2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Trajectory Optimization for Car-Like Vehicles in Structured and Semi-Structured Environments*

Clemens Nietzschmann[1], Sebastian Klaudt[1], Christoph Klas[2], Devid Will[1] and Lutz Eckstein[3]

Abstract— **In this paper we propose a local trajectory planner for front steered car-like vehicles based on a combined direct optimization of the lateral and longitudinal vehicle guidance. The planner is designed for continuously optimizing a local trajectory based on a provided reference path in a structured or semi-structured driving environment. The planner respects constraints of the driving dynamics as well as actuator limitations and avoids static and dynamic obstacles. It is not restricted to a limited set of maneuvers. The implementation of the planner allows an online adaptation of the resulting driving behavior to satisfy different comfort or driving style demands. After Software-in-the-Loop simulations the algorithm was tested in two different real-world driving scenarios in ika's automated vehicle which provides interfaces for full lateral and longitudinal control.**

I. INTRODUCTION

Besides perception, localization and behavior planning, the trajectory optimization is a key component for implementing automated driving functions. To achieve a comfortable and safe driving experience a motion planning module must provide input for the vehicle's longitudinal as well as steering controllers. This input is always a trade-off between keeping the vehicle's motion safe, reasonably fast and comfortable. One approach to solve the trajectory planning problem is the formulation of an optimal control problem (OCP) [1], which can be solved using dynamic programming, direct methods and indirect methods [2]. Dynamic programming is based on Bellman's principle of optimality [3], but suffers from the curse of dimensionality restricting the usable number of states. Indirect methods use the necessary condition of optimality to formulate a boundary value problem which is then solved numerically or if possible analytically. Since the handling of active constraints is difficult, the use of this approach is limited. Therefore, direct methods are preferable for local trajectory planning as well as for optimizing a coarse reference path [4].

We suggest formulating the trajectory planning problem as a constrained non-linear optimization problem with a quadratic cost function with small residuals based on a kinematic single track model. The left and right boundary of the driving corridor and the reference path are interpolated with cubic splines to provide smooth gradients for evaluating partial derivatives of the objective function. Dynamic obstacles are modeled as scalar potentials mapped to their predicted movement directions. They are incorporated into the objective function as quadratic penalty terms. During development special attention was paid to a wide area of application regarding the necessary interfaces for the input data as well as an adaptable driving behavior. Since a direct optimization cannot guarantee a global optimal solution a behavior generation module is required to handle situations in which discrete decisions must be made (e.g. initiation of lane change maneuvers, entering an intersection, etc.).

A. Related Work

Intelligent vehicle literature shows that a lot of different path and trajectory planning algorithms tailored to specific use cases have been developed over the past decades. Research showed that combinatorial problems as e.g. finding the shortest path in an unstructured cluttered environment can be efficiently tackled by algorithms based on graph search algorithms (e.g. Dijkstra, A* or State Lattice Algorithm) [5]. However, generating smooth trajectories respecting vehicle dynamics and actuator limitations in low as well as middle to high speed ranges remains a difficult problem. Due to recent progress in the field of specialized optimization algorithms, code generation and faster processors, direct numeric optimization is applicable even to complex non-linear optimal control problems requiring runtimes in the range of milliseconds [6], [7], [8]. For this reason, a shift from sampling based methods for trajectory generation (e.g. [9]) to model predictive control approaches can be observed [10], [11], [12]. Although MPC formulations based on continuous optimization in general cannot guarantee global optimality, they scale well with the number of states, are not restricted to a fixed maneuver set and can handle multiple constraints. This makes them particularly suitable for local trajectory optimization.

II. LOCAL TRAJECTORY PLANNER

The proposed trajectory planner is designed for automated driving of front steered car-like vehicles in a velocity range of 0 - 130 kph. This creates multiple challenges since the algorithm has to support a wide variety of driving maneuvers, avoid collisions with other road users and provide a comfortable parameterizable driving behavior while keeping a real-time feasible run-time. The range of perceived comfort

*Research is funded by the German Federal Ministry of Education and Research within the project PARIS (reference number: 16ES0602).

[1]Clemens Nietzschmann, Sebastian Klaudt, Devid Will are with the driver assistance department of the Institute for Automotive Engineering (ika), RWTH Aachen University, 52074 Aachen, Germany {clemens.nietzschmann, sebastian.klaudt, devid.will}@ika.rwth-aachen.de

[2]Christoph Klas is with the Forschungsgesellschaft fuer Kraftfahrwesen mbH Aachen, 52074 Aachen, Germany christoph.klas@fka.de

[3]Lutz Eckstein is head of the Institute for Automotive Engineering (ika), RWTH Aachen University, 52074 Aachen, Germany lutz.eckstein@ika.rwth-aachen.de

