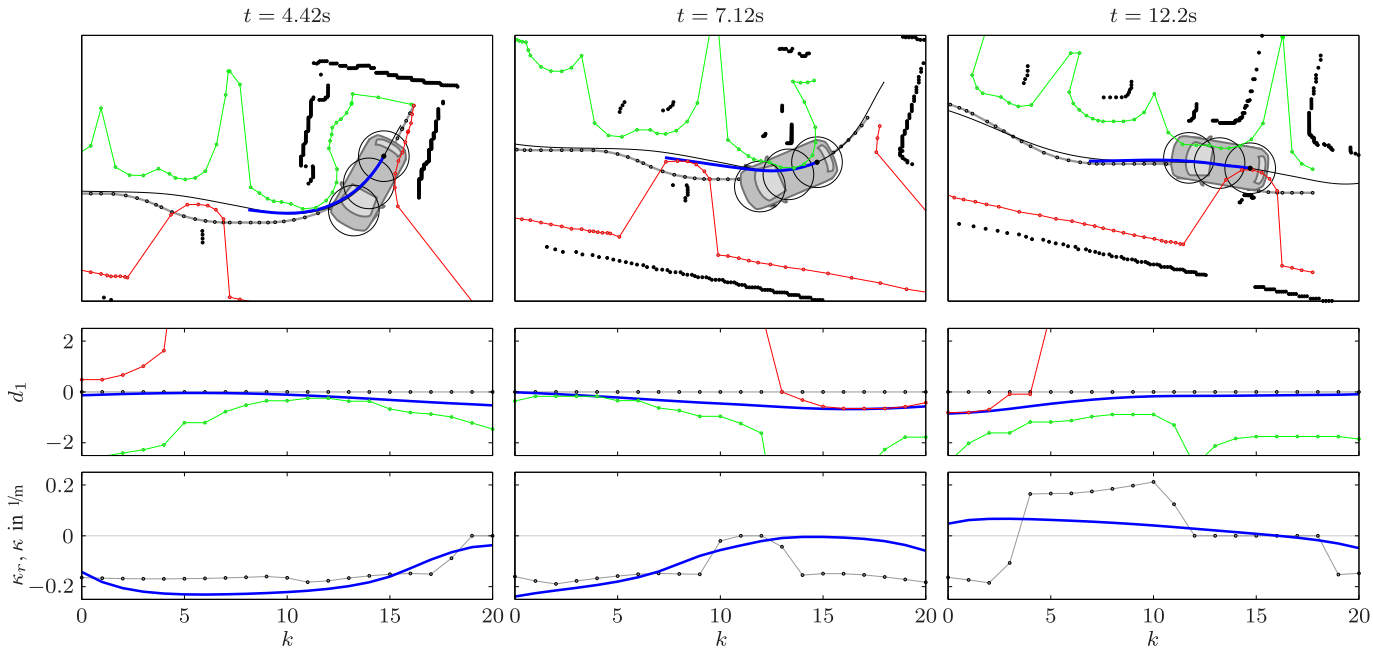

Fig. 4. Bird's eye view of the moving vehicle for the parking scenario at three consecutive instants of time with its correspondent results for the lateral position for the rear axle d_1 and the curvature κ, where the blue bold line represents the optimized trajectory.

Equivalently, the running cost function (11) can be summed up for all k, stating the optimization's cost function over the entire prediction horizon N. Given the state vector sequence $\mathbf{x}$ and the input vector sequence $\mathbf{u}$ the cost function can be expressed as

$$\begin{aligned} J(\mathbf{x},\mathbf{u}) &= \boldsymbol{x}_N^{\mathrm{T}} \boldsymbol{P} \boldsymbol{x}_N + \sum_{k=1}^{N-1} \boldsymbol{x}_k^{\mathrm{T}} \boldsymbol{Q}_k \boldsymbol{x}_k + \sum_{k=0}^{N-1} \boldsymbol{u}_k^{\mathrm{T}} \boldsymbol{R}_k \boldsymbol{u}_k, \\ &= \mathbf{x}^{\mathrm{T}} \boldsymbol{\mathcal{Q}} \mathbf{x} + \mathbf{u}^{\mathrm{T}} \boldsymbol{\mathcal{R}} \mathbf{u}, \end{aligned} \tag{18}$$

where $\boldsymbol{\mathcal{R}} = \mathrm{diag}(\boldsymbol{R}_0, \ldots, \boldsymbol{R}_{N-1})$ and $\boldsymbol{\mathcal{Q}} = \mathrm{diag}(\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_{N-1}, \boldsymbol{P})$,[1] with $\boldsymbol{P} = \boldsymbol{Q}_N$ emphasizing the terminal cost term which is chosen specifically in order to meet stability issues of the closed loop system [6].

