2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)
Windsor Oceanico Hotel, Rio de Janeiro, Brazil, November 1-4, 2016

Hybrid Discrete-Parametric Optimization for Trajectory Planning in On-Road Driving Scenarios

Felix Kunz and Klaus Dietmayer

Abstract— **Dynamic on-road driving scenarios require robust methods for planning a safe and feasible vehicle motion coping with both static and dynamic obstacles. Many of the different approaches which have been proposed to tackle this challenge are based on optimal control and employ local continuous or discrete optimization schemes. While discrete methods possess the ability to find reasonable solutions in a search space incorporating local minima, they tend to sacrifice optimality for real-time performance. On the other hand, local continuous methods require suitable initialization to handle the combinatorial nature of on-road driving scenarios but are capable of quickly returning an optimized solution. The presented work proposes a hybrid solution which embraces both strategies to unite their distinct advantages. A discrete optimization scheme is augmented by parametric optimization to achieve both low planning times as well as robust (re-)initialization.**

I. INTRODUCTION

A. Motivation

During the past decades, considerable progress has been achieved in the field of autonomous driving. One of its fundamental tasks is planning vehicle trajectories that are not only safe but also offer a high level of driving comfort and satisfy constraints which ensure their feasibility. Asserting these properties can be challenging as planning algorithms have to cope with complex dynamic on-road scenarios comprising static and dynamic obstacles. Trajectory planning in such dynamic environments demands real-time planning capabilities to ensure sufficiently low reactiveness towards changes in the environment. The proposed trajectory planning architecture tackles this problem using a two-step optimization scheme combining the strength of discrete and parametric optimization.

B. Related Work

A variety of approaches to trajectory planning which have been developed over the last years can be categorized as sampling based methods. An advantage of many of these discrete methods is their capability to address the combinatorial aspects arising in on-road motion planning. However, the complexity of the search does require high computational cost and depending on the discretization and formulation of the optimization problem are only capable of delivering suboptimal solutions. A well-known approach which belongs to the class of incremental random sampling-based methods is the rapidly-exploring random trees (RRT) algorithm [1]. It has been designed originally for planning in unstructured environments. However, extensions have been proposed which adapt the RRT for on-road driving. An example is presented in [2] where an adaptation is proposed which employs biased sampling to enable real-time on-road planning. In contrast to random sampling, state lattices are based on deterministic discretization. State lattices have been adapted for on-road driving [3], [4] by a lattice deformation w.r.t. the course of the road. However, in order to be capable of coping with dynamic obstacles, further adaptations are necessary as presented in [5]. The consideration of dynamics and the inclusion of time lead to a large discrete search space which comes at the cost of high planning times, i.e. including the time required for the lattice generation. The authors of [6] have presented an approach based on the solution of a constrained optimization problem using a continuous optimization scheme. Compared to sampling-based and discrete methods this approach offers the advantage that the potential optimality of the solution is not restricted by discretization. However, a suitable method for initialization is required and an additional method to deal with combinatorial aspects arising in on-road driving scenarios is necessary [7]. While local continuous methods are capable of achieving low planning times, their performance closely relies on the choice of a suitable initialization and they converge to minima in the proximity of the initialization point.

Some of the proposed methods such as [8], [9], [10] combine the aforementioned strategies to improve the quality of the solution and decrease the required planning times. The authors of [8] propose a method which employs model predictive control to generate candidate trajectories reaching a set of sampled target states. An approach which combines a sampling-based method with continuous optimization is presented in [9]. Multiple optimization techniques are employed to generate reference trajectories which are used in a subsequent step for focused spatiotemporal sampling in order to generate smooth parametric trajectories. While both methods allow fast trajectory generation, the discretization and trajectory representation employed may result in suboptimality of the solutions.

The presented work aims at the combination of a discrete and continuous optimization scheme to merge the strengths of both strategies, i.e., a hybrid planning scheme which is able to achieve robust initialization and low planning times. The contribution of this work is based on an extension to the discrete method proposed in [11] which has been employed successfully in the experimental vehicle at Ulm University [12]. The method is augmented with a scheme for parametric

Felix Kunz and Klaus Dietmayer are with driveU / Institute of Measurement, Control and Microtechnology, University of Ulm, 89081 Ulm, Germany. {felix.kunz, klaus.dietmayer} at uni-ulm.de

optimization to allow a reduction of planning times while retaining the advantages of the discrete optimization scheme.