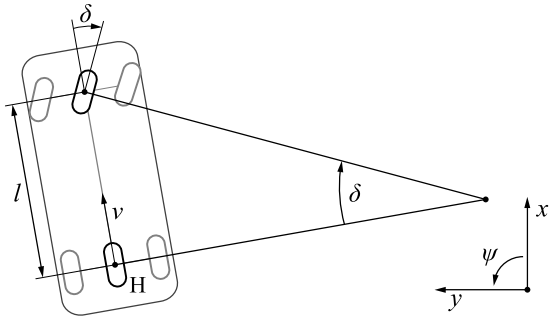

Fig. 1. Implemented kinematic single-track model

varies vastly from driver to driver, e.g. regarding the driving behavior on tight curves (lateral) or during deceleration (longitudinal). Therefore, a further requirement is the demand for an online adaptability of the comfort-relevant shape of the resulting trajectory. The planner needs to be able to consider personal wishes in addition to safety and physical constraints by design. The required output is a continuous trajectory. It defines the desired longitudinal and lateral motion, which are then tracked by suitable controllers. To solve the described problem, we formulate the trajectory planning problem as a non-linear optimization problem.

A. Vehicle Model

Since the trajectory planner is designed for comfortable, not time-optimal, driving maneuvers as well as low-speed and high-speed maneuvers a standard kinematic single-track model is used as system model $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))$:

$$\begin{aligned} \dot{x} &= v\cos(\psi) \\ \dot{y} &= v\sin(\psi) \\ \dot{s} &= v \\ \dot{v} &= a_{long} \\ \dot{a}_{long} &= j_{long} \\ \dot{\psi} &= \frac{v}{l}\tan(\delta) \\ \dot{\delta} &= \alpha. \end{aligned} \tag{1}$$

The system comprises seven states $\mathbf{x} = (x, y, s, v, a_{long}, \psi, \delta)^T$ and two control inputs $\mathbf{u} = (j_{long}, \alpha)^T$: x, y and ψ is the position in two-dimensional vehicle coordinates, s is the driven distance, v is the velocity and a_{long} is the longitudinal acceleration at the center of the ego-vehicle's rear axis H, l is the wheelbase and δ is the Ackermann steering angle (see fig. 1). The first control input is the longitudinal jerk j_{long} and the second is the steering rate α.

B. Optimization Problem

As stated the trajectory planning problem is formulated as an OCP with an objective function (2) which should be minimized for the prediction horizon t_f comprising a running cost term L and terminal cost term E subject to the system model (3), the initial condition (4) and path constraints (5) - (9).

$$\min_{\mathbf{u}(t)} J(\mathbf{x}(t), \mathbf{u}(t)) = \int_0^{t_f} L(\mathbf{x}(t), \mathbf{u}(t))\,\mathrm{d}t + E(\mathbf{x}(t_f)) \tag{2}$$

$$\text{subject to} \quad \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \tag{3}$$

$$\mathbf{x}(0) = \hat{\mathbf{x}}_0 \tag{4}$$

$$v \geq 0 \tag{5}$$

$$\|\mathbf{a}\|_2 \leq a_{max}, \quad \mathbf{a} = (a_{long}, a_{lat})^T \tag{6}$$

$$|j_{long}| \leq j_{max} \tag{7}$$

$$|\delta| \leq \delta_{max} \tag{8}$$

$$|\alpha| \leq \alpha_{max} \tag{9}$$

J is a least-squares cost function with small residuals and the problem is discretized with the direct multiple shooting approach [13]. The trajectory is sampled over time and in the vehicle's ego coordinate frame. Note that in contrast to other formulations like [10] or [11] this formulation does not include any path constraints related to collision avoidance since these are integrated into the running cost term L in form of quadratic penalty terms [14].

The reason is, that an automated vehicle does not only have to avoid collision with other road participants, but must also guarantee reasonable safe distances. For instance the desired lateral safe distance to an oncoming vehicle is greater than to a vehicle driving in front of the ego-vehicle. A scalable cost function term is one solution to account for this need without the numerical issues introduced by integrating the safe distances as path constraints. For instance, in case of an aggressive cut-in of another vehicle on the highway the safe distance in longitudinal direction is violated, therefore the optimization problem becomes unsolvable. Our penalty function based approach can handle this situation without any adaptation of the problem formulation.

Equation (5) can also be reformulated so that only driving in reverse direction is allowed. The maximum permissible values for the steering angle (8), the longitudinal jerk (7) and steering rate α (9) take actuator limitations into account. The restriction of the maximum overall acceleration (6) prevents the generation of non-drivable trajectories and can be used to restrict the acceleration e.g. in case of slippery road conditions.