As we want to solve the optimization problem for the input vector sequence $\mathbf{u}$, on the one hand, we substitute the state vector sequence $\mathbf{x}$ in (18) using (17) which yields to

$$\begin{aligned} J(\boldsymbol{x}_0, \mathbf{z}, \mathbf{u}) &= \mathbf{x}^{\mathrm{T}} \boldsymbol{\mathcal{Q}} \mathbf{x} + \mathbf{u}^{\mathrm{T}} \boldsymbol{\mathcal{R}} \mathbf{u} \\ &= \mathbf{u}^{\mathrm{T}} \boldsymbol{H} \mathbf{u} + 2 \left(\boldsymbol{x}_0^{\mathrm{T}} \boldsymbol{F} + \mathbf{z}^{T} \boldsymbol{G} \right) \mathbf{u} + \boldsymbol{O} \end{aligned} \tag{19}$$

where

$$\boldsymbol{H} = \boldsymbol{\mathcal{B}}^{\mathrm{T}} \boldsymbol{\mathcal{Q}} \boldsymbol{\mathcal{B}} + \boldsymbol{\mathcal{R}}, \;\; \boldsymbol{F} = \boldsymbol{\mathcal{A}}^{\mathrm{T}} \boldsymbol{\mathcal{Q}} \boldsymbol{\mathcal{B}}, \;\; \boldsymbol{G} = \boldsymbol{\mathcal{E}}^{\mathrm{T}} \boldsymbol{\mathcal{Q}} \boldsymbol{\mathcal{B}}$$

with $J(\boldsymbol{x}_0, \mathbf{z}, \mathbf{u})$ being only dependent on a variable $\mathbf{u}$ and $\boldsymbol{O}$ representing a constant additional cost. On the other hand, in preparation for a suitable formulation of the system constraints we define the system output in reference to the input vector sequence $\mathbf{u}$ using the same substitution by

$$\mathbf{y} = \boldsymbol{\mathcal{C}}[\boldsymbol{\mathcal{A}} \boldsymbol{x}_0 + \boldsymbol{\mathcal{B}} \mathbf{u} + \boldsymbol{\mathcal{E}} \mathbf{z}]. \tag{20}$$

[1] As the cost term $\boldsymbol{x}_0^{\mathrm{T}} Q_0 \boldsymbol{x}_0$ cannot be influenced by the system input $\mathbf{u}$, this term is not included in the cost function $J(\boldsymbol{x}_0, \mathbf{z}, \mathbf{u})$.

In order to avoid infeasibility issues caused by the imposed constraints, we introduce so-called *slack*-variables $\boldsymbol{\epsilon} \geq 0$ according to [6]. By including these variables into the optimization problem as additional optimization variables, we penalize the violation of the constraints in the cost function (19). This essentially leads to a *softening* of the constraints' impact. Unlike comparable works (such as [1] and [28]), we only soften constraints for selected outputs $y_{i_s}(k)$, $\forall i_s \in \mathcal{I}_s, \mathcal{I}_s \subseteq [1, \ldots, p]$ at all time steps $k \in \mathcal{K}_s$, $\mathcal{K}_s \subseteq [1, \ldots, N]$ over the prediction horizon N. This is motivated by our path-following application. In contrast to the limitations on κ, which are fairly constant, the lateral obstacle position may suddenly change due to sensor noise as the vehicle moves forward. As this situation affects the feasibility of the optimization problem only shortly before an obstacle is passed, we only soften the collision avoidance constraints y_{i_s}, $\forall i_s \in \mathcal{I}_s = [1, 2, 3]$ over a short prediction horizon N_S with $N_S < N$. Using the same *slack*-variable for the system outputs $y_{i_s}(k)$, with $k \in \mathcal{K}_s = [1, \ldots, N_S]$, but for both vehicle sides independently, we define a total number of $r = 2$ *slack*-variables, namely $\boldsymbol{\epsilon} = [\epsilon_u, \epsilon_l]$ for a number of $p_s = |\mathcal{I}_s| N_S$ soft output equations, in which $pN = p_s + p_h$. Therefore, we modify the cost function (19)

$$\begin{aligned} \tilde{J}(\boldsymbol{x}_0, \mathbf{z}, \mathbf{u}, \boldsymbol{\epsilon}) &= J(\boldsymbol{x}_0, \mathbf{z}, \mathbf{u}) + \boldsymbol{k}_1^{\mathrm{T}} \boldsymbol{\epsilon} + \boldsymbol{\epsilon}^{\mathrm{T}} \boldsymbol{K}_2 \boldsymbol{\epsilon} \\ &= \tilde{\mathbf{u}}^{\mathrm{T}} \tilde{\boldsymbol{H}} \tilde{\mathbf{u}} + \tilde{\boldsymbol{F}} \tilde{\mathbf{u}} \end{aligned} \tag{21}$$

with $\boldsymbol{k}_1 = [k_{11}, k_{12}]^{\mathrm{T}}$ and $\boldsymbol{K}_2 = \mathrm{diag}(k_{21}, k_{22})$ being weighting factors as well as $\tilde{\boldsymbol{H}} = \mathrm{diag}(\boldsymbol{H}, \boldsymbol{K}_2)$, $\tilde{\boldsymbol{F}} = \left[2 \left(\boldsymbol{x}_0^{\mathrm{T}} \boldsymbol{F} + \mathbf{z}^{\mathrm{T}} \boldsymbol{G} \right) \boldsymbol{k}_1^{\mathrm{T}} \right]$ and $\tilde{\mathbf{u}} = \left[[\mathbf{u}^{\mathrm{T}}, [\epsilon_u, \epsilon_l]^{\mathrm{T}}] \right]^{\mathrm{T}}$. Based on the definition of $y_{i_s}(k)$, we also modify the output vector sequence (20) by defining two separate vector sequences for

Authorized licensed use limited to: Yuanfei Lin. Downloaded on August 12,2021 at 13:20:26 UTC from IEEE Xplore. Restrictions apply.