C. Overview

The remainder of this paper is organized as follows: In Section II a definition of the motion planning problem is provided. This includes the formulation of the optimization problem and the elaboration on the discrete scheme which is employed for solving the problem. In Section III, a continuous optimization scheme is introduced for the identical optimization problem and both schemes are incorporated into a hybrid optimization architecture. Simulation results and implementation detail are presented in Section IV and a conclusion is provided in Section V.

II. MOTION PLANNING

The discrete planning scheme including the formulation of the optimization problem employed in the context of this work is based on [11] where trajectories are described in relation to a given reference curve. After a brief introduction of the environment model, Frenet coordinates which allow such a relative description are explained. Then, the formulation of the optimization problem is presented and finally, the discrete optimization scheme is explained.

A. Environment Representation

The environment model which is utilized in the context of this work is depicted in Fig. 1. Lanes are represented by reference curves which are stored as polylines in a high precision digital map. Static obstacles are modeled using boundary lines which exclude any of them from the drivable area. Dynamic obstacles are represented as tracked objects. For a more detailed description of the environment model please refer to [13].

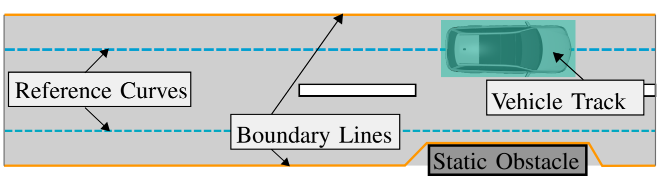

Fig. 1: Schematic depiction of the environment model: Reference curves model the course of the lanes. Boundary lines exclude all static obstacles from the free space.

B. Frenet Coordinates

A key concept employed in [11] is mimicking human driving by generating vehicle motion in relation to the course of the road. For this purpose, a reference curve $\boldsymbol{r}(l)$ which is parameterized by its arc length l is used to describe the vehicle motion using Frenet coordinates: The arc length l along the reference curve corresponds to the abscissa whereas the lateral offset to the curve is described by the ordinate d. Using the normal vector $\boldsymbol{n}(l)$ a trajectory can be described in Cartesian coordinates $\boldsymbol{x}(t) = (x(t), y(t))$ as

$$\boldsymbol{x}(t) = \boldsymbol{r}(l(t)) + \boldsymbol{n}(l(t)) \cdot d(t).$$

The transformation of a path from Frenet to Cartesian coordinates is schematically depicted in Fig. 2.

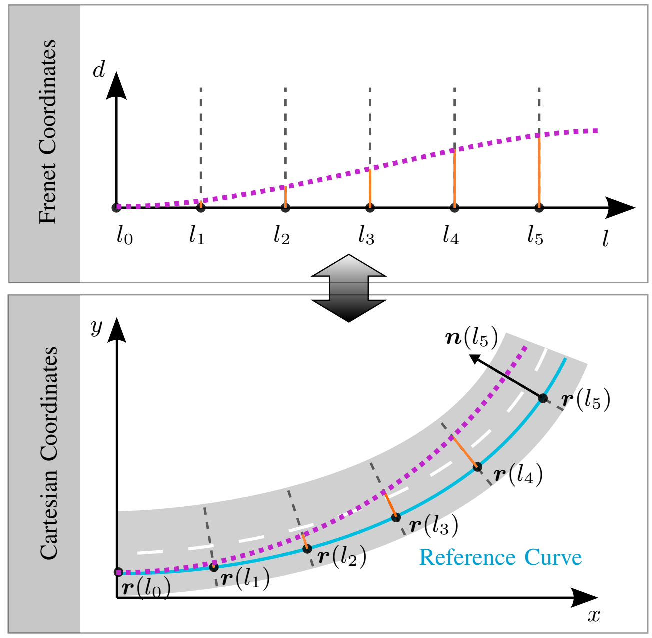

Fig. 2: Relation between Frenet and Cartesian coordinates. The dashed path is depicted in Frenet (upper-) and Cartesian coordinates (lower image) in relation to the reference curve (cyan). l describes the arc length along the reference curve $\boldsymbol{r}(l)$ and d represents the perpendicular offset.