C. Environment Representation

In order to provide a generic interface regarding the input data needed, all paths, lanes and boundaries have to be provided as fixed-size set of ordered sampled points (x, y) in vehicle coordinates as shown in fig. 2. The reference path moreover needs to provide the desired velocity at each sample point and it does not necessarily need to be collision-free. Static obstacles have to be included in the left and right boundary. From the sampled points, a smoothed cubic spline interpolation is done for the reference path and boundaries as the direct numeric optimization requires a twice continuously differentiable objective function. For computing the shortest distance to the left and right boundary $(d_{n,l}, d_{n,r})$ the vehicle geometry is approximated by three circles. Note that the

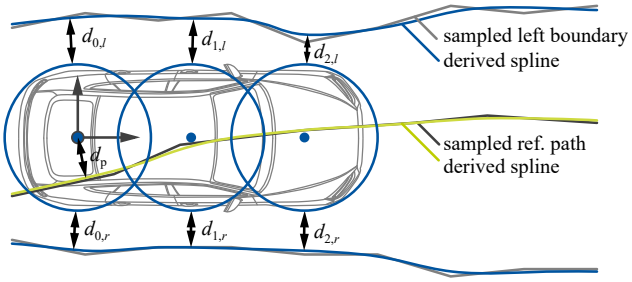

Fig. 2. Illustration of the sampled boundary description of the driving corridor, the reference path and their corresponding spline interpolations. The shape of the vehicle is approximated by three circles (blue) and the distances used in the objective function are shown in black.

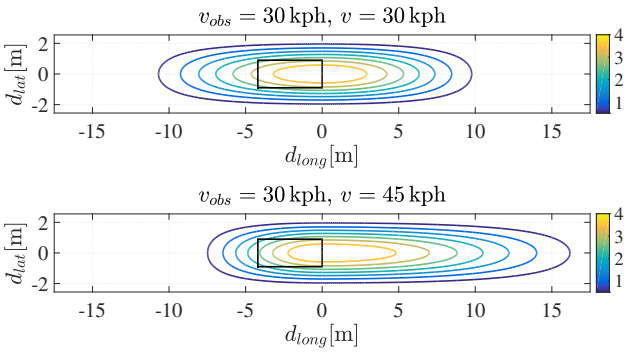

Fig. 3. Illustration of the penalty function for avoiding dynamic obstacles based on a scalar potential which is essentially the multiplication of two cosine functions. Note that the vehicle illustrated by the black rectangle is the target vehicle and the ego vehicle is closing in from the right side and the scalar potential is related to the center of the rear axle H. The scalar potential depends on the desired time gap p_τ specified by the user and the relative velocity between the obstacle and the ego vehicle. The distance d_{lat} is the lateral distance of the ego vehicle's current position H to the predicted movement path of the dynamic obstacle and d_{lon} is the integrated arc length along the predicted movement direction between the nearest point on the movement direction relative to H and the position of the obstacle.

distance calculation between a point and a spline poses an additional non-linear optimization problem within the trajectory planning problem.

Other road users are treated as dynamic obstacles. Besides pure collision avoidance, it is necessary to keep reasonable lateral and longitudinal distances to other road users to provide a safe driving experience. Therefore, all dynamic obstacles are modeled as scalar potentials, which are mapped to the predicted object movement direction based on their particular length, width, orientation and current velocity. In case of driving in a structured environment a simple prediction model assuming a constant velocity as in [11] is used, with the difference that a unique movement direction can be assigned to each dynamic obstacle. A movement direction must be specified as fixed-size ordered set of sample points (x,y) similar to the reference path. Fig. 3 illustrates the undistorted scalar potential for a target vehicle driving straight.

The given reference velocity profile is smoothed in a preprocessing step based on the two user parameters $p_{a,lat}$ and $p_{a,lon}$, which specify the desired lateral and longitudinal acceleration.

D. Objective Function

The discretized objective function J with N intervals has been formulated as follows:

$$\min_{\substack{\mathbf{x}_0,\dots,\mathbf{x}_N \\ \mathbf{u}_0,\dots,\mathbf{u}_{N-1}}} \sum_{k=1}^{N} [w_v c_{v,k}^2 + w_p c_{p,k}^2 + w_a c_{a,k}^2 + w_j c_{j_{lat},k}^2 + w_j c_{j,k}^2 + w_\alpha c_{\alpha,k}^2 + w_{stat} c_{stat,k}^2 + w_{dyn} c_{dyn,k}^2] + w_\psi c_{\psi,N}^2. \tag{10}$$

