Trajectory Planning for Automated Vehicles in Overtaking Scenarios

Maximilian Graf[1] Oliver Speidel[1] Klaus Dietmayer[1]

***Abstract*—Overtaking is a challenging task in the field of autonomous driving, especially on roads with an opposite lane and oncoming vehicles. Since trajectory planning is repeated cyclic it is highly important to trigger the maneuver only if it is guaranteed that collision-free trajectories that satisfy kinematic constraints exist at each planning step. The goal of this paper is to present an algorithm for planning overtaking trajectories on large temporal horizons in real-time. The main idea is as follows: once overtaking is desired by the behavior module an initial trajectory is simulated using a path tracking control algorithm for lane changing combined with a classical PI-controller for approaching the target speed. The controllers are parametrized in a way that the simulated trajectory will satisfy kinematic constraints. If no collisions are detected a corridor containing the simulated trajectory is created to state constraints for a subsequent optimal control problem to relax the trajectory and smooth it to be comfortable to the vehicle passengers.**

I. Introduction

A. Motivation

Trajectory planning is a crucial requirement for autonomous driving. Several methods able to plan safe and comfortable trajectories for lane keeping in real-time exist already [1], [2], [3], [4]. In scenarios where e.g. a tractor drives in front of the ego vehicle, more complex tasks, such as overtaking may be desired by the vehicle passengers. Obviously passing is a complicated procedure for an automated vehicle but for humans as well. This becomes clear when considering a person who has to look around several times to percept the environment and to decide whether overtaking is safe or not. Furthermore one has to additionally take into account social criteria like being polite and not to merge back into the original lane too early. In order to realize autonomous driving in the long-term and to enable whole missions of an automated vehicle it is necessary to be able to overtake in order to not being stopped by slow moving traffic participants for too much time. The goal of this paper is to present a real-time capable approach for trajectory planning in overtaking scenarios. We will show how to ensure the existence of valid trajectories in terms of collision-freeness and kinematic feasibility in each planning step once the maneuver is triggered. Therefore the temporal horizon is chosen sufficiently large in order to make sure the maneuver is completed at the end of the trajectory. The rough procedure is as follows: in order to predict further traffic participants, the well-known Kalman filter is utilized. Probabilistic prediction information is used to calculate forbidden zones that may be occupied by the predicted objects in the future. Subsequently a rough ”reference trajectory” is simulated using a pure-pursuit-steering controller for lane changing combined with a PI-velocity controller for approaching the target speed. Satisfaction of kinematic constraints is ensured since controllers are parameterized properly. We introduce a concept for a driving corridor that is placed around the reference trajectory and will be checked against the forbidden zones in order to identify collisions. Finally an optimal control problem is stated to relax the reference trajectory in terms of acceleration and jerk. In addition to the problem statement in [4] constraints are added from the driving corridor to make sure that relaxing the trajectory will not lead to collisions. In case that collisions with forbidden zones are detected the planner will not overtake but follow the leading vehicle instead. To make sure that once overtaking is triggered there will exist valid trajectories in each subsequent step, the prediction is parametrized conservative and thus the corridor will be enlarged in each step.

B. Related Work

The method presented in [5] splits the task of overtaking into 3 phases: diverting from the lane, driving straight in the adjacent lane and return to the original lane. The main idea is then to model an overtaking trajectory using quintic polynomials and to determine the exact coefficients by solving an optimal control problem to minimize the total kinetic energy exerted during overtaking by adhering to kinematic constraints at the same time. In [6] an approach is shown, that uses trigonometric functions to describe the overtaking path. The authors argue, that the path benefits from its trigonometric nature, e.g. differentiability and thus smoothness is guaranteed to be present everywhere on the path. To consider kinematic constraints, such as lateral acceleration, the curvature is reduced by stretching the sinus function and thus stretching the absolute distance covered during the overtaking process. The approach in [7] states an optimal control problem whose solution results in an overtaking trajectory. Constraints are composed from geometric information of the road and surrounding vehicles. Furthermore kinematic restrictions are incorporated as constraints as well. To not pass too close, the deviation towards a desired lateral distance to the leading vehicle is incorporated into the cost functional. Furthermore deviations from the target speed are penalized to make sure that passing will succeed sufficiently fast. To generate a smooth and fuel-saving behavior the models controls

