2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

An Ad-hoc Sampling-based Planner for On-road Automated Driving

Christian Lienke[1], Martin Keller[2], Karl-Heinz Glander[2] and Torsten Bertram[1]

Abstract— **The paper at hand proposes a real-time capable approach to trajectory planning. An online sampling strategy is chosen, exploiting the structure of the surrounding environment. Lateral states are sampled from state space, whereas longitudinal states are generated via sampling from the action space. The combination yields breakpoints, which are then used to generate a candidate trajectory via spline interpolation. A bi-level candidate evaluation strategy is presented assessing comfort and human-like driving as well as a post-check of collision avoidance with accurate geometric modeling. The result is a reactive feedback motion planner, which shows promising results with respect to on-road automated driving.**

I. Introduction

A. Motivation

With the increasing interest on self driving vehicle functions, a demand for reliable and fast algorithms for motion planning arises. A major challenge represents the increasing number of traffic participants as well as the comprehensive cognitive tasks, which have to be mastered in highly complex traffic scenarios. In the architecture of automated vehicles, beside perception and control the decision and motion planning part clearly affect the performance of the ego vehicle. Evolving trajectory planning algorithms reflect the complexity of the task of automated driving. The challenge regarding the motion planning task is to find a comfortable and moreover feasible and collision free trajectory in a limited amount of time, guaranteeing real-time performance in a dynamic environment. In the past years several approaches to trajectory planning have been developed. An overview about motion planning techniques for automated vehicles is given in [1]. Regarding the requirements of motion planning algorithms and the characteristics of sampling-based trajectory planning algorithms it is not surprising that sampling-based planners gained more and more interest in research and application.

B. Related Work

Sampling-based approaches often make use of simplifications of the motion planning problem to meet requirements like e.g. the limited computation time. The main goal is to cover a continuous space with a finite number of samples. For structured environments a lane adapted reparametrisation of the work space is proposed in [2], wherein a spatio-temporal state lattice is generated with quintic polynomials. The authors of [3] also use polynomial trajectories and present a maneuver-based adaption of the cost function, assuming a high level behavioral layer. Like mentioned in [4], a drawback of this is that inconsistencies in behavioral and trajectory planning may lead to instability in planned motions. Furthermore the authors address the problem of dimensionality resulting from the combinatorial aspect of sampling-based methods and propose an efficient pruning of the search space. Instead of sampling in state space the tentacles approach [5] samples in action space, assuming continuous curvature. Likewise, the authors of [6] present a model predictive approach, which fixes the control input within the entire prediction horizon. Another approach is presented in [7], wherein sample states are connected via forward simulation of a simple vehicle model in combination with a feedback control. In [8], based on the predicted trajectories of obstacle vehicles, a decomposition of the current scene in distinctive areas is made. Thus points are generated that the trajectory has to pass through, satisfying vehicle dynamics at the same time. The aforementioned approaches address at least one of the aspects that have to be solved within a sampling-based trajectory planner. This is the choice of the utilized architecture, the applied sampling strategy and the description of vehicle motion. Within this work a sample-based planner for on-road automated driving, which is able to efficiently generate convenient trajectories for consistent maneuver decisions is developed. Real time capability is achieved by online sampling, generating a meaningful set of trajectory candidates starting from the current vehicle state. Sample states are generated for longitudinal and lateral quantities, sampling from action and state space, respectively. By this the advantages of both ideas are combined, as sampling the longitudinal acceleration ensures feasibility, whereas lateral sampling with respect to the detected lanes exploits the structure of the environment, capturing high-level driving behavior. Note that desired longitudinal behavior is not neglected, as a target state is considered. To model vehicle motion a spline-based formulation is chosen to enable a large prediction horizon without loss of performance in model accuracy and run time. Via interpolation a set of combined lateral and longitudinal candidate trajectories is generated. The evaluation strategy is designed to find the best trajectory in dependence on the current situation. The decision whether to conduct a respective maneuver is subject to the candidate evaluation function. Because of the close connection between behavior and trajectory planning the objective function has to be designed comprehensively to enable a reliable decision of the automated vehicle. The architecture of the proposed approach is illustrated in Fig. 1.