Note that all objective function terms $c_{*,k}$ at a specific discretization point k depend on the states $\mathbf{x}_k$ and controls $\mathbf{u}_k$, see eq. (11) - (15). The prediction horizon t_f is the product of the number of discretization intervals N and the sample time t_s. The first term $c_{v,k}$ evaluates the current deviation from the desired reference velocity and is the driving component in the problem formulation. The second term $c_{p,k}$ penalizes the deviation from the reference path. The following terms are mainly comfort-related: $c_{a,k}$ limits the overall acceleration, $c_{j_{lat},k}$ and $c_{j,k}$ limit the lateral and longitudinal jerk and $c_{\alpha,k}$ limits the steering rate. The last two parts of the running cost term $c_{stat,k}$ and $c_{dyn,k}$ are the penalty terms for keeping the vehicle inside the driving corridor and avoid collisions with dynamic obstacles. The final term $c_{\psi,N}$ evaluates the deviation of the yaw angle ψ from the reference path spline at the end of the prediction horizon. All weight parameters w are time invariant. In general, w_{dyn} and w_{stat} should be one order of magnitude larger than the rest of the weights and w_ψ depends on the number of discretization intervals.

$$c_{v,k} = \frac{v_{ref} - v_k}{\max(v_{ref}, 2.78)} \qquad c_{p,k} = \frac{d_{p,k}}{p_{p,ref}} \tag{11}$$

$$c_{a,k} = \frac{\|\mathbf{a}_k\|_2^2}{p_{a,ref}^2} \qquad c_{j_{lat},k} = \frac{j_{lat,k}}{p_{j,ref}} \tag{12}$$

$$c_{j,k} = \frac{j_k}{p_{j,ref}} \qquad c_{\alpha,k} = \frac{\alpha_k}{p_{\alpha,ref}} \tag{13}$$

$$c_{stat,k} = c_{s,left,k} + c_{s,right,k} \tag{14}$$

$$c_{s,left,k} = \sum_{n=0}^{2} \begin{cases} 0 & d_{n,left,k} > p_{d,stat} \\ \frac{1+0.1\,v_k}{3\,p_{d,stat}} \left(1 - \frac{d_{n,left,k}}{p_{d,stat}}\right) & d_{n,left,k} \le p_{d,stat} \end{cases} \tag{15}$$

All cost terms are online-scalable by a set of user parameters p, similar to the scaling of the scalar potential field. Only for $c_{v,k}$ the scaling factor is capped at 2.78 m/s to ensure numeric stability for low velocities.

The user parameters of the objective function also provide the instrument to adapt the vehicle's behavior to fit the particular passenger's comfort requirements. By adapting those parameters, the optimization result can be specifically shaped within the available collision-free and vehicle-drivable space, which is guaranteed by the objective function.

III. IMPLEMENTATION

The trajectory optimization is implemented in C++11 single-threaded, without dynamic memory allocations and provides prediction horizons up to 8.0 s with a sample time t_s of 0.1 s or 0.2 s. Elektrobit's Automotive Data and Time-Triggered Framework (ADTF) is used as middleware for the integration in ika's research vehicle. All parameters and weights used in the objective function can be changed at run-time. The code generation tool of the ACADO Toolkit [15] in combination with the solver qpOases [16] has been used to generate tailored C-code for integrating the system model and solving the discretized optimization problem using the Real-Time-Iteration scheme and a Gauss-Newton approximation of the Hessian [17]. The partial derivatives of the objective function $\frac{\partial J_k}{\partial \mathbf{x}_k}$ and $\frac{\partial J_k}{\partial \mathbf{u}_k}$ can be expressed analytically and are implemented as custom C-function in the code generated by the ACADO Toolkit. To achieve run-times feasible in real time, all iterative optimization processes like the trajectory optimization itself but also the point-to-spline distance calculation have a fixed maximum iteration count.

A. Trajectory Reference Information

Within this paper, the trajectory optimization method is applied and tested in two different use-cases: Automated driving in semi-structured areas such as parking areas and driving on urban roads.

Semi-structured in this case means, that no separate lanes for each driving direction can be modeled explicitly, but a lane-network can be mapped onto the parking area to model specific traffic rules for accessing the parking spots [18]. To test the trajectory optimization in such a scenario, it was combined with the path planner for the automated valet parking system presented in [19]. Within the system, the lane-network and the parking spots were modeled as a graph, allowing the generation of a route to the desired parking spot, which obeys the traffic rules. Drivable and non-drivable areas are modeled in an occupancy grid. The path planner, which is implemented in [19] uses the occupancy grid map for collision checks and the route within the heuristic and cost function for both navigating and parking into the parking spot. The trajectory optimization however requires the boundaries as a set of ordered sampled points (x,y). Therefore, boundaries were generated as the closest distance between the planned path and the first occupied cell in the direction of the normal vector at each sample point of the path. The planned path is converted section-wise into a vehicle-center path. A section is defined as a set of sampled poses from the path having the same direction of driving. As described in [20], the closest part of the current section is determined using its euclidean distance to the vehicle's current position. Since the path planner already considers the vehicle's dynamics and constraints within its control-set, this part of the reference path could be forwarded directly to the controller. However, in this test-scenario the path is used as an input to the trajectory optimization. The trajectory optimization uses the constant velocity from the planned path as an upper limit for its calculation. The boundaries are also selected section-wise and are converted into vehicle-centered coordinates to be used in the optimization.