C. Optimization Problem Definition

An important requirement for trajectory planning is the generation of a smooth and comfortable vehicle motion which is closely related to the jerk. In order to generate such a smooth motion, the time integral of the jerk $\dddot{p}(t)$ where $p(t)$ denotes a one-dimensional position over time (i.e. $l(t)$ or $d(t)$) is considered in the cost functional

$$J(p(t)) = \int_{t_0}^{t_f} \frac{1}{2} \dddot{p}^2(\tau)\, \mathrm{d}\tau + V(\boldsymbol{p}(t_f), T) \tag{1}$$

with $T = t_f - t_0$. Besides the cost for the jerk, the cost functional further contains a Mayer term $V(\boldsymbol{p}(t_f), T)$ which adds additional costs depending on the deviation of the target state $\boldsymbol{p}(t_f) = [p(t_f), \dot{p}(t_f), \ddot{p}(t_f)]^{\mathrm{T}}$ at time t_f from a desired target state and the length of the time interval T.

The cost functional in Eq. (1) is employed for both lateral and longitudinal movement w. r. t. the reference curve. The resulting total cost functional is given by

$$J(l(t), d(t)) = J(l(t)) + \kappa_{lat} \cdot J(d(t)) \tag{2}$$

where κ_{lat} represents a weight factor for the lateral costs. The feasibility of the solutions is enforced by constraints which limit permitted curvature, acceleration and velocity.

D. Discrete Optimization Scheme

It can be shown, that quintic polynomials are the function class which minimize the cost functional presented in Eq. (1), i.e. a quintic polynomial represents the jerk optimal solution for a transition from an initial state $\boldsymbol{p}(t_0) = [p(t_0), \dot{p}(t_0), \ddot{p}(t_0)]^{\mathrm{T}}$ to a target state $\boldsymbol{p}(t_f) = [p(t_f), \dot{p}(t_f), \ddot{p}(t_f)]^{\mathrm{T}}$. The key idea of the discrete optimization scheme presented in [11] is that instead of solving the problem directly regarding the constraints, a set of candidate trajectories is generated which are checked for the compliance with all constraints in a subsequent step.

The complete planning procedure comprises the following steps: First, a set of candidate trajectories for both lateral and longitudinal motion is generated in Frenet coordinates in the form of quintic polynomials. Then all of the trajectories are combined, i.e. each lateral trajectory is combined with each longitudinal trajectory. These candidate trajectories in Frenet coordinates are then transformed point-wise to Cartesian coordinates and checked whether they violate any constraint. All trajectories, which fail any of the constraint checks are discarded from the set of possible solutions. This ensures that all remaining trajectories are collision-free and feasible. Finally, the candidate trajectory which has the lowest total cost of the remaining trajectories is chosen as solution. A challenge of this method lies in the large set of candidate trajectories which have to be evaluated: A set of 100 longitudinal and 100 lateral Frenet trajectories results in 10.000 combinations. If each trajectory with a temporal length of 5 s is sampled at 20 Hz, a total number of one million trajectory states have to be tested for the violation of any constraints in each planning step. While the evaluation of kinematic constraints which assure the feasibility of the solution is relatively simple, collision checks come at higher computational cost. For this reason a special collision checking strategy has been developed which allows direct collision checks in Frenet coordinates without the need of a preceding transformation to Cartesian coordinates [14].

An advantage of the decomposition into lateral and longitudinal motion is that the desired behavior may be defined independently. The desired longitudinal or lateral behavior can be achieved through the definition of the Mayer term. E.g. the generation of a stopping maneuver can be realized by setting $V(\boldsymbol{l}(t_f), T) = \frac{1}{2}\left(l(t_f) - l_{\text{stop}}\right)^2 + \kappa_{\text{time}} \cdot T$ and restricting the sampled target states to $\dot{l}(t_f) = 0$ and $\ddot{l}(t_f) = 0$. In this manner, the longitudinal behavior for stopping, following a leading vehicle or driving at a desired velocity can be realized.