[1] M. Graf, O. Speidel and K.Dietmayer are with the Institute of Measurement, Control and Microtechnology, Ulm University, 89081 Ulm, Germany, e-mail: firstname.lastname@uni-ulm.de.

are penalized as well. Surrounding vehicles are respected by adding further constraints in form of ”ramp barriers” to achieve convexity. Diverse maneuver variants and how they can be described combinatorial is investigated in [8]. Once a maneuver is identified, a trajectory is calculated by solving an optimal control problem similar to the one in [3]. To plan overtaking trajectories, constraints are added to the problem to enforce the desired behavior and to make the solution converge into a trajectory that passes the leading vehicle. The authors of [9] argue that optimal control problems for trajectory planning require constraints that are intrinsically logical. Thus, they state a mixed integer quadratic problem (MIQP) and wrap continuous constraints into logical ones. To perform overtaking, ramp barriers like already shown in [7] are reformulated as logical constraints and incorporated into the mixed integer quadratic program.
The goal of this paper is to present a novel approach that checks the existence of a feasible trajectory in terms of constrains by executing a prior simulation phase with subsequent collision checking. Only if it is indicated that at least one feasible solution exists, the simulated overtaking trajectory gets further processed.

II. Probabilistic forbidden zones

Prediction of traffic participants is indispensable for trajectory planning. To plan conservative and to leave enough leeway to react in time to unexpected behavior, the prediction has to provide more information than only an estimate of deterministic vehicle states over time. A popular way that is proven to deliver excellent results in practice is to utilize probabilistic approaches. E.g. in [10] a particle filter together with an appropriate driver model is applied to estimate the probability distribution of vehicle states. In [11] a nonlinear bicycle model combined with information from a digital map is used within an extended Kalman filter. The result consists of diverse motion hypothesis that are represented as discrete normal distributed states over time.
Since we aim to perform trajectory planning based on prior performed predictions, we use a linear Kalman filter whose equations are very fast to evaluate and to have enough time left for the subsequent planning. Because overtaking shall be performed on sections with low curvature only, where vehicles usually do not have to decrease their speed due to the road structure, we use a constant velocity model within the Kalman filter. Deviations between real and predicted movement are considered by uncertainty of the Gaussian probability distribution. The discretized model equation is straight forward as follows:

$$\underbrace{\begin{bmatrix} s_{t+\Delta t} \\ \dot{s}_{t+\Delta t} \end{bmatrix}}_{\boldsymbol{x}_{t+\Delta t}} = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}}_{\boldsymbol{A}_{\Delta t}} \underbrace{\begin{bmatrix} s_t \\ \dot{s}_t \end{bmatrix}}_{\boldsymbol{x}_t}. \tag{1}$$

Probabilistic information at time t is described by $\boldsymbol{\mu}_t$ and $\boldsymbol{P}_t$, mean and covariance of a Gaussian probability distribution function. While the mean $\boldsymbol{\mu}_t$ is equal to $\boldsymbol{x}_t$, the covariance is propagated by

$$\boldsymbol{P}_{t+\Delta t} = \boldsymbol{A}_{\Delta t}^{\mathrm{T}} \boldsymbol{P}_k \boldsymbol{A}_{\Delta t} + \boldsymbol{Q}. \tag{2}$$