The second use-case for the trajectory optimization was driving in an urban scenario with turns and a roundabout. The necessary precise map information for this use-case was created as a series of WGS84 coordinates using QGIS [21] and digital orthophotos with a resolution of 20 cm/pixel. This information provides the reference path as well as fixed static driving corridor boundaries. The conversion into a local vehicle-centered reference trajectory is done by the same principle as in the valet parking scenario.

B. Trajectory Tracking Controller

To closely follow the optimized trajectory, separate tracking controllers for lateral and longitudinal vehicle guidance are used.

The lateral controller generates a steering angle command. It combines a curvature-based feed-forward controller with a feed-back yaw rate controller. Feed-back control is implemented in a cascaded structure of PID-controllers to reflect the integrative relation of the lateral motion with respect to the given trajectory. The outer loop tracks the lateral deviation from the intended trajectory and generates a reference value for the relative yaw angle. From this, the heading controller level calculates the necessary yaw rate. The resulting yaw rate request is combined with the feed-forward yaw rate demand on the basis of the trajectory's curvature. This overall yaw rate is furthermore stabilized by a quick differential yaw rate feed-back controller, which improves the tracking stability especially at higher velocities. Based on an inverse vehicle model, the steering angle is calculated for the current working point. This approach allows the use of the controller in different vehicles by adapting the underlying model. To account for the non-linearities of the lateral motion, the controller gains are adjusted along the longitudinal velocity. The lateral controllers have been parameterized to work over the full velocity range, including reverse driving scenarios.

The longitudinal motion is controlled by commanding the longitudinal acceleration of the vehicle. The acceleration profile of the optimal trajectory is directly used as a feed-forward input. A feed-back loop tracks the velocity error and applies additional acceleration commands by means of a gain-scheduled linear controller. An additional low-level acceleration controller checks the resulting longitudinal movement and, if necessary, adapts the command, which is sent to the interface to account for interface non-linearities.

C. Test Vehicle

The tests to analyze and evaluate the performance of the closed-loop system are conducted in ika's automated vehicle, which is shown in fig. 4. The vehicle is a modified 2008 VW Passat CC, which is equipped with an extensive sensor set. Amongst other sensors, it includes an RTK system, an object and lane detection camera, two long- and mid-range as well as four short-range radar sensors, two laser scanners and

Fig. 4. ika's automated vehicle: 2008 Passat CC with environment sensor equipment (left); trunk view with additional computing hardware (right)

twelve ultrasonic sensors. The setup provides a 360°-field-of-view around the vehicle. Furthermore, it enables precise localization in the traffic environment.

The vehicle is equipped with several computing devices. The trajectory optimization and the necessary input modules are running in ADTF under Ubuntu 16.04 on a pc with an Intel Xeon E5-2630-v4 (max. core frequency 3.10 GHz), which is installed in the trunk of the vehicle. The vehicle control module is implemented on a dSpace MicroAutoBox. It runs the longitudinal and lateral tracking controllers and generates the necessary control commands. Furthermore, it implements the CAN communication with the vehicle interfaces and ensures fail-safe behavior.

The lateral motion of the vehicle is controlled by the steering wheel angle interface of the electronic power steering system (EPS), which has been modified to operate over the full velocity range. For driving in structured environments, the longitudinal movement is controlled by an acceleration request, which is commanded to the vehicle's adaptive-cruise-control interface. At low velocities or in scenarios which require a driving direction change, as e.g. in parking maneuvers, the longitudinal guidance is realized by commanding a retro-fitted brake booster, the accelerator pedal value and a shift-by-wire interface. To achieve this, a low-level controller converts the requested longitudinal acceleration into the necessary commands and allows the use of the identical longitudinal tracking controller for all scenarios.

IV. Test Results

After successful tests in simulation, a validation of the closed-loop performance of the trajectory planner was conducted in real driving scenarios. The two presented scenarios illustrate the wide application range of the proposed system. The system parameterization for the test cases is given in Tab. I. The planning frequency was set to 10 Hz.

A. Automated Valet Parking

Fig. 5 shows the processed map of RWTH Aachen University's parking garage for two test scenarios: Parking the vehicle into a spot (upper part of the figure) and pulling out of the parking spot and driving to a designated pick-up area (lower part). For each scenario, the figure shows the generated additional boundary on both sides of the planned path.