[1]Christian Lienke and Torsten Bertram are with the Institute of Control Theory and Systems Engineering, TU Dortmund University, 44227 Dortmund, Germany, https://orcid.org/0000-0002-5952-9246, christian.lienke@tu-dortmund.de

[2]Martin Keller and Karl-Heinz Glander are with Active & Passive Safety Technology, ZF Group, 40547 Düsseldorf, Germany

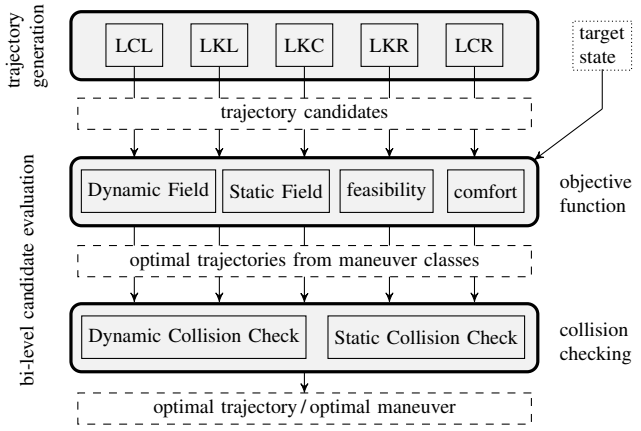

Fig. 1: Architecture of the developed approach. Generated trajectories are based on lateral maneuver classes lane change left (LCL), lane keeping left (LKL), lane keeping center (LKC), lane keeping right (LKR) and lane change right (LCR). The proposed sampling strategy still enables independent longitudinal behavior. Desired high level behavior is represented by a given target state. Nevertheless the decision is made at the trajectory planning level via the bi-level candidate evaluation strategy.

This paper is organized as follows. The underlying trajectory representation is introduced in Section II. The two major parts of the sampling-based approach are then discussed in detail. Section III presents the developed sampling strategy for trajectory generation and Section IV explains the utilized candidate evaluation. The results of the proposed approach are discussed in Section V. Section VI concludes the paper.

II. Sampling-Based Trajectory Planning

The proposed method is designed to generate sample states for breakpoints $\underline{\mathbf{x}}_\iota$ with respect to time instances t_ι. In this section the underlying representation of candidate trajectories is introduced. In the following the earth, vehicle and curvilinear coordinate frame indicated by leading superscripts are denoted E, F and K, respectively.

A. Spline-based Trajectory Planning

To interpolate between sampled states a spline in ${}^F x(t)$ and ${}^F y(t)$ is determined, where the interpolant is defined by

$$\mathbf{x}(t) = \begin{cases} \mathbf{s}_\iota(t) & t_\iota \leq t \leq t_{\iota+1} \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

For $\eta - 1$ spline segments and $t \in [t_\iota \; t_{\iota+1}]$ spline $\mathbf{s}$ is

$$\mathbf{s}_\iota(t) = \mathbf{c}_{\nu,\iota}(t-t_\iota)^\nu + \mathbf{c}_{\nu-1,\iota}(t-t_\iota)^{\nu-1} + \cdots + \mathbf{c}_{1,\iota}(t-t_\iota) + \mathbf{c}_{0,\iota}. \tag{2}$$

The properties of the spline require that the spline passes through breakpoints $\mathbf{s}_\iota(t_\iota) = \underline{\mathbf{x}}_\iota$ and that continuity between each spline interval is satisfied. With start and end conditions for $\underline{\mathbf{x}}_\iota = [{}^F x_\iota, {}^F y_\iota]^T$ and the respective derivatives spline coefficients $\mathbf{c}_{0\ldots\nu,\iota}$ can be determined analytically. The $\iota = 1 \ldots \eta$ breakpoints of each trajectory, denoted with the underline, are given by

$$B = \{\underline{\mathbf{x}}_1, \underline{\dot{\mathbf{x}}}_1, \ldots, \underline{\mathbf{x}}_1^{(\mathcal{V})}, \underline{t}_1, \ldots, \underline{\mathbf{x}}_\eta, \underline{\dot{\mathbf{x}}}_\eta, \ldots, \underline{\mathbf{x}}_\eta^{(\mathcal{V})}, \underline{T}_p\}, \tag{3}$$

with $\mathcal{V} = (\nu - 1)/2$. To determine the optimum time for the transition from one interval to the next, time instances $\underline{t}_\iota$ are part of the sampling. The continuous spline-based formulation allows the choice of resolution in time ΔT independently of breakpoints ι and fixed prediction horizon T_p. Thus, by interpolation $n = T_p/\Delta T + 1$ trajectory points $\mathbf{x}_k = [{}^F x_k, {}^F y_k]^{\mathrm{T}}$, are gained yielding ego vehicle trajectory

$$\mathbf{x}_{ego} = [\mathbf{x}_1, t_1, \mathbf{x}_2, t_2, \ldots, \mathbf{x}_k, t_k, \ldots, \mathbf{x}_n, T_p]^{\mathrm{T}} \tag{4}$$

with constant time intervals $t \in [t_k \; t_{k+1}]$ for $k = 1 \ldots n$.