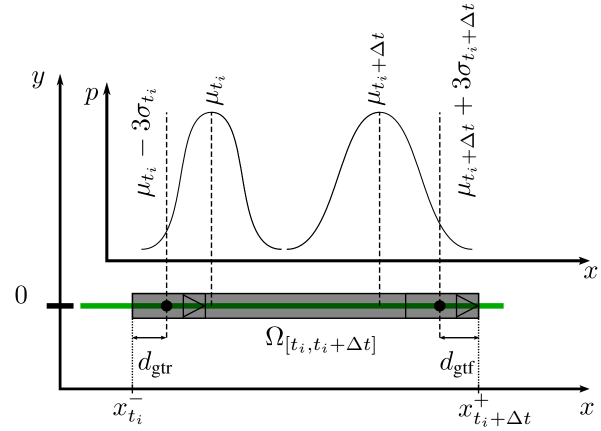

Fig. 1. Probabilistic forbidden zones within the time interval $[t_i, t_{i+1}]$ to consider longitudinal prediction uncertainty.

The process noise $\boldsymbol{Q}$ can be increased if the model does not match reality with sufficient accuracy. When applying equations 1 and 2, it is implicitly assumed that all vehicles keep staying on the lane they were associated with.
To attain probabilistic forbidden zones, the Kalman equations of prediction are applied for each relevant vehicle on the road over the temporal planning horizon T. Vehicle shapes are described as usual by rectangles. Therefore distances from a vehicles gravity center to its rear respective front are indicated by d_{gtf} respective d_{gtr}. Vehicle width is expressed by w. The forbidden zone for the i-th vehicle at time t is then represented by

$$\Omega_{t,i} = [x_{t_i}^-, x_{t_i}^+] \times [-w_i/2, +w_i/2]. \tag{3}$$

With

$$\begin{aligned} x_{t_i}^- &= \mu_{t,i} - 3\sigma_{t,i} - d_{\mathrm{gtr},i} \\ x_{t_i}^+ &= \mu_{t,i} + 3\sigma_{t,i} + d_{\mathrm{gtf},i}. \end{aligned} \tag{4}$$

To make sure that the trajectory is not only collision-free at each discrete time instant, but also between them, equation 3 is adapted for time intervals as follows:

$$\Omega_{[t_i, t_i+\Delta t]} = [x_{t_i}^-, x_{t_i+\Delta t}^+] \times [-w_i/2, +w_i/2]. \tag{5}$$

III. Trajectory simulation

After predicting relevant vehicles as explained in the previous section an initial trajectory is simulated and checked for collisions against the forbidden zones. During the simulation phase, all vehicles except for the leading vehicle, which we will refer to as the ”overtaking target”, are disregarded. For trajectory simulation the following vehicle model is used:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ a \\ \delta \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ v/L\tan\delta \\ a \\ u_1 \\ u_2 \end{bmatrix}. \tag{6}$$

With the vehicles gravity center indicated by $[x, y]$, yaw angle θ, velocity v, wheelbase L, steering angle δ and acceleration

a. Inputs u_1 and u_2 are the control variables which are to be set by the controllers. Road structure is represented by the center lines of the adjacent lanes relevant for overtaking. To make sure the maneuver is completed at the end of the planning horizon, the simulation must run through the 3 phases of overtaking: leaving, passing and merging. During simulation, the reference trajectory is generated by applying a pure-pursuit steering controller for path tracking combined with a PI-velocity controller for approaching the target speed as to see in [12]. To stay between acceptable acceleration bounds the following relation can be deduced:

$$\begin{aligned} a_{\text{lat}} &\overset{!}{\leq} a_{\text{lat,max}} \\ \Leftrightarrow \dot{\theta} v &\overset{!}{\leq} a_{\text{lat,max}} \\ \Leftrightarrow v^2/L \tan\delta &\overset{!}{\leq} a_{\text{lat,max}} \\ \Leftrightarrow \delta &\overset{!}{\leq} \arctan\left(a_{\text{lat,max}} L/v^2\right) = \delta_{\text{max}}. \end{aligned} \tag{7}$$