TABLE I
Used parameter Sets

Parameter	Valet Parking	On-Road Driving
$p_{a,ref}$	4.0 m/s^2	4.0 m/s^2
$p_{a,lat}$	1.5 m/s^2	2.0 m/s^2
$p_{a,lon}$	1.25 m/s^2	1.25 m/s^2
$p_{\alpha,ref}$	0.25 rad/s	0.35 rad/s
$p_{j,ref}$	0.5 m/s^3	3.0 m/s^3
$p_{p,ref}$	1.0 m	1.0 m
$p_{stat,ref}$	0.4 m	0.35 m
p_τ	1.0 s	1.0 s
w_v	0.10	0.10
w_a	0.05	0.10
w_j	0.05	0.10
w_α	0.05	0.10
w_p	0.10	0.10
w_{stat}	1.00	1.00
w_{dyn}	1.00	1.00
w_ψ	1.00	1.00
N/t_s	40/0.2 s	60/0.1 s
nQP-Iterations	10	8

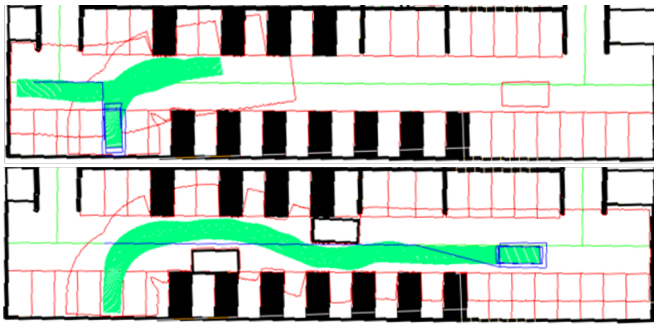

Fig. 5. Automated Valet Parking Scenario: Parking the vehicle into a spot (upper part); Pulling out of the parking spot, passing a tight passage and driving to a designated pick-up area (lower part)

The results for the backward perpendicular parking maneuver are shown in fig. 6. The reference steering profile (dashed line in the lower part of the figure) planned by the valet parking system shows a saw tooth profile. This originates from the discrete control-set used during path planning. The reference velocity is constant with only a single cusp point, at which the vehicle is supposed to change its driving direction. The actual measurements for steering wheel angle and velocity taken during the test drive (solid lines) with active trajectory optimization show a smooth result. The acceleration planned by the trajectory optimization and executed by the vehicle leads to a smooth velocity profile. It shows that the optimization follows to the given reference speed of 4 kph, which is below idle speed of the vehicle for both, forward and backward, driving, and plans a smooth stopping of the vehicle at the cusp point and at the end of the path. The steering profile shows that the trajectory optimization smooths out the steering, avoiding unnecessary steering wheel movement, which happens if

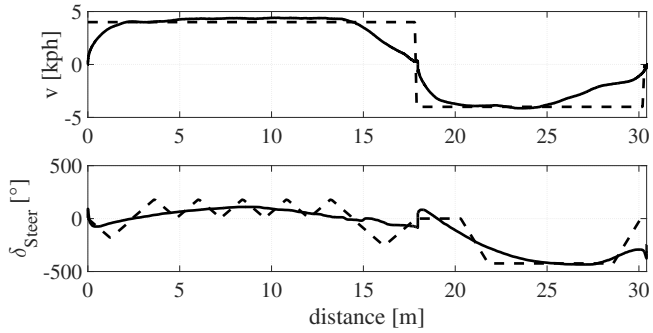

Fig. 6. Scenario Reverse Parking: Velocity (upper part) and Steering Wheel Angle (lower part) over distance driven. Solid lines correspond to the measurements with optimization and dashed lines to the reference values from the path planner

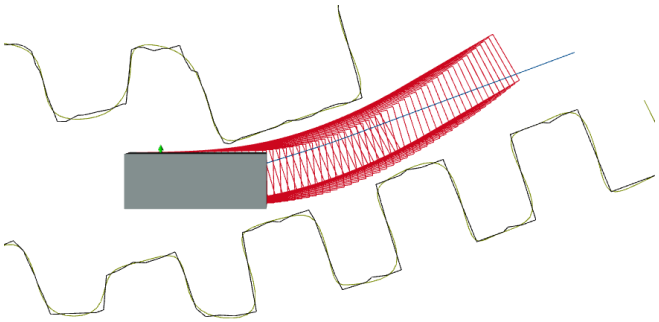

Fig. 7. Local boundaries, planned path and optimized trajectory in a narrow passage.

merely the reference trajectory is fed into the controller.

Fig. 7 shows a situation in which the vehicle had to pass a narrow passage between two static obstacles to drive into the pick-up area. Compared to fig. 5, the boundaries, the reference path and the planned trajectory are displayed in vehicle coordinates, as used in the optimization and in the vehicle controllers.

Fig. 8 shows the corresponding measurements for this scenario. As for the parking maneuver, the trajectory optimization avoided unnecessary steering and smoothened the vehicle's lateral movement. As to the longitudinal movement, the trajectory optimization slowed down the vehicle from initially 4 kph down to around 3 kph when it was passing the narrow passage. The reason for this behavior is that the cost function term c_{stat} (15) also depends on v and not just on the distance to the left and right boundary. This way, the system shows a more intuitive driving behavior, compared to just forwarding the reference speed to the controller and ignoring the narrow passage.