III. HYBRID OPTIMIZATION SCHEME

While the discrete optimization approach presented in the previous section offers a convenient interface for achieving a desired driving behavior it also has two drawbacks: First, any subsequent planning step is independent of the previous solution. In many cases, the solution of a subsequent planning step will be the same (or slightly altered depending on the changes in the environment). This is in contrast to many continuous approaches where effort in a subsequent optimization step may be reduced by using the previous solution for initialization. Second, the large quantity of candidate trajectories impedes the evaluation of several behavior strategies due to the computational demand. To overcome these drawbacks, a continuous optimization scheme to the problem described in the previous section is introduced.

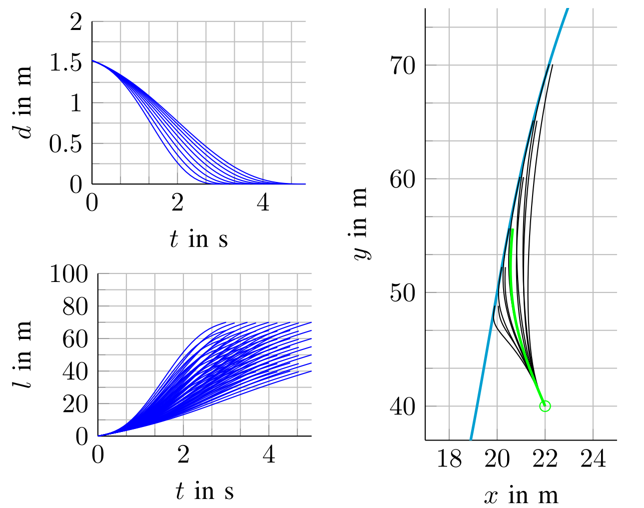

Fig. 3: Left: Set of lateral and longitudinal trajectories in Frenet coordinates. Right: Depiction of the paths resulting from the combination and transformation to Cartesian coordinates using a reference curve (cyan). The path of the trajectory with the lowest cost is colored green.

A. Continuous optimization scheme

As the following considerations address both lateral and longitudinal motion in Frenet coordinates, the position will be commonly denoted as p. Using a quintic polynomial, the motion in Frenet coordinates is described as

$$p(t) = \begin{cases} \sum_{i=0}^{5} c_{\mathrm{p},i} \cdot t^i, & , t \leq t_{\mathrm{f,p}} \\ p(t_{\mathrm{f,p}}) + \dot{p}(t_{\mathrm{f,p}})\Delta t + \frac{1}{2}\ddot{p}(t_{\mathrm{f,p}})\Delta t^2 & , t > t_{\mathrm{f,p}} \end{cases}$$

with coefficients $c_{\mathrm{p},i}$ and $\Delta t = (t - t_{\mathrm{f,p}})$ where $t_{\mathrm{f,p}}$ denotes the final time at which the target state $\boldsymbol{p}(t_{\mathrm{f,p}})$ is reached. From this time forth, the trajectory is merely extrapolated as stated above. As the initial state $\boldsymbol{p}(t_0)$ is fixed, so are the coefficients

$$c_{0,\mathrm{p}} = p(t_0), \quad c_{1,\mathrm{p}} = \dot{p}(t_0), \quad c_{2,\mathrm{p}} = \frac{1}{2}\ddot{p}(t_0).$$

The remaining coefficients $c_{3,\mathrm{p}}, c_{4,\mathrm{p}}$ and $c_{5,\mathrm{p}}$ as well as the target time $t_{\mathrm{f,p}}$ are free and therefore included as optimization variables. For this reason, taking lateral and longitudinal motion into account, there are 8 optimization variables

$$\mathbf{x}_{\text{opt}} = [c_{3,\mathrm{l}}, c_{4,\mathrm{l}}, c_{5,\mathrm{l}}, t_{\mathrm{f,l}}, c_{3,\mathrm{d}}, c_{4,\mathrm{d}}, c_{5,\mathrm{d}}, t_{\mathrm{f,d}}]^{\mathrm{T}}. \quad (3)$$

The cost function can be evaluated completely in Frenet coordinates and allows the derivation of an analytical solution for the gradient.