Thus, the steering angle δ is limited according to $|\delta| \leq \delta_{\text{max}}$. For each phase of overtaking it is merely left to decide which of both center lines need to be tracked. Figure 2 visualizes the procedure: during the first 2 phases, the left lane can be tracked, this leads to leaving and passing as desired. At each time instant the ego vehicles position is projected onto the right center line, resulting in the position $x_{\text{ego}}(t_i)$. As soon as $x_{\text{ego}}(t_i) > x_{\text{target}}(t_i) + x_{\text{safety}}$ is satisfied, the right lane will be tracked again and the vehicle returns back to the original lane. The safety distance can e.g. be determined by $x_{\text{safety}} = v_{\text{target}} T_{\text{gap}}$. Once the simulation has finished, the lateral deviation of the last position of the reference trajectory $y(T)$ and the right center line is compared. In case of a distance smaller than a predefined value, we assume that the ego vehicle has returned back to the right lane and overtaking is completed.

IV. Corridor Calculation

The main purpose of the corridor is to define the free-space to be incorporated into the optimal control problem as constraints. Obviously the actual free-space may become arbitrary complex and therefore extremely difficult to express from a mathematical point of view. Thus, it is desirable to chose a simple representation to favor the subsequent trajectory optimization. Additionally it must be ensured that at least one trajectory exists within the corridor that stays between dedicated acceleration bounds and thus satisfies the kinematic constraints.
This is one major feature of our approach: since the corridor will be constructed directly from the reference trajectory which is calculated as explained in section III, it is guaranteed that there exists at least one feasible solution within the corridor. A representation of free-space that can efficiently be integrated into the optimal control problem as constraints, is to define a circle for every discrete time instant of the prediction horizon. Each circle defines the space in which the temporal dedicated trajectory support point (which indicates the vehicles gravity center) must be located in. Figure 3 gives an illustrative impression of the corridor consisting of circles. To avoid collisions not only at every discrete time instant but also between them, the space that can potentially be covered at each interval $[t_i, t_{i+1}]$ must be described mathematically. Therefore figure 4 shows two adjacent trajectory support points inside their dedicated circles. The space covered by the vehicle shape according to the positions $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+1}$ is colored gray. For the sake of simplicity the yaw angle is assumed to remain constant within each interval.

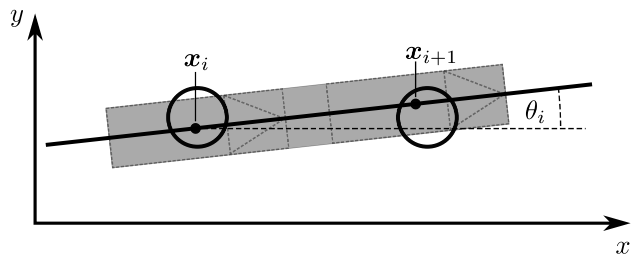

Fig. 4. Space reached by the vehicle shape within $[\boldsymbol{t}_i, \boldsymbol{t}_{i+1}]$. The yaw angle is assumed to remain constant within the interval. Support must be located inside the circles to not violate the corresponding constraints.

The corridor has to guarantee collision avoidance for a trajectory whose support points are all located inside the circles. The space that can be reached by the vehicle within the temporal interval $[t_i, t_{i+1}]$ is the union of reached spaces according to figure 4 but for any possible configuration of $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+1}$.
Thus, the next issue is to describe the potential reachable space for 2 adjacent support points that may vary arbitrarily inside their circles. The actual shape is complex and therefore hard to express. Not only for the sake of simplicity but also to stay real-time capable, it is advisable to do an over approximation. To keep the following deduction as simple as possible we assume without loss of generality that the circle mid-points have same y coordinates. In the first step, the inner tangent is calculated as illustrated in figure 5.

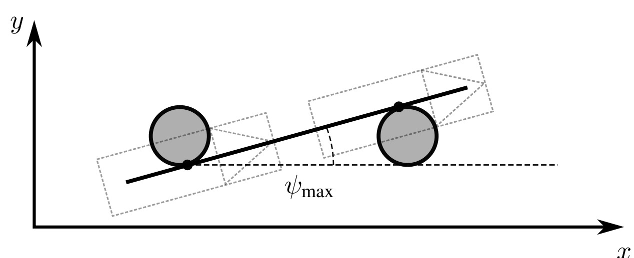

Fig. 5. Inner tangent between 2 adjacent circles. The connecting line indicates the maximum angle range $2\delta\psi_{\text{max}}$ the vehicle may vary within.

Subsequently extreme values of the y coordinate are approximated as too see in figure 6.

From these corner points we are able to determine a rectangle that covers the space between the circles. It remains to compute the space covered by the vehicle rear and front extensions. Therefore circles with the radii $d_{\text{gtf}} + w/2$ and $d_{\text{gtr}} + w/2$ are placed around the smaller circles. Figure 7

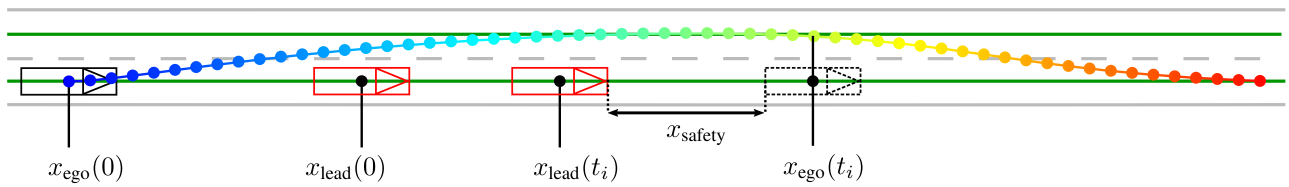

Fig. 2. Condition for changing the lane back: each trajectory point is projected onto the original lane. As soon as the projected vehicle shape is farther away from the overtaking target than s_{safety} the tracked path is changed to the center line of the right lane. The color coding of the simulated trajectory (from blue to red) indicates the time instances, blue indicates $t = 0$ and red $t = T$.

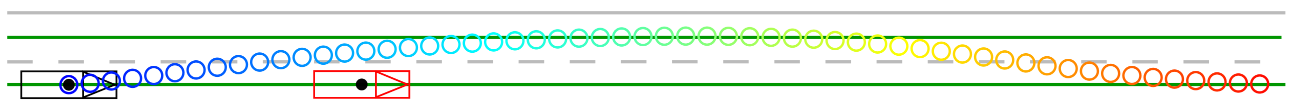

Fig. 3. Illustration of the driving corridor. Each circle is dedicated to a trajectory support point and indicates the area inside which the point may vary to stay collision-free.

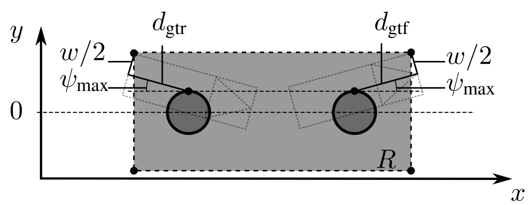

Fig. 6. Determination of highest y values reached by the vehicle shape.

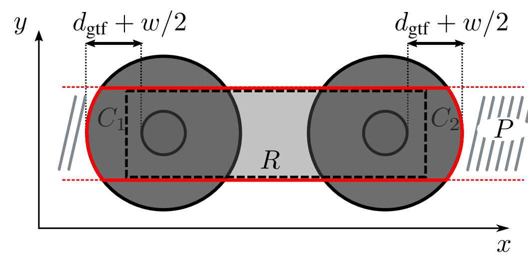

Fig. 7. Over approximation of the potential covered vehicle shape.

illustrates the procedure. Using the whole gray area would lead to very restrictive behavior where collisions are detected unnecessarily and overtaking trajectories will never be found. Since it is known that the maximum and minimum values of y the vehicle can cover, are the maximum and minimum values of y of the rectangle, the red bounded area in figure 7 can be used for collision checking instead. This area can be expressed by applying logical operations as follows:

$$S = R \cup (C_1 \cap P) \cup (C_2 \cap P). \tag{8}$$

Where P indicates the area between the upper and lower dashed line in figure 7. Thus, the approximated shape in figure 7 must be checked for collisions with forbidden zones corresponding to the same temporal interval. Since the approximated shape composes of a rectangle and 2 circles and the forbidden zones are described by rectangles as well, all shapes to deal with are convex. This can be exploited by using convex collision checking algorithms such as the separate axis theorem to significantly reduce the run time. Finally, the circle radii have to be determined. To keep things simple, the same radius r is assigned to each circle of the corridor. Subsequently collision checking is done as described above. To figure out what radius is suitable, diverse radii from $r_{\text{max}}, \ldots, r_{\text{min}}$ are applied, starting with the largest value. The first radius that does not result in a collision is considered as suitable.

V. Optimization Problem Statement

To obtain the final trajectory an optimal control problem is stated to smooth the reference trajectory within the corridor. Therefore the problem statement in [4] is extended for additional constraints. The general statement is as follows:

$$\min_{\boldsymbol{x}_i, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N-1}} J[\boldsymbol{x}_i, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N-1}] \tag{9}$$

$$\text{s.t. } h(\boldsymbol{x}_i) \leq 0, \;\; i = 3, \ldots, N-1.$$

With the cost function

$$J[\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N-1}] = \sum_{i=2}^{N-2} L(\boldsymbol{x}_i, \boldsymbol{x}_{\text{dd},i}, \boldsymbol{x}_{\text{ddd},i}) \Delta t$$

$$L(\boldsymbol{x}_i, \boldsymbol{x}_{\text{dd},i}, \boldsymbol{x}_{\text{ddd},i}) = w_{\text{spatial}} j_{\text{spatial},i} + w_{\text{acc}} j_{\text{acc},i} + w_{\text{jerk}} j_{\text{jerk},i} \tag{10}$$

$$j_{\text{spatial}} = ||\boldsymbol{x}_i - \boldsymbol{x}_{\text{Ref}}||_2^2 \tag{11}$$

$$j_{\text{acc}} = ||\boldsymbol{x}_{\text{dd},i}||_2^2 \tag{12}$$

$$j_{\text{jerk}} = ||\boldsymbol{x}_{\text{ddd},i}||_2^2 \tag{13}$$

Acceleration $\boldsymbol{x}_{\mathrm{dd},i}$ and jerk $\boldsymbol{x}_{\mathrm{ddd},i}$ are calculated using finite differences:

$$\boldsymbol{x}_{\mathrm{dd},i} = \frac{\boldsymbol{x}_{i+1} - 2\boldsymbol{x}_i + \boldsymbol{x}_{i-1}}{\Delta t^2}, \quad (14)$$

$$\boldsymbol{x}_{\mathrm{ddd},i} = \frac{-\boldsymbol{x}_{i-2} + 3\boldsymbol{x}_{i-1} - 3\boldsymbol{x}_i + \boldsymbol{x}_{i+1}}{\Delta t^3}. \quad (15)$$

There are two kinds of constraints: spatial constraints to make the trajectory remain within the corridor and kinematic constraints to stay within acceleration bounds. Spatial constraints are as follows:

$$h(\boldsymbol{x}_i) = ||\boldsymbol{x}_i - \boldsymbol{x}_{\mathrm{Ref},i}||_2^2 - R_i^2 \leq 0. \quad (16)$$

Kinematic constraints are formulated as:

$$h(\boldsymbol{x}_i) = ||\boldsymbol{x}_{\mathrm{ddd},i}||_2^2 - a_{\mathrm{max}}^2 \leq 0. \quad (17)$$

The spatial term penalizes the deviation from the reference trajectory. Without smoothing terms the unconstrained global optimal solution would obviously be identical with the reference trajectory. To relax the trajectory in terms of acceleration and jerk, the global optimum is moved by adding appropriate smoothing terms 12 and 13. To make sure the solution