III. Sampling Strategy

The proposed strategy utilizes a spline formulation in combination with the application of curvilinear transformation. Lateral and longitudinal position ${}^F x(t)$ and ${}^F y(t)$ of breakpoints $\underline{\mathbf{x}}_\iota$ are supposed to be sampled in curvilinear coordinates. Lane markers serve as reference for transformation in the vehicle frame. By means of this transformation the proposed sampling-based trajectory planner is independent of road curvature and works in straight as well as in curved lane scenarios. Transformed breakpoints are then connected via spline-based interpolation, as described in Section II, resulting in a set of candidate trajectories, which are evaluated in vehicle coordinates. For this reason and in contrast to the evaluation in curvilinear coordinates, no time consuming point wise transformation of obstacle trajectories is needed. Moreover transforming only the breakpoints with interpolation and evaluation in vehicle frame permits that not every trajectory point has to be transformed. For accuracy reasons the resolution of breakpoints could be chosen according to the road curvature, minimizing the deviation to the reference lane caused by interpolation.

A. Lateral Sampling for Structured Environments

The way of sampling the lateral position strongly exploits the structure of the surrounding environment. A set of lateral maneuver classes is defined based on the outcome of high-level behavior planning. Chosen maneuver classes are lane change left (LCL), lane keeping left (LKL), lane keeping center (LKC), lane keeping right (LKR) and lane change right (LCR), accounting for a higher variability within the ego lane and enabling for example gap approaching in complex merging maneuvers. The use of curvilinear coordinates

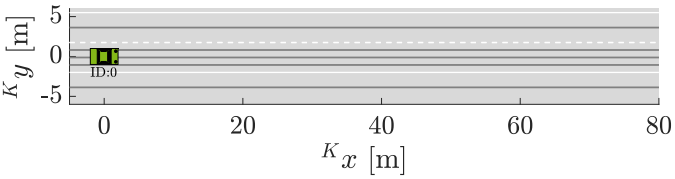

Fig. 2: Lateral reference paths in curvilinear coordinates for five modeled maneuver classes.

K rigorously simplifies the sampling for the lateral position, as each maneuver class is directly linked to a fixed value for ${}^K y(t)$. The corresponding reference paths are illustrated in Fig. 2.

B. Longitudinal Sampling using Adaptive Discretization

Due to limited computational resources a finite but meaningful set of trajectory candidates has to be generated. This motivates the application of an action-based sampling strategy for sampling longitudinal quantities such as position ${}^{K}x(t)$ and velocity ${}^{K}v(t)$. The longitudinal acceleration ${}^{K}a_x(t)$ is assumed as the control input. A second order model is utilized to generate the longitudinal samples for position ${}^{K}x(t)$ and velocity ${}^{K}v(t)$ with respect to the curvilinear frame K.

$$\begin{bmatrix} {}^{K}\dot{x}(t) \\ {}^{K}v(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} {}^{K}x(t) \\ {}^{K}v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} {}^{K}a_x(t) \tag{5}$$

Now a finite set of control inputs ${}^{K}a_x \in \mathcal{U}$ has to be found. Assuming that the optimal control input for the current planning step is close to the one of the last planning cycle a finer resolution around the actual longitudinal acceleration is desired. Nevertheless minimum and maximum values should always be considered. From these two aspect an adaptive discretization strategy is derived, analogous to [9]. Two parabolas f_1 and f_2 are utilized to transform the linearly discretized values z online and in dependence on actual value z_{act}. Coefficients are therefore determined from conditions

$$f_1(z_{\min}) = z_{\min}, \quad f_1(\bar{z}) = z_{\text{act}}, \quad f_1'(\bar{z}) = 0 \tag{6}$$

$$f_2(z_{\max}) = z_{\max}, \quad f_2(\bar{z}) = z_{\text{act}}, \quad f_2'(\bar{z}) = 0 \tag{7}$$

with $\bar{z} = (z_{\max} - z_{\min})/2$. Then the adaptive value $f(z)$ is

$$f(z) = \begin{cases} f_1(z) & z < (z_{\max} - \bar{z}) \\ f_2(z) & z >= (z_{\max} - (\bar{z}) \end{cases}. \tag{8}$$

The result of the proposed transformation yielding the adaptive discretization is shown in Fig. 3.