Analog to the discrete optimization scheme, the generation of feasible and safe solutions is assured through a set of constraints. The evaluation of these constraints requires that trajectory states are transformed to Cartesian coordinates. Kinematic constraints are added which restrict velocity, acceleration and curvature to assure the feasibility of the solutions and prohibit aggressive driving behavior. The limits and constraints are equivalent to the ones employed in the discrete scheme. However, the formulation of the constraint for collision free trajectories differs due to two reasons: First, the method for collision checks employed in the discrete scheme is tailored to cope with a large number of checks. This comes at the cost of a higher precomputational effort. Second, the calculation of the Jacobian for the constraints requires the calculation of the distance to obstacles which is not provided by the former method. For these reasons, the calculation of a pseudo distance is employed as presented in [15].

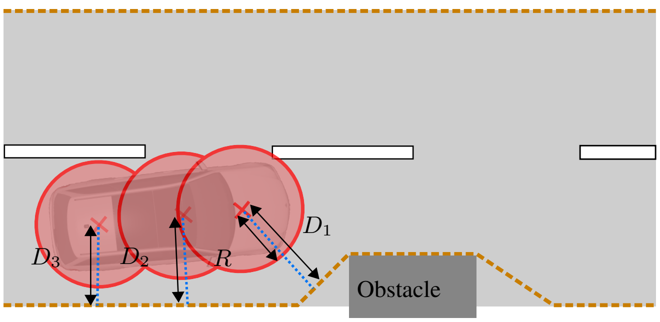

Fig. 4: Schematic illustration of collision constraints. The distance D_i to the boundary lines (orange) has to be larger than the radius R of the disks used to approximate the vehicle shape.

The vehicle shape is approximated by disks with radius R as shown in Fig. 4 and the pseudo distance D is calculated for each disk center. A vehicle pose is collision free, if

$$\min_i (D_i) > R.$$

A safety margin may be added to the radius to ensure that the vehicle does not pass obstacles too closely. The vehicle shape is computed for discrete times by calculating the vehicle position and heading angle. Moreover, the shape is stretched to account for the distance traveled within the corresponding time intervals. Dynamic obstacles are considered by treating them as static obstacles within time intervals. This is accomplished by extending their shapes to account for the motion within each interval. The resulting shapes are modeled as trapezoids as shown in Fig. 6.

The continuous method allows a fast solution of the problem. However, it also has some disadvantages compared to the discrete approach: An unfavorable initialization may lead to significantly increased planning times or even failure to converge. Moreover, the continuous approach delivers a local solution and therefore requires an additional strategy for reinitialization in order prevent it from getting stuck in a local minimum.

B. Hybrid planning architecture

In order to combine the strength of both methods and overcome their shortcomings, a hybrid planning architecture is proposed. It is important to emphasize that both planning schemes solve the same optimization problem in terms of costs which makes the solutions of both methods directly comparable. An overview of the proposed combination is presented in Fig. 5. Safeguards which assure long term safety are provided by a behavior layer and allow a limitation of the planning horizon to 5 s. This is achieved by the specification of suitable lateral and longitudinal objectives for the trajectory layer. The objectives are passed through the specification of the Mayer term (see Sec. II) and constraints. For example, the behavioral layer may confine the maximum lateral offset to prohibit trajectories which enter the adjacent lane.

An initial planning step is performed using the discrete method which does not require any specific initialization. This initial solution can then be utilized as initialization for the continuous planning method in a subsequent planning step. The continuous method uses its preceding solution in the next optimization steps. This procedure allows shorter planning cycles which increases the planner's responsiveness allowing it to react quickly to changes in the environment.

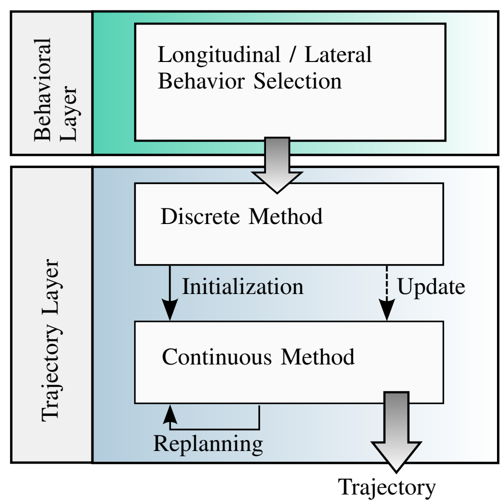

Fig. 5: The discrete methods generates a solution according to the directive specified by the behavioral layer. The solution is used for initialization and periodical updates of the faster continuous optimization scheme.