B. Automated Driving in Structured Environments

Fig. 9 shows the route on RWTH's Campus Melaten having a speed limit of 50 kph. The route contains a roundabout which is a challenging scenario for the combined optimization. The vehicle has to decelerate before entering the roundabout, realize a reasonable transition between the occurring longitudinal and lateral acceleration and provide

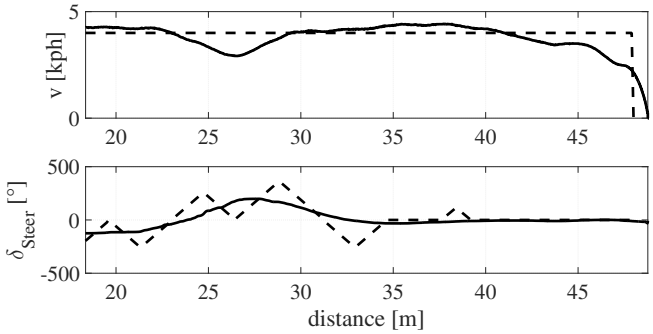

Fig. 8. Maneuvering: Velocity (upper part) and Steering Wheel Angle (lower part) over distance driven. Solid lines correspond to the measurements with optimization and dashed lines to the reference values from the path planner

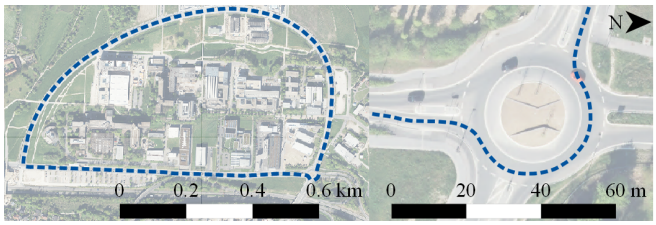

Fig. 9. Route (dashed line) around RWTH's Campus Melaten (left) and an enlarged detail of the roundabout (right).

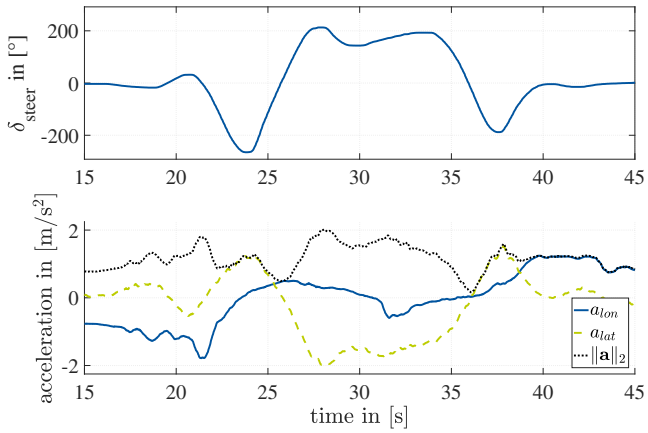

Fig. 10. Steering wheel angle and accelerations measured with the vehicle's sensors while driving through the roundabout shown in fig. 9.

a smooth steering performance while driving through the roundabout.

The resulting steering wheel angle δ_{steer} (fig. 10) shows smooth steering and the lateral acceleration a_{lat} is kept below the specified value $p_{a,lat}$ of 2.0 m/s^2. In contrast to the open loop simulations the resulting longitudinal acceleration exceeds the desired acceleration $p_{a,lon}$ of 1.25 m/s^2 especially in a negative direction (-1.9 m/s^2 at $t \approx 21.5$ s), which indicates that a further refinement of the system setup regarding the closed loop performance of the adaptive-cruise-control interface in combination with the trajectory planner is necessary.

Nevertheless, the resulting overall acceleration $\|\mathbf{a}\|_2$ shows

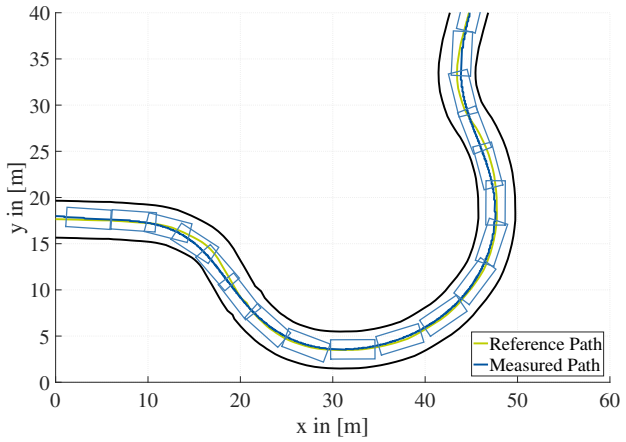

Fig. 11. Comparison between the actual reference path (green) and the measured trajectory (blue).

a smooth, jerk-optimal profile which was also confirmed by the subjective perception of passengers in the vehicle. Fig. 11 shows the reference path, the boundaries and the measured trajectory of the vehicle. As expected, the planner deliberately deviates from the reference path to provide a comfortable profile for the passenger while keeping the vehicle safely within the boundaries of the driving corridor. The measured runtimes of the trajectory optimization for both test cases were in the range of 30-50 ms.