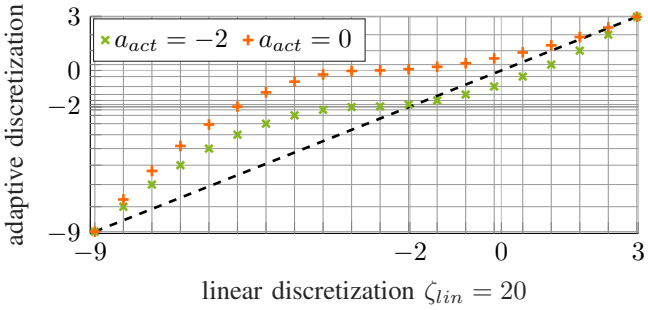

Fig. 3: Nonlinear mapping from linear to adaptive discretization, exemplarily shown for actual values $a_{act} = -2$ and $a_{act} = 0$. The minor grid lines show the desired higher resolution around the actual value for the adaptive discretization with respect to $a_{act} = -2$.

The transformation is adapted in each planning cycle in dependence on the actual acceleration of the ego vehicle. The benefit arising from adaptive discretization is shown by evaluating a highway entrance scenario. This scenario contains varying longitudinal accelerations, necessary to successfully master the dynamic traffic. For comparison a reference trajectory is generated using a linear discretization with quasi-continuous values. Thus the optimal solution with respect to the objective function is approximated. The relative costs $\Delta\Phi$ are calculated as the difference from the open-loop results between the reference Φ_{opt} and the (sub-)optimal solution Φ of the approach with the respective longitudinal sampling strategy. Four variants have been tested, comparing linear and adaptive discretization with respect to the number of sampled longitudinal accelerations $\zeta = 5$ and $\zeta = 10$. The results are shown in Fig. 4. Therein the advantage of the adaptive method is clearly visible. Comparing the adaptive sampling strategies, the effect of a decreased number of outliers can be explained by discretizing around the last optimal acceleration value in the next planning cycle. Note, the reference is still just an approximation of the optimal solution enabling relative costs less than zero. Obviously

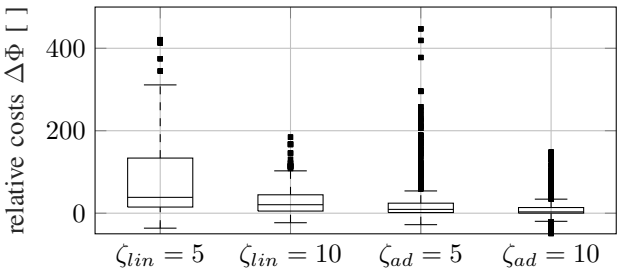

Fig. 4: Boxplot of relative costs in comparison to linear discretization with approximated quasi-continuous values as reference.

the changes in the dynamic environment are slower than the computation time for one planning cycle such that the algorithm has no problem to adapt the transformation of the acceleration, capturing the influence of the current situation on the behavior of the ego vehicle.

C. Combined Lateral and Longitudinal Trajectories

The sampling-based planner is designed for simultaneous planning of longitudinal and lateral motion. Therefore lateral and longitudinal sample states are transformed and combined, generating the candidate trajectories via spline interpolation in a next step.

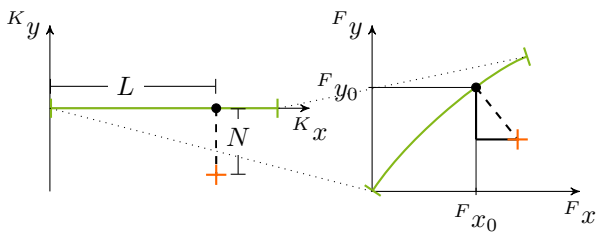

Fig. 5: Fast transformation from curvilinear frame K to vehicle coordinates F, using a look-up table and calculating the arc length by numerical integration of the reference polynomial. Evaluating the look-up table for ${}^{K}x = L$ yields ${}^{F}x_0$ and from the reference polynomial ${}^{F}y_0$ is calculated. With orthogonal distance ${}^{K}y = N$ desired coordinates in vehicle frame are then given by simple trigonometry.