However, due to the locality of the solution, the discrete method is employed at a slower cycle time to generate updates which are used as reinitialization for the continuous method. The decision on whether to choose the trajectory generated by the discrete or continuous method is made by a comparison of the associated costs. If the costs of the output of the discrete method are lower, then it may be chosen to reinitialize the continuous planner. To increase reactivity, it may also be passed to vehicle control directly.

Another possibility which arises due to the lower planning times required by the continuous method is the parallel planning for two distinct behavior selections. The authors of [11] have proposed such an evaluation of different behavioral modes. However, merely employing the discrete method comes at significantly increased computational cost.

A secondary alternative mode such as planning an emergency maneuver can be realized in a concurrent fashion using the procedure described above.

IV. RESULTS

The presented planning algorithm has been implemented in C++. All experiments have been carried out using a 4 GHz Core i7 CPU. A first runtime evaluation of both methods has been conducted for two scenarios which both employ boundary lines with a segment length of $2\,\mathrm{m}$ including 100 segments each and dynamic obstacles. The generated trajectories have a temporal length of $5\,\mathrm{s}$ and are sampled at $50\,\mathrm{Hz}$. The discrete planning scheme requires the specification of the number of candidate trajectories which are considered in each planning step. Choosing a fine discretization allows achieving solutions, which are closer to the optimal solution. However, the evaluation of a large set of candidate trajectories also requires increased planning times which reduces reactivity of the planner to changes in the environment. A good compromise between accuracy and speed has been achieved using set sizes of 200 longitudinal and 150 lateral trajectories allowing average planning times of $97.3\,\mathrm{ms}$. A parallel computation of the constraint evaluation has been realized using OpenMP. However, the increased planning time is only required for the initialization and cyclic update of the continuous method. In contrast to the discrete method, the continuous method requires an average planning time of only $26.3\,\mathrm{ms}$ (single thread). The continuous method relies on SNOPT [16], [17] which employs a sparse sequential quadratic programming (SQP) algorithm for the solution of the optimization problem. For the employment in the hybrid approach, the maximum number of iterations of the SQP algorithm has been limited to restrict the maximum planning times. The hybrid approach executes the two optimization schemes at different frequencies: The discrete method is run at a cycle time of $150\,\mathrm{ms}$ while the continuous method is triggered every $50\,\mathrm{ms}$. These cycle times are reasonable considering the required planning times. To ensure that the