Fig. 5. Prototype vehicle during a fully automated parking maneuver.

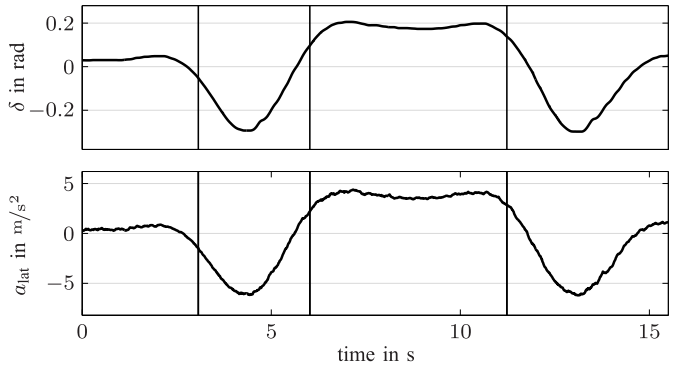

Fig. 6. Measurement signals for the steering angle δ and the lateral acceleration a_{lat} of the closed-loop system for the roundabout maneuver.

the *soft* outputs $\mathbf{y}_s$ and the *hard* outputs $\mathbf{y}_h$ as

$$\begin{aligned}\mathbf{y}_h &= \boldsymbol{\mathcal{C}}_h[\boldsymbol{\mathcal{A}}\boldsymbol{x}_0 + \boldsymbol{\mathcal{B}}\mathbf{u} + \boldsymbol{\mathcal{E}}\mathbf{z}],\ \boldsymbol{\mathcal{C}}_h = \text{diag}(\boldsymbol{C}_{h,1}, \dots \boldsymbol{C}_{h,N}),\\ \mathbf{y}_s &= \boldsymbol{\mathcal{C}}_s[\boldsymbol{\mathcal{A}}\boldsymbol{x}_0 + \boldsymbol{\mathcal{B}}\mathbf{u} + \boldsymbol{\mathcal{E}}\mathbf{z}],\ \boldsymbol{\mathcal{C}}_s = \text{diag}(\boldsymbol{C}_{s,1}, \dots \boldsymbol{C}_{s,N_S})\end{aligned} \tag{22}$$

with $\boldsymbol{C}_{h,\boldsymbol{k}} = [0,0,0,1]\ \boldsymbol{C}_{\boldsymbol{k}}$ and $\boldsymbol{C}_{s,\boldsymbol{k}} = \boldsymbol{I}_{|\mathcal{I}_s|\times p}\ \boldsymbol{C}_{\boldsymbol{k}}\ \forall k \in \mathcal{K}_s$, but $\boldsymbol{C}_{h,k} = \boldsymbol{C}_k\ \forall k \notin \mathcal{K}_s$, where $\boldsymbol{I}_{|\mathcal{I}_s|\times p}$ denotes the upper $|\mathcal{I}_s|$ rows of $\boldsymbol{I}_p$. This way, given the output constraints in accordance with the output vector sequences $\mathbf{y}_h$ and $\mathbf{y}_s$

$$\begin{aligned}\mathbf{y}_{h,\text{min/max}} &= \left[\boldsymbol{y}_{h,1,\text{min/max}}^{\mathrm{T}}, \dots, \boldsymbol{y}_{h,N,\text{min/max}}^{\mathrm{T}}\right]^{\mathrm{T}},\\ \mathbf{y}_{s,\text{min/max}} &= \left[\boldsymbol{y}_{s,1,\text{min/max}}^{\mathrm{T}}, \dots, \boldsymbol{y}_{s,N_s,\text{min/max}}^{\mathrm{T}}\right]^{\mathrm{T}},\end{aligned}$$

where similarly for the upper bound $\boldsymbol{y}_{h,k,\text{min}}^{\mathrm{T}} = [\kappa_{\min}(k)]$, $\boldsymbol{y}_{s,k,\text{min}}^{\mathrm{T}} = \left[d_{1,\min}(k), d_{2,\min}(k), d_{3,\min}(k)\right]\ \forall k \in \mathcal{K}_s$, and $\boldsymbol{y}_{h,k,\text{min}}^{\mathrm{T}} = \left[d_{1,\min}(k), d_{2,\min}(k), d_{3,\min}(k), \kappa_{\min}(k)\right]\ \forall k \notin \mathcal{K}_s$, we are able to define the following inequality constraints for the modified cost function (21)

$$\begin{aligned}\mathbf{y}_h - \mathbf{y}_{h,\max} &\le \mathbf{0}, \quad \mathbf{y}_s - \mathbf{y}_{s,\max} \le \epsilon_u, \quad -\epsilon_u \le \mathbf{0},\\ -\mathbf{y}_h + \mathbf{y}_{h,\min} &\le \mathbf{0}, \quad -\mathbf{y}_s + \mathbf{y}_{s,\min} \le \epsilon_l, \quad -\epsilon_l \le \mathbf{0}.\end{aligned}$$