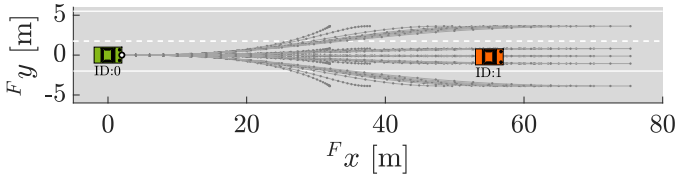

Fig. 6: Combined lateral and longitudinal candidate trajectory set for the ego vehicle in a straight lane scenario.

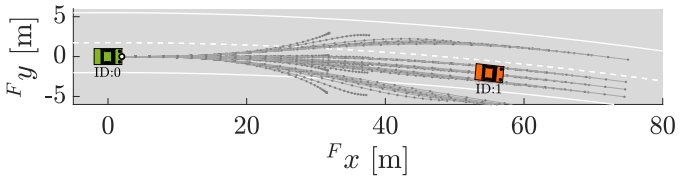

Fig. 7: Combined lateral and longitudinal candidate trajectory set for the ego vehicle in a curved lane scenario.

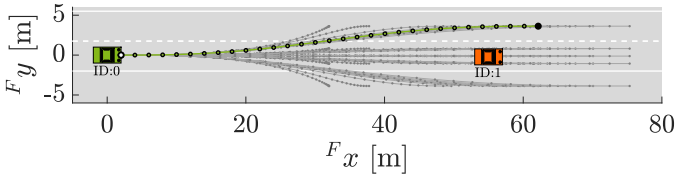

Fig. 8: Evaluation of combined lateral and longitudinal trajectories in a straight lane scenario.

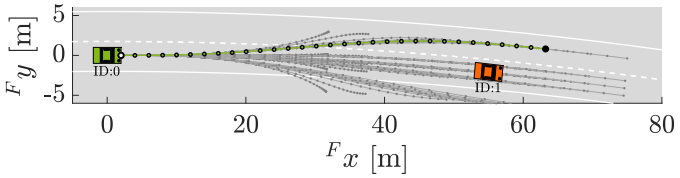

Fig. 9: Evaluation of combined lateral and longitudinal trajectories in a curved lane scenario.

To transform from curvilinear to vehicle coordinates a reference lane marker detected by the ego vehicle's camera system is chosen. The reference lane marker is defined as a cubic polynomial f_{ref}. To calculate the arc length the definite integral is approximated and the trapezoidal rule is applied. Thus a look-up table is created for equally spaced queries within the range of the detected lane marker ${}^F x_i \in [x_s, x_e]$, leading to a mapping from ${}^K x$ to the initial solution of the transform ${}^F x_0$. The initial value for the lateral part is given by ${}^F y_0 = f_{\text{ref}}({}^F x_0)$ and with orthogonal slope $m_\perp = -1/f'_{\text{ref}}({}^F x)$ the transformed position yields

$$ {}^F x = {}^F x_0 + {}^K y \cdot 1/\sqrt{1+m_\perp^2} \tag{9} $$

$$ {}^F y = {}^F y_0 + {}^K y \cdot m_\perp/\sqrt{1+m_\perp^2} \tag{10} $$

The transformation from curvilinear to vehicle coordinates is visualized in Fig. 5. After the position has been transformed to vehicle coordinates, breakpoints are assembled and a set of trajectories is generated by spline interpolation (see Section II). The resulting set of candidate trajectories $\mathcal{C}$ is shown at the example of a scenario with one static obstacle for a straight (Fig. 6) and curved lane scenario (Fig. 7). Trajectories that end unaligned to the road indicate a candidate at which the ego vehicle comes to a full stop with $v = 0\,\text{km/h}$.

IV. Fast Candidate Evaluation

A key aspect of sampling-based trajectory planning is the accurate and fast evaluation of candidate trajectories. In this contribution a bi-level approach is favored, which combines the aspects of comfortable, human-like driving and accurate collision checking (see Fig. 1). The reason for this is twofold, as the trajectory should reflect human driving behavior on one side and ensure safe and precise maneuvering on the other. Candidate trajectories of each maneuver class are evaluated by a comprehensive objective function, modeling physically inspired safety distances for dynamic collision avoidance, taking as well static collision avoidance, feasibility and comfort objectives into account. Thus the optimal trajectory for each maneuver class is chosen. For safety reasons a post-check for each selected optimal maneuver trajectory with an accurate geometric representation of the vehicles is performed in a next step. In case the post-check fails any backup strategy can be carried out. The globally optimal trajectory is found by comparing the best trajectories of each maneuver class via the cost function as described in equation (11), where the optimal trajectory simultaneously represents the best maneuver class.