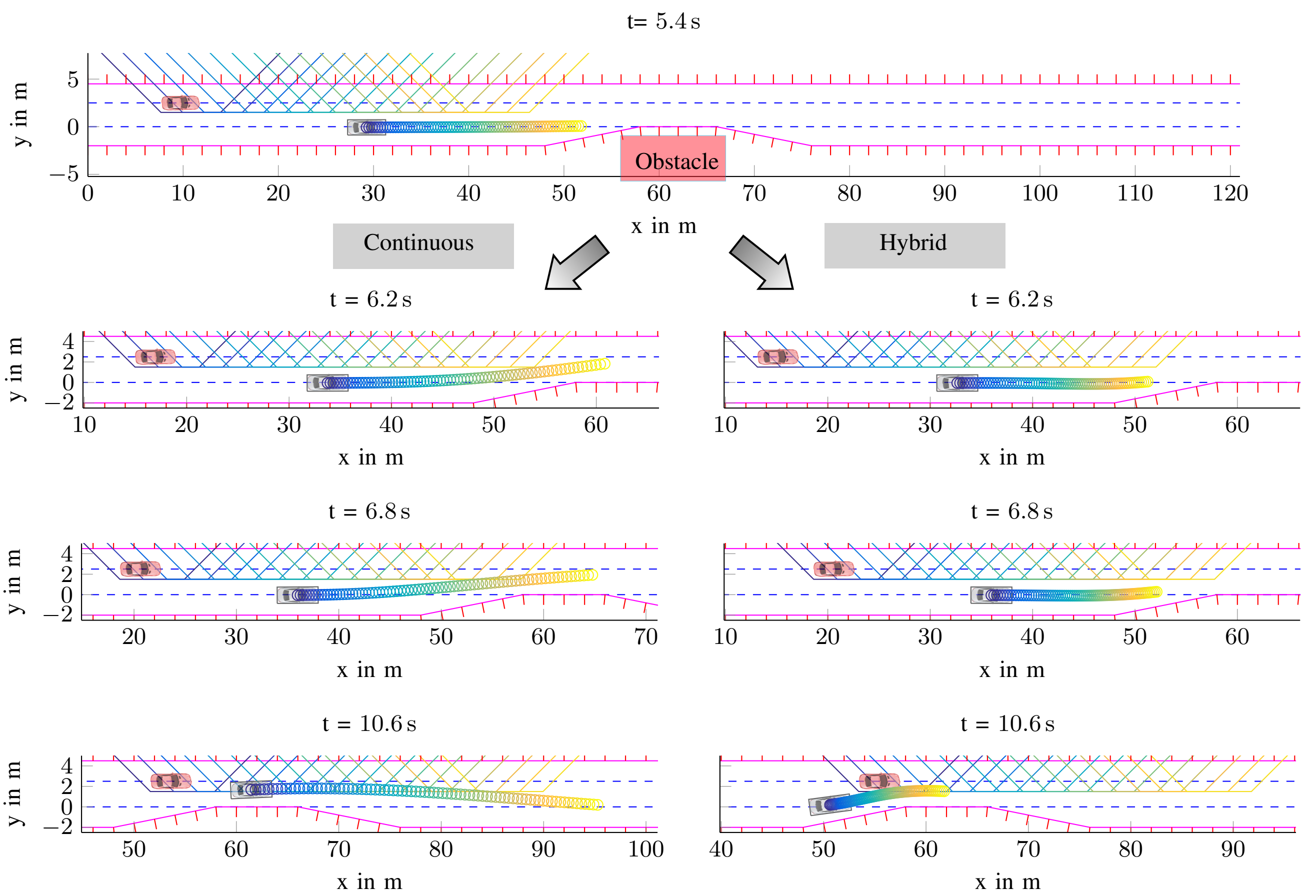

Fig. 6: Comparison of continuous (left) and hybrid (right) solution: The autonomous vehicle (grey) is forced to circumvent an obstacle while another vehicle (red) is approaching on the adjacent lane at a higher velocity. The prediction of the dynamic obstacle (trapezoids) as well as the trajectory (circles) are depicted for a duration of $4\,\mathrm{s}$. At $t = 6.2\,\mathrm{s}$, the continuous scheme is forced by the collision constraints to generate trajectories with high acceleration. In contrast, the hybrid approach is able to find the optimal solution. This leads to a safer behavior by letting the vehicle on the adjacent lane pass before circumventing the obstacle.

methods adhere to the cycle times they may be finalized prematurely as an additional safeguard.

The benefit of the combination of both solution schemes is shown exemplary in the scenario presented in Fig. 6. Initially, the autonomous vehicle is driving at a speed of $5\,\mathrm{m/s}$ on the right lane of a four-lane road. A narrowing of the lane requires the vehicle to enter the adjacent lane in order to circumnavigate the obstacle. However, another vehicle travels at a faster velocity of $9\,\mathrm{m/s}$ on the left-hand lane. At $t = 5.4\,\mathrm{s}$, the trajectory which is planned using the continuous method starts to deviate from the right reference curve. At $t = 6.8\,\mathrm{s}$, the prediction of the approaching vehicle forces the autonomous vehicle to plan a trajectory which accelerates in order to prevent a collision. The locality of the solution achieved by the continuous optimization scheme prevents the method from finding the global optimal solution. Instead, the system vehicle further accelerates to pass the narrowing which results in an aggressive driving maneuver. In contrast, the hybrid approach successfully finds the optimal solution which corresponds to letting the other vehicle pass and entering the adjacent lane afterwards. Based on the speed of the approaching vehicle and the constraint on the acceleration, the continuous method can even be incapable of finding any solution in such a situation by itself.

V. CONCLUSION

This contribution has presented a hybrid architecture for trajectory planning which employs the combination of a discrete and continuous planning scheme. The continuous, parametric optimization scheme has been introduced as an extension to the method proposed by the authors of [11]. Compared to the discrete method it allows a significant reduction of planning times. Simulation results of the proposed combination show the exploitation of the advantages of both strategies: Shorter planning times allowing an increased reactivity to changes in the environment as well as a robust reinitialization strategy for scenarios in which the continuous method gets stuck due to constraints or local minima or even fails in case of infeasibility. Moreover, the reduced planning times allow the evaluation of additional maneuvers specified by the behavioral layer such as, e.g., emergency maneuvers. Future work will focus on the adaptation of the cost parameters to further ameliorate the driving behavior. Moreover, the number of candidate trajectories which are evaluated by the discrete method can be decreased as the continuous method is used to directly enhance the results. A good compromise between the global search capability and performance will be investigated to allow for a further reduction of planning times. The robust capabilities of the proposed planning approach will be demonstrated in the experimental vehicle of Ulm University.

REFERENCES

[1] S. LaValle and J. Kuffner, J.J., "Randomized kinodynamic planning," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 1, 1999, pp. 473–479 vol.1.

[2] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, Sept 2009.

[3] M. Rufli and R. Siegwart, "On the design of deformable input- / state-lattice graphs," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 3071–3077.

[4] U. Schwesinger, M. Rufli, P. Furgale, and R. Siegwart, "A sampling-based partial motion planning framework for system-compliant navigation along a reference path," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 391–396.

[5] J. Ziegler and C. Stiller, "Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, pp. 1879–1884.

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

[7] P. Bender, Ö. S. Tas, J. Ziegler, and C. Stiller, "The combinatorial aspect of motion planning: Maneuver variants in structured environments," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, June 2015, pp. 1386–1392.

[8] T. Howard, C. Green, D. Ferguson , and A. Kelly , "State space sampling of feasible motions for high-performance mobile robot navigation in complex environments," *Journal of Field Robotics*, vol. 25, no. 6-7, pp. 325–345, June 2008.

[9] T. Gu, J. Snider, J. M. Dolan, and J. w. Lee, "Focused trajectory planning for autonomous on-road driving," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 547–552.

[10] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," in *Proceedings of the First International Symposium on Search Techniques in Artificial Intelligence and Robotics (STAIR-08)*. Chicago, USA: AAAI, June 2008.

[11] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenét frame," in *IEEE International Conference on Robotics and Automation*, 5 2010, pp. 987–993.

[12] F. Kunz, D. Nuss, J. Wiest, H. Deusch, S. Reuter, F. Gritschneder, A. Scheel, M. Stuebler, M. Bach, P. Hatzelmann, C. Wild, and K. Dietmayer, "Autonomous driving at ulm university: A modular, robust, and sensor-independent fusion approach," in *Intelligent Vehicles Symposium*, 2015, pp. 666–673.

[13] D. Nuss, M. Stuebler, and K. Dietmayer, "Consistent environmental modeling by use of occupancy grid maps, digital road maps, and multi-object tracking," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, June 2014, pp. 1371–1377.

[14] F. Kunz and K. Dietmayer, "Fast collision checking with a frenet obstacle grid for motion planning," in *10. Uni-DAS e.V. Workshop Fahrerassistenzsysteme*, 2015, pp. 95–104.

[15] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. Keller, E. Kaus, R. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M. Enzweiler, C. Knöppel, J. Hipp, M. Haueis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein, and E. Zeeb, "Making bertha drive - an autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[16] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Rev.*, vol. 47, pp. 99–131, 2005.

[17] P. E. Gill, W. Murray, M. A. Saunders, and E. Wong, "User's guide for SNOPT 7.4: Software for large-scale nonlinear programming," Department of Mathematics, University of California, San Diego, La Jolla, CA, Center for Computational Mathematics Report CCoM 15-3, 2015.