During the on-road test drives the system encountered several situations with camera-detected lead vehicles, which were taken into account within the optimization as dynamic obstacles. Without further analysis at this point, it can be stated, that the trajectory optimization was also able to respect the specified time gap p_τ.

V. OUTLOOK

The results show, that the trajectory optimization in connection with a suitable controller is able to safely and comfortably guide the vehicle over a speed range up to 50 kph for urban scenarios as well as for narrow low-speed scenarios including reversing. A coarse reference path without special continuity requirements can be used as input for the local trajectory optimization. This means that the input from a pre-processing module, like a park planner or a behavior generation module, can utilize less complex models to generate the reference path. The optimization showed good performance in traffic situations such as vehicle-following.

In the next phase, the optimization will be evaluated in real-world drives at higher velocities, e.g. in highway scenarios and in dense traffic. Besides an evaluation of the extended speed range, a sensitivity analysis will be carried out for all parameters, inputs from the reference path as well as the perception module. In addition, the adaptability of the trajectory optimization to different driving habits will be examined by means of a real-world driving study. For this purpose manually driven trajectories will be recorded and used to fit the objective function to generate a similar behavior during automated driving.

REFERENCES

[1] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[2] M. Diehl, H. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast Motions in Biomechanics and Robotics*, M. Diehl and K. Mombaur, Eds. Springer-Verlag Berlin Heidelberg, 2006, pp. 65–93.

[3] R. Bellman, "The theory of dynamic programming," *Bulletin of the American Mathematical Society*, vol. 60, no. 6, pp. 503–515, 1954.

[4] M. Werling, "Integrated trajectory optimization," in *Handbook of Driver Assistance Systems*, H. Winner, S. Hakuli, F. Lotz, and C. Singer, Eds. Springer International Publishing, 2016, pp. 1413–1435.

[5] D. Gonzlez, J. Prez, V. Milans, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, April 2016.

[6] A. Domahidi, "Methods and tools for embedded optimization and control," Ph.D. dissertation, Eidgenssische Technische Hochschule ETH Zürich, 2013.

[7] G. Frison, H. H. B. Sørensen, B. Dammann, and J. B. Jørgensen, "High-performance small-scale solvers for linear model predictive control," in *2014 European Control Conference (ECC)*, June 2014, pp. 128–133.

[8] R. Quirynen, M. Vukov, and M. Diehl, "Multiple shooting in a microsecond," in *Multiple Shooting and Time Domain Decomposition Methods*, T. Carraro, M. Geiger, S. Körkel, and R. Rannacher, Eds. Springer International Publishing, 2015, pp. 183–201.

[9] M. Werling, S. Kammel, J. Ziegler, and L. Gröll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 346–359, 2012.

[10] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for bertha - a local, continuous method," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, June 2014, pp. 450–457.

[11] B. Gutjahr, L. Gröll, and M. Werling, "Lateral vehicle trajectory optimization using constrained linear time-varying mpc," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1586–1595, June 2017.

[12] G. Schildbach, "A new nonlinear model predictive control algorithm for vehicle path tracking," in *Advanced Vehicle Control: Proceedings of the 13th International Symposium on Advanced Vehicle Control (AVEC'16), September 13-16, 2016, Munich, Germany*. CRC Press, 2016, p. 139.

[13] H. Bock and K. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603 – 1608, 1984, 9th IFAC World Congress: A Bridge Between Control Science and Technology, Budapest, Hungary, 2-6 July 1984.

[14] J. Nocedal and S. Wright, *Numerical Optimization*, ser. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006.

[15] B. Houska, H. J. Ferreau, and M. Diehl, "An auto-generated real-time iteration algorithm for nonlinear mpc in the microsecond range," *Automatica*, vol. 47, no. 10, pp. 2279 – 2285, 2011.

[16] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

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

[18] D. Dolgov and S. Thrun, "Autonomous driving in semi-structured environments: Mapping and planning," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 3407–3414.

[19] S. Klaudt, A. Zlocki, and L. Eckstein, "A-priori map information and path planning for automated valet-parking," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 1770–1775.

[20] D. Will, S. Klaudt, E. Beizerov, and L. Eckstein, "Automated parking applications from environmental modeling to planning and tracking of a feasible trajectory," in *24th Aachen Colloquium Automobile and Engine Technology 2015*, October 2015, pp. 625–638.

[21] QGIS Development Team, "Qgis geographic information system," 2017. [Online]. Available: http://qgis.osgeo.org