$$ \mathbf{x}^*_{ego} = \arg\min_{\mathbf{x}_{ego}\in\mathcal{C}} \; (\omega_c b_d + \omega_c b_s + \Phi), \tag{11} $$

with costs Φ of the objective function and

$$ b_d = \begin{cases} 1 & \text{if dynamic collision} \\ 0 & \text{otherwise} \end{cases}, b_s = \begin{cases} 1 & \text{if static collision} \\ 0 & \text{otherwise} \end{cases} \tag{12} $$

with weight ω_c for accurate collision checking. Fig. 8 and Fig. 9 show the result of the evaluation for the straight and the curved lane scenario, respectively.

A. Objective Function

For evaluation a holistic comprehensive objective function is defined. Terms of the objective function capture dynamic and static collision avoidance (see [10] for the utilized environment model), feasibility and comfort. The cost function is defined as

$$ \Phi = \mathbf{e}^{\mathrm{T}} \mathbf{\Omega} \mathbf{e}, \tag{13} $$

with weight matrix $\mathbf{\Omega} \in \mathbb{R}^{d_{err}\times d_{err}}$ and error vector

$$ \mathbf{e} = [\mathbf{o}_1, \mathbf{o}_2, \dots \mathbf{o}_z, \chi(\mathbf{h}_1), \chi(\mathbf{h}_2), \dots, \chi(\mathbf{h}_{n-1})]^{\mathrm{T}} \tag{14} $$

with objectives $\mathbf{o}_i$ and inequality constraints considered by

$$ \chi(\mathbf{h}_k) = \max\{[\mathbf{0}, \mathbf{h}_k]\}. \tag{15} $$

Thus constraints imposed by the static and dynamic environment, as well as constraints on the vehicle dynamics like maximum acceleration defined by Kamm's circle and non-holonomic vehicle behavior are captured as penalty functions. The deviation from the desired target state provided by a high level behavior planner is formulated as an objective. To account for comfort objectives $\mathbf{o}_i$ lateral and longitudinal acceleration and lateral and longitudinal jerk are minimized.

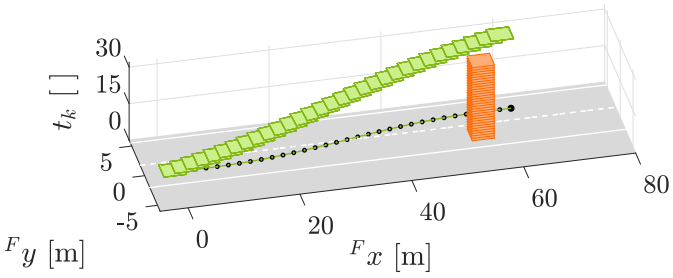

Fig. 10: Accurate dynamic collision checking in space-time domain using the separating axis theorem. The ego and obstacle vehicles are modeled with a rectangular shape. The ego and obstacle vehicle trajectory are then checked for intersections at each instant in time t_k.

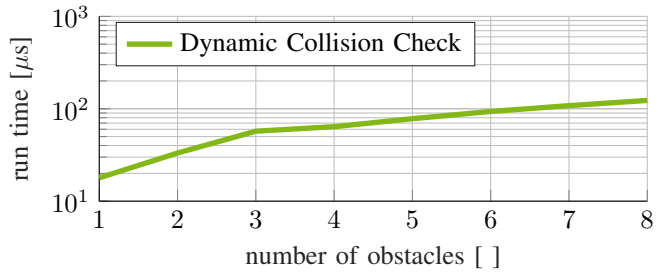

Fig. 11: Mean run time for accurate dynamic collision checking using bounding boxes in dependence on the number of obstacles for 10000 test runs with trajectories of 31 trajectory points.

B. Highly Accurate Collision Checking

To guarantee collision avoidance of the candidate trajectories, an accurate geometric modeling of the vehicles is essential. In [11] vehicle shapes are modeled with multiple circles. Intuitively a more accurate representation of the rectangular shape of vehicles is by means of a bounding box. In this contribution bounding boxes are favored to precisely match the ego and obstacle vehicles shape. For dynamic collision checking the accurate modeling comes at the cost of higher computational burden. Strategies have been developed to overcome this problem. The authors of [12] for example, introduce a hierarchical pruning of possible collisions, reducing the number of accurate collision checks. In contrast multiple interference tests, check each ego candidate trajectory point-wise against all obstacle trajectories. This corresponds to the worst case, which must always be considered to be calculated in feasible time. To account for the dynamic environment collision checking is performed in space-time domain. The Separating Axis Theorem is utilized to check for intersections between two rectangles in time. An example is illustrated in Fig. 10. Note that the reliability of the performed collision check is subject to the resolution of the trajectory. The runtime of the dynamic collision checking is depicted in Fig. 11. Static collision checking, to detect a collision with road boundaries, is performed by calculating the closest distance between the ego vehicle's bounding box and the respective lane marker.