With the additional input constraint (9) and using (22), we integrate these inequality constraints into a single matrix valued condition for the optimization variables $\tilde{\mathbf{u}}$ given by

$$\tilde{\boldsymbol{A}}_c\tilde{\mathbf{u}} \le \tilde{\mathbf{b}}_c \tag{23}$$

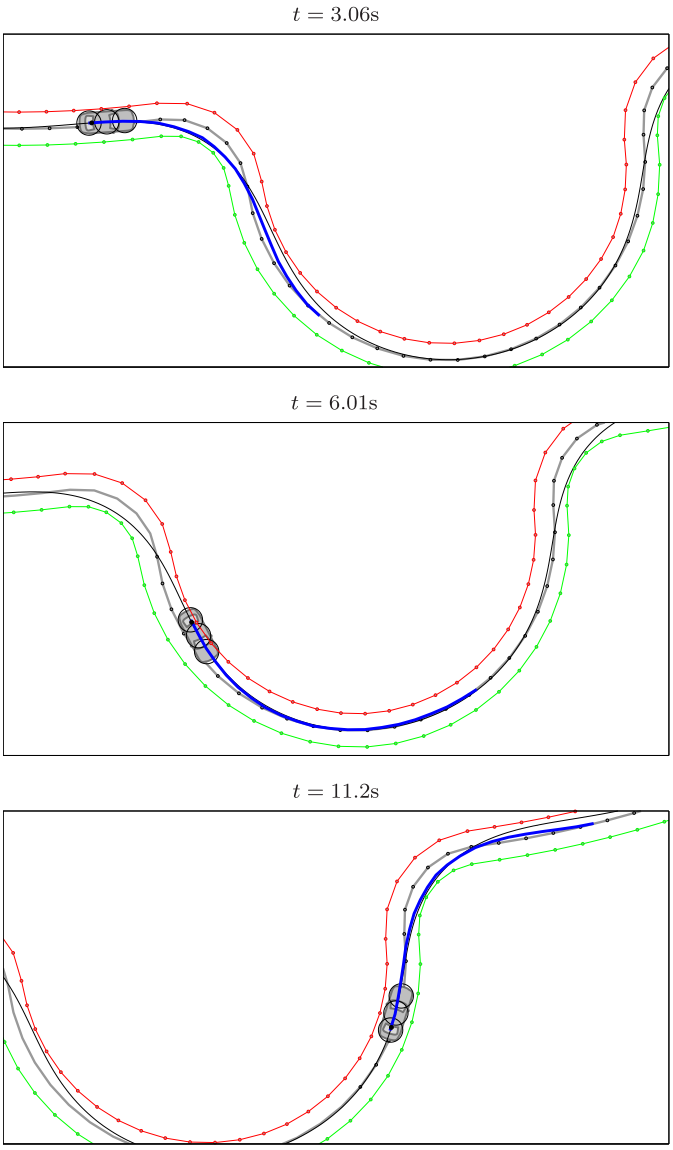

Fig. 7. Bird's eye view of the moving vehicle for the roundabout maneuver at three consecutive time instants.

where

$$\tilde{\boldsymbol{A}}_c = \begin{bmatrix} \begin{bmatrix} \boldsymbol{I}_N \\ -\boldsymbol{I}_N \end{bmatrix} & \mathbf{0}_{2N\times r} \\ \begin{bmatrix} \boldsymbol{\mathcal{C}}_h\boldsymbol{\mathcal{B}} \\ -\boldsymbol{\mathcal{C}}_h\boldsymbol{\mathcal{B}} \end{bmatrix} & \mathbf{0}_{2p_h\times r} \\ \begin{bmatrix} \boldsymbol{\mathcal{C}}_s\boldsymbol{\mathcal{B}} \\ -\boldsymbol{\mathcal{C}}_s\boldsymbol{\mathcal{B}} \end{bmatrix} & \begin{bmatrix} -\mathbf{1}_{p_s} & \mathbf{0}_{p_s} \\ \mathbf{0}_{p_s} & -\mathbf{1}_{p_s} \end{bmatrix} \\ \mathbf{0}_{r\times N} & -\boldsymbol{I}_r \end{bmatrix},$$

$$\tilde{\mathbf{b}}_c = \begin{bmatrix} \begin{bmatrix} \dot{\kappa}_{\max}\ \mathbf{1}_N \\ -\dot{\kappa}_{\min}\ \mathbf{1}_N \end{bmatrix} \\ \begin{bmatrix} \mathbf{y}_{h,\max} - \boldsymbol{\mathcal{C}}_h[\boldsymbol{\mathcal{A}}\boldsymbol{x}_0 + \boldsymbol{\mathcal{E}}\mathbf{z}] \\ -\mathbf{y}_{h,\min} + \boldsymbol{\mathcal{C}}_h[\boldsymbol{\mathcal{A}}\boldsymbol{x}_0 + \boldsymbol{\mathcal{E}}\mathbf{z}] \end{bmatrix} \\ \begin{bmatrix} \mathbf{y}_{s,\max} - \boldsymbol{\mathcal{C}}_s[\boldsymbol{\mathcal{A}}\boldsymbol{x}_0 + \boldsymbol{\mathcal{E}}\mathbf{z}] \\ -\mathbf{y}_{s,\min} + \boldsymbol{\mathcal{C}}_s[\boldsymbol{\mathcal{A}}\boldsymbol{x}_0 + \boldsymbol{\mathcal{E}}\mathbf{z}] \end{bmatrix} \\ \mathbf{0}_r \end{bmatrix}.$$

Thus, in combination with the cost function (21) this formulation still constitutes a QP with increased size.