V. Results

To show the general functioning of the developed algorithm simulations are performed with the help of a realistic dynamic environment simulation [13]. The simulated ego vehicle is equipped with radar sensors to the front, rear and sides and a front facing camera. Thus information about the static and dynamic environment is provided. The reference lane marker for the transformation from curvilinear to vehicle frame is chosen to be the one with the largest view range, purely relying on camera data. Hence the inclusion of map data would further enhance the quality of results. The algorithm works likewise receding horizon control as the environment information used for trajectory generation and collision avoidance is updated cyclically. The mean run time for the algorithm on a single core of an Intel Core i5@3.30GHz with 6MB cache is about 30 ms for a set of 50 trajectories, which are generated from 10 sampled accelerations for each lateral maneuver class. For trajectory setup interpolation order is chosen to be $\nu = 5$ with a prediction horizon of $T_p = 3$ s. Trajectories for LKC are designed as a spline with $\eta = 3$ breakpoints and an intermediate time of $t_{\iota=2} = 1.5$ s, whereas for the other maneuver classes the number of breakpoints is $\eta = 2$. A sequence of a complex highway entrance scenario is shown in Fig. 13. In the first phase the ego vehicle follows its slower leading vehicle (ID:7) on the curved entrance lane. Then the ego vehicle merges on the highway. After letting faster obstacle vehicle (ID:10) pass on the left the ego vehicle accelerates to reach a desired speed of 90 km/h and initiates an overtaking maneuver by performing a lane change to the left. The velocity of the ego vehicle is shown in Fig. 12. The interval from 8 s to 26 s is highlighted, indicating the correspondence to the results shown in Fig. 13. The investigated scenario

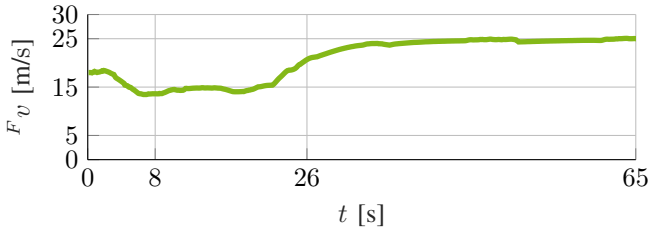

Fig. 12: Velocity of the ego vehicle during the simulated highway entrance scenario. The interval between 8 s and 26 s matches the depicted results of Fig. 13.

shows the advantages of the proposed approach, as safe and comfortable maneuvers are performed. Moreover, the ego vehicle successfully adapts its behavior in accordance to the current traffic situation, resulting in sophisticated vehicle maneuvers. Still, in this context the impact of the prediction horizon can not be neglected.

VI. Conclusion

This contribution presents a sampling-based trajectory planning algorithm that performs online generation and evaluation of suitable trajectory candidates. Lateral sampling is based on lateral maneuver classes exploiting the lane discrete

structure for on-road automated driving. In contrast to the sampling in state space for lateral quantities, longitudinal samples are generated from action space, assuming longitudinal acceleration as input. An adaptive discretization is used to increase sampling efficiency. The spline formulation and the application of curvilinear coordinates prove to be beneficial, as the proposed sampling-based trajectory planner is real-time capable and works in straight as well as in curved lane scenarios. The candidate evaluation strategy ensures collision avoidance with trajectories being evaluated comprehensively additionally taking feasibility and comfort into account. The results show that the ego vehicle is able to perform for example vehicle following and merging, which goes even beyond specified lateral maneuvers for trajectory generation. Further work will deal with improvements regarding the target state generation. Moreover parallelization would significantly speed up the process of trajectory generation and evaluation.

REFERENCES

[1] D. Gonzalez, J. Perez, V. Milanes, 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, 2016.

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

[3] 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 (ICRA)*, 2010, pp. 987–993.

[4] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 4889–4895.

[5] F. v. Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche, "Driving with tentacles: Integral structures for sensing and motion," *Journal of Field Robotics*, vol. 25, no. 9, pp. 640–673, 2008.