Authorized licensed use limited to: Yuanfei Lin. Downloaded on August 12,2021 at 13:20:26 UTC from IEEE Xplore. Restrictions apply.

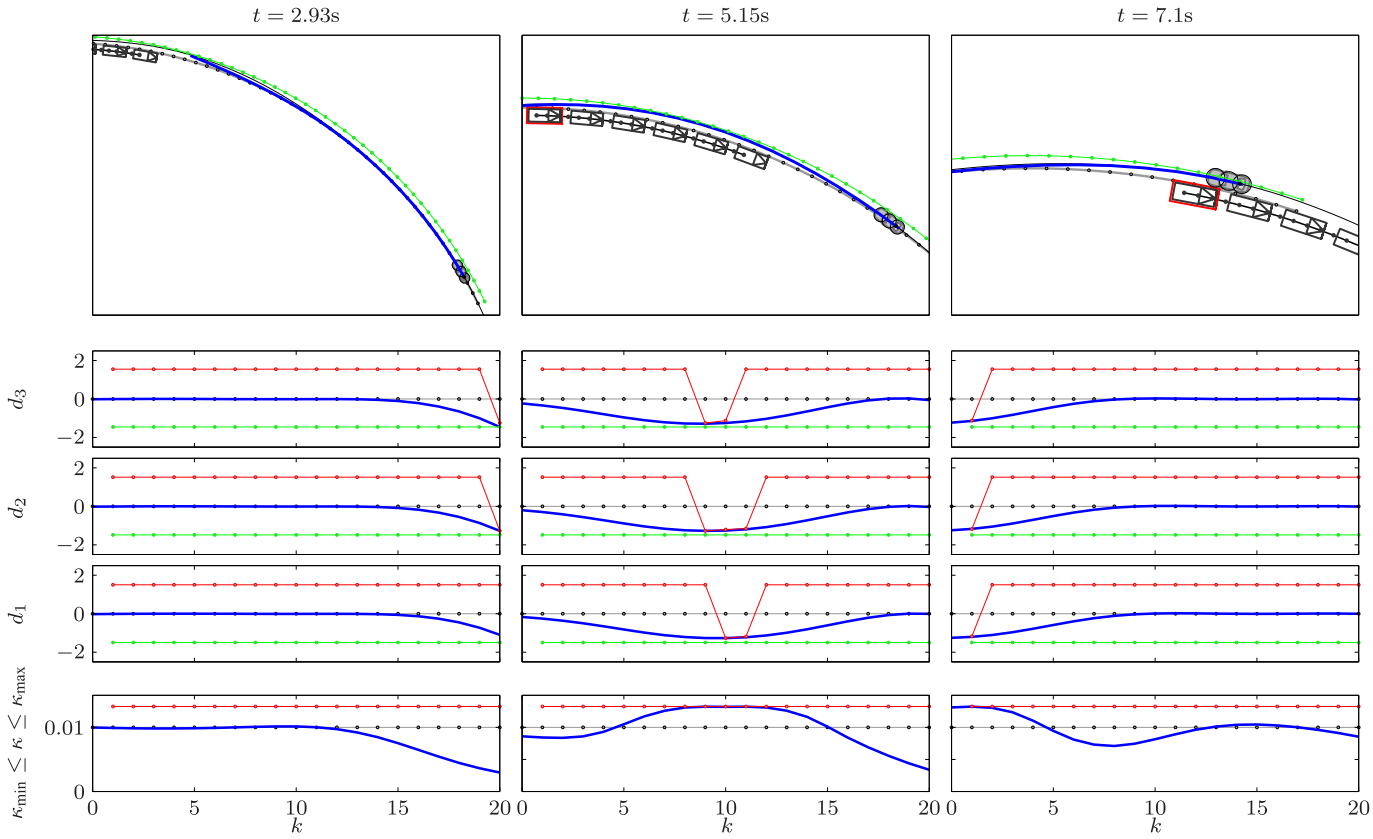

Fig. 8. Bird's eye view of the moving vehicle demonstrating dynamic obstacle avoidance at three consecutive instants of time with the corresponding results for the lateral positions d_1, d_2, d_3 and the curvature κ, where the blue bold line represents the optimized trajectory

VI. Evaluation of the Trajectory Optimization Algorithm

A. Implementation and Test Setup

For the functional demonstration of the proposed algorithm we equipped a BMW i3 and a 5-series BMW with laser scanners, for obstacle detection. Additionally, the software of the electric power steering, the electric braking system, and the engine control system was modified to allow for external inputs for the steering, braking and engine torque signals. The standard sensors' yaw rate, velocity, and acceleration measurements are fed to an odometry model for self-localization (see e. g. [5]). Note, that no external signals for positioning, such as GPS, were used. The algorithm's routines were developed in the Matlab/Simulink environment. Solely the minimization problem itself is solved by a standard QP-solver implemented in the C programming language. All computations run on a dSpace Autobox DS1005 featuring a PowerPC 750GX processor (1GHz) for rapid prototyping applications that require low computation power. Nevertheless a turnaround time of 6ms could be achieved when solving the optimization problem. Furthermore, we choose a prediction horizon of 4.0s by setting $N = 20$ and the optimization step size to 200ms. Additional optimization parameters were set to $N_s = 4$, $u_{\max} = -u_{\min} = 0.25\,^{1}/_{\text{ms}}$ and $\kappa_{\max,\delta} = -\kappa_{\min,\delta} = 0.25\,^{1}/_{\text{m}}$. As the car can travel a considerable distance during a fraction of a second, the computation time and the actuator time delays cannot be neglected. We therefore utilize a delay compensation by prediction [12]. Then, the optimized trajectory is passed on to a stabilizing I/O-linearizing feedback controller, which runs with a cycle time of 10ms and feeds its control signals to the vehicle's actuators. With this setup the algorithm's performance is demonstrated for three different real-world use-cases.

B. Experimental Results

For visualization of the experimental results three snapshots of the vehicle from a bird's eye view are shown in Fig. 4, 7, and 8, along with other measurement signals. These top views are based on the self-localization outlined above. The movement of the vehicle is illustrated by the black line drawn by the rear axle center point. Moreover, the reference curve is depicted by a gray line with black dots, whereas the lateral left and right bounds are colored in red and green, respectively. The optimal trajectory is presented by a bold blue line.

First, a parking use-case is addressed featuring high curvature references and static obstacle avoidance (see Fig. 5). As outlined in Fig. 4, in this scenario the vehicle is driving out of its parking space at speeds of about $1\,^{\text{m}}/_{\text{s}}$ illustrated by three consecutive instants of time. When moving along the reference curve, which serves only as a rough plan, the vehicle is forced to swerve to the right due to some static obstacle marked by the black dots and the red dotted line. Already at the first time instant pictured, the proposed algorithm plans around the

Authorized licensed use limited to: Yuanfei Lin. Downloaded on August 12,2021 at 13:20:26 UTC from IEEE Xplore. Restrictions apply.

obstacle (bold blue line) to avoid a collision with the vehicle front. As represented by the graphs for the lateral position d_1 and the optimized curvature κ in Fig. 4, the algorithm uses the maximum admissible deviation to both sides (red and green dotted line) of the reference curve smoothing out the discontinuous reference curvature to ensure a comfortable driving behavior while successfully avoiding collisions.

Second, a roundabout maneuver is presented illustrating the algorithm's ability of generating user-acceptable trajectories also for dynamic maneuvers. In order to achieve this, the vehicle deviates from the reference curve given from a digital street map (dotted gray line) minimizing the lateral jerk and acceleration, as shown in Fig. 7. Before entering the roundabout the vehicle deviates to the left to increase the turning radius before cutting the corner. By doing so, the algorithm maximizes the driver's comfort resulting in a very smooth steering action even for maneuvers featuring high lateral accelerations of $a_{\text{lat}} = 7.0\text{m/s}^2$, as depicted by the closed loop measurement signals in Fig. 6. In this figure, the three time instants of the snap-shots are marked by vertical black lines. This behavior continues as the vehicle is driving through the roundabout depicted by the optimized trajectory (bold blue line) in the snap-shot pictures of Fig. 7. Please note the difference between the end of the optimized (blue) and the eventually driven path (black), which is due to the receding optimization horizon.

Finally, a third use-case is presents dynamic obstacle avoidance at higher speeds of about 20m/s. As illustrated for three consecutive time instants in Fig. 8, the algorithm consistently plans a collision avoiding maneuver based on the predicted movement of the dynamic object. Here, the future object positions result from a kinematic prediction with respect to the reference curve and are marked by gray rectangular boxes. The object's current position is highlighted by an additional red box emphasizing an integrated safety margin. In the first instant of time, an evasive maneuver is planned as indicated by the collision avoidance constraints for the lateral positions d_2 and d_3. For the following instants of time, these constraints move closer towards the current ego vehicle position, as the ego vehicle is approaching the oncoming dynamic object. At the second and third time step, both the constraints for collision avoidance and the traction constraints for the curvature κ with $\mu = 0.5$ are active, as presented in the lower graphs of Fig. 8. Therefore, the algorithm not only avoids the impending collision successfully, as shown by the top view snap-shot for the third time instant, but also ensures stable driving conditions throughout the maneuver at the same time.

VII. Conclusion and Future Work

Facing the problem of planning a comfortable and safe future vehicle motion for automated vehicle applications in real time, this paper presents a highly efficient trajectory optimization algorithm for lateral vehicle guidance along a given reference curve. Although this algorithm only allows the formulation of local optimal control problems, its elaborate linear-quadratic problem formulation features several beneficial properties that motivate its usage for a broad range of assisted and automated driving applications.

First, if there exists a solution to the optimization problem, the solution quickly converges under guarantee and can be determined by a low-performance electronic control unit in milliseconds. Second, as the solution lies in the continuous input and state space, a consistent replanning of the optimal solution is assured, minimizing the effect of sensor noise and system disturbances. Third, a great number of system constraints can be accounted for at very little additional computational costs. Therefore, as demonstrated for different use-cases, traction constraints and actuator limits can be easily considered. Also collision avoidance cannot only be achieved for static obstacles, but also for dynamic ones. At last, even reference curves with high and discontinuous curvatures (e. g. given by sparse road maps or generated for low-dimensional system models) can be tracked smoothly leading to a very comfortable and qualified driving behavior.

Future research will focus on a more substantiated method for tuning the cost functions weighting factors and on the combined optimization of the lateral and longitudinal motion planning by solving the lateral optimization problem for a certain number of different future velocity profile candidates.

References

[1] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma, "An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios," *Int. J. Vehicle Auto. Syst.*, vol. 8, nos. 2–4, pp. 190–216, 2010.

[2] M. Bahadorian, B. Savkovic, R. Eaton, and T. Hesketh, "Robust model predictive control for automated trajectory tracking of an unmanned ground vehicle," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2012, pp. 4251–4256.

[3] R. Bellman, "The theory of dynamic programming," Rand Corporation, Santa Monica CA, USA, Tech. Rep. Number RAND-P-550, 1954.

[4] A. Bemporad and C. Rocchi, "Decentralized linear time-varying model predictive control of a formation of unmanned aerial vehicles," in *Proc. 50th IEEE Conf. Decision Control Eur. Control Conf. (CDC-ECC)*, Dec. 2011, pp. 7488–7493.

[5] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile robot positioning—Sensors and techniques," *J. Robotic Syst.*, vol. 14, pp. 231–249, 1997.

[6] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control For Linear and Hybrid Systems*. Cambridge, U.K.: Cambridge Univ. Press, in press.

[7] T. Brandt, "A predictive potential field concept for shared vehicle guidance," Ph.D. dissertation, Heinz Nixdorf Inst., Univ. Paderborn, Paderborn, Germany, 2008.

[8] A. Carvalho, Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, "Predictive control of an autonomous ground vehicle using an iterative linearization approach," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 2335–2340.

[9] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "A hierarchical model predictive control framework for autonomous ground vehicles," in *Proc. Amer. Control Conf.*, Jun. 2008, pp. 3719–3724.

[10] P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H. E. Tseng, "A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems," in *Proc. 46th IEEE Conf. Decision Control*, Dec. 2007, pp. 2980–2985.

[11] S. J. Fesharaki and H. A. Talebi, "Active front steering using stable model predictive control approach via LMI," *J. Control Eng. Appl. Informat.*, vol. 16, no. 2, pp. 90–97, 2014.

[12] R. Findeisen, L. Grüne, J. Pannek, and P. Varutti. (May 2011). "Robustness of prediction based delay compensation for nonlinear systems," [Online]. Available: https://arxiv.org/abs/1105.3268

[13] L. Fletcher *et al.*, "The MIT-Cornell collision and why it happened," *J. Field Robot.*, vol. 25, no. 10, pp. 775–807, Oct. 2008.

Authorized licensed use limited to: Yuanfei Lin. Downloaded on August 12,2021 at 13:20:26 UTC from IEEE Xplore. Restrictions apply.

[14] Y. Gao *et al.*, "Spatial predictive control for agile semi-autonomous ground vehicles," in *Proc. 11th Int. Symp. Adv. Vehicle Control*, 2012. [Online]. Available: http://www.me.berkeley.edu/~frborrel/pdfpub/pub-1054.pdf
[15] A. Gray, M. Ali, Y. Gao, J. Hedrick, and F. Borrelli, "Semi-autonomous vehicle control for road departure and obstacle avoidance," *IFAC Control Transp. Syst.*, pp. 1–6, Sep. 2012.
[16] T. Gu and J. M. Dolan, "On-road motion planning for autonomous vehicles," in *Intelligent Robotics and Applications*. Berlin, Germany: Springer, 2012, pp. 588–597.
[17] T. Gu and J. M. Dolan, "Toward human-like motion planning in urban environments," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 350–355.
[18] T. Gu, J. Snider, J. M. Dolan, and J.-W. Lee, "Focused trajectory planning for autonomous on-road driving," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2013, pp. 547–552.
[19] B. Gutjahr and M. Werling, "Automatic collision avoidance during parking and maneuvering—An optimal control approach," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 636–641.
[20] A. Katriniok and D. Abel, "LTV-MPC approach for lateral vehicle guidance by front steering at the limits of vehicle dynamics," in *Proc. 50th IEEE Conf. Decision Control Eur. Control Conf. (CDC-ECC)*, Dec. 2011, pp. 6828–6833.
[21] A. Katriniok, J. P. Maschuw, F. Christen, L. Eckstein, and D. Abel, "Optimal vehicle dynamics control for combined longitudinal and lateral autonomous vehicle guidance," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2013, pp. 974–979.
[22] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1991, pp. 1398–1404.
[23] B. H. Krogh, "A generalized potential field approach to obstacle avoidance control," in *Proc. Int. Robot. Res. Conf.*, Bethlehem, PA, USA, 1984.
[24] K. Kunz, S. M. Huck, and T. H. Summers, "Fast model predictive control of miniature helicopters," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2013, pp. 1377–1382.
[25] X. Li, Z. Sun, Q. Chen, and J. Wang, "A novel path tracking controller for ackerman steering vehicles," in *Proc. 32nd Chin. Control Conf. (CCC)*, Jul. 2013, pp. 4177–4182.
[26] X. Li, Z. Sun, D. Liu, Q. Zhu, and Z. Huang, "Combining local trajectory planning and tracking control for autonomous ground vehicles navigating along a reference path," in *Proc. IEEE 17th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 725–731.
[27] J. R. McBride *et al.*, "A perspective on emerging automotive safety applications, derived from lessons learned through participation in the darpa grand challenges," *J. Field Robot.*, vol. 25, no. 10, pp. 808–840, Oct. 2008.
[28] M. A. Mousavi, Z. Heshmati, and B. Moshiri, "LTV-MPC based path planning of an autonomous vehicle via convex optimization," in *Proc. 21st Iranian Conf. Elect. Eng. (ICEE)*, May 2013, pp. 1–7.
[29] J. Nilsson, M. Ali, P. Falcone, and J. Sjöberg, "Predictive manoeuvre generation for automated driving," in *Proc. 16th Int. IEEE Annu. Conf. Intell. Transp. Syst.*, Oct. 2013, pp. 418–423.
[30] S.-R. Oh and J. Sun, "Path following of underactuated marine surface vessels using line-of-sight based model predictive control," *Ocean Eng.*, vol. 37, nos. 2–3, pp. 289–295, 2010.
[31] J. M. Park, D. W. Kim, Y. S. Yoon, H. J. Kim, and K. S. Yi, "Obstacle avoidance of autonomous vehicles based on model predictive control," *Proc. Inst. Mech. Eng., Part D, J. Automobile Eng.*, vol. 223, no. 12, pp. 1499–1516, Dec. 2009.
[32] J. B. Rawlings, "Tutorial overview of model predictive control," *IEEE Control Syst.*, vol. 20, no. 3, pp. 38–52, Jun. 2000.
[33] B. Reimer, B. Mehler, and J. F. Coughlin, "An evaluation of driver reactions to new vehicle parking assist technologies developed to reduce driver stress," MIT Center for Transportation Logistics, Cambridge, MA, USA, Tech. Rep., 2010, pp. 1–26, vol. 4.
[34] G. Tanzmeister, M. Friedl, D. Wollherr, and M. Buss, "Path planning on grid maps with unknown goal poses," in *Proc. Conf. Intell. Transp. Syst.*, Oct. 2013, pp. 430–435.
[35] V. Turri, A. Carvalho, H. E. Tseng, K. H. Johansson, and F. Borrelli, "Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2013, pp. 378–383.
[36] J. Wei, J. M. Snider, T. Gu, J. M. Dolan, and B. Litkouhi, "A behavioral planning framework for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 458–464.
[37] M. Werling, T. Gindele, D. Jagszent, and L. Gröll, "A robust algorithm for handling moving traffic in urban scenarios," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2008, pp. 1108–1112.
[38] M. Werling, S. Kammel, J. Ziegler, and L. Gröll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *Int. J. Robot. Res.*, vol. 31, no. 3, pp. 346–359, 2012.
[39] M. Werling and D. Liccardo, "Automatic collision avoidance using model-predictive online optimization," in *Proc. IEEE 51st Annu. Conf. Decision Control (CDC)*, Dec. 2012, pp. 6309–6314.
[40] J. Ziegler and C. Stiller, "Fast collision checking for intelligent vehicle motion planning," in *Proc. Intell. Vehicles Symp.*, Jun. 2010, pp. 518–522.
[41] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for Bertha—A local, continuous method," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 450–457.

Benjamin Gutjahr received the degrees from Universität Stuttgart, Germany; University of Connecticut, USA; and Norwegian University of Science and Technology, Norway, all in engineering cybernetics, and the Dipl.-Ing. degree in 2012. He is currently working toward the Ph.D. degree with the BMW Group Research and Technology and Karlsruhe Institute of Technology. His main area of research is trajectory planning and vehicle control for automated collision avoidance.

Lutz Gröll received the degree in automatic control from Technische Universität Dresden, Germany, and the Ph.D. degree in electrical engineering from Technische Universität Dresden, Germany, in 1995. Since 2001, he has been the Head of Research Group Process Modeling and Control Engineering with the Institute for Applied Computer Science, Karlsruhe Institute of Technology, Germany. His research interests are in parameter identification, nonlinear control, and optimization theory.

Moritz Werling received the degree in mechanical engineering from Technische Universität Karlsruhe, Germany, and Purdue University, USA, and the Dr.Ing. degree (Hons.) from Technische Universität Karlsruhe in 2010. Since 2011, he has been a Research Engineer with the BMW Group Research and Technology, Munich, Germany. He was part of the Team AnnieWAY in the 2007 DARPA Urban Challenge and Team Stanford Racing in 2009. His research interests include trajectory planning and vehicle control for Advanced Driver Assistance Systems. He received the KIT Best Dissertation Award for Systems and Processes in 2011 and the Ernst-Schoemperlen Best Dissertation Award in 2012.

Authorized licensed use limited to: Yuanfei Lin. Downloaded on August 12,2021 at 13:20:26 UTC from IEEE Xplore. Restrictions apply.