[6] M. Keller, C. Hass, A. Seewald, and T. Bertram, "A model predictive approach to emergency maneuvers in critical traffic situations," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2015, pp. 369–374.

[7] 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 *IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 391–396.

[8] J. Schlechtriemen, K. P. Wabersich, and K.-D. Kuhnert, "Wiggling through complex traffic: Planning trajectories constrained by predictions," in *IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 1293–1300.

[9] A. Makarow, M. Keller, C. Rösmann, and T. Bertram, "Model predictive trajectory set control with adaptive input domain discretization," in *Proceedings of the 2018 American Control Conference*, 2018, pp. 3159–3164.

[10] C. Lienke, M. Keller, K.-H. Glander, and T. Bertram, "Environment modeling for the application in optimization-based trajectory planning," in *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 498–503.

[11] J. Ziegler and C. Stiller, "Fast collision checking for intelligent vehicle motion planning," in *IEEE Intelligent Vehicles Symposium (IV)*, 2010, pp. 518–522.

[12] D. Ferguson, M. Darms, C. Urmson, and S. Kolski, "Detection, prediction, and avoidance of dynamic obstacles in urban environments," in *IEEE Intelligent Vehicles Symposium (IV)*, 2008, pp. 1149–1154.

[13] C. Wissing, T. Nattermann, K.-H. Glander, A. Seewald, and T. Bertram, "Environment simulation for the development, evaluation and verification of underlying algorithms for automated driving," in *AmE - Automotive meets Electronics 7th GMM-Symposium*, 2016, pp. 9–14.

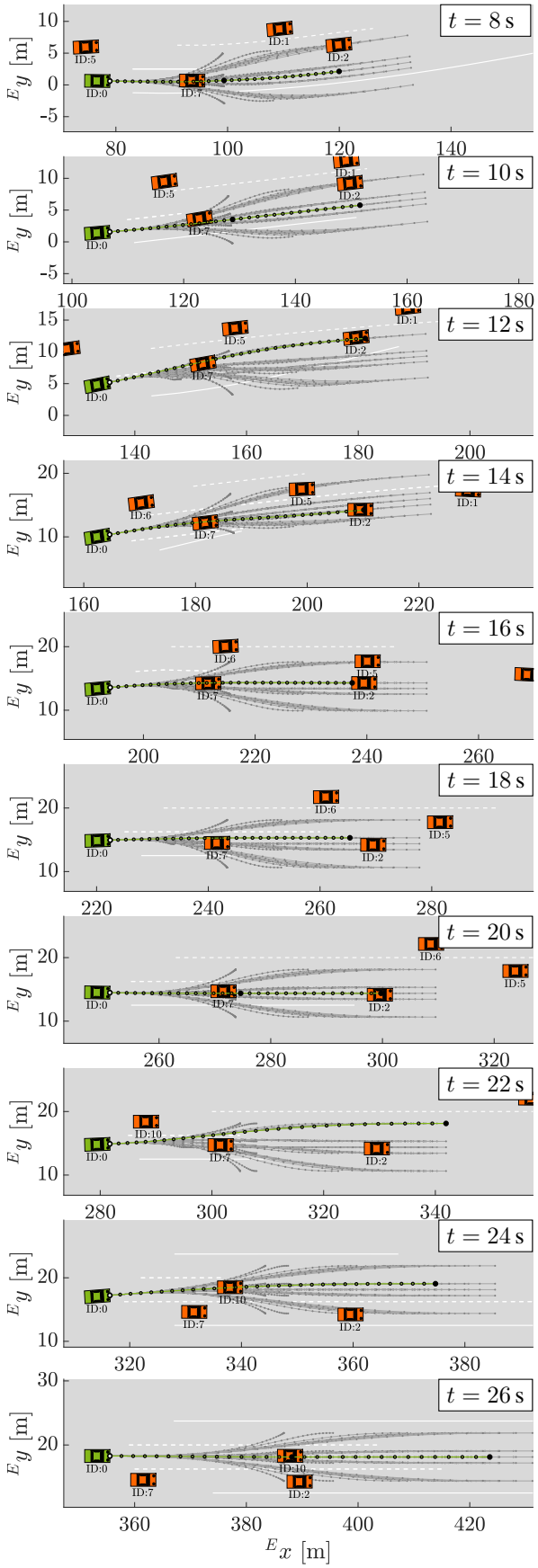

Fig. 13: Result for a complex highway entrance scenario, shown in steps of 2 s for a sequence between 8 s and 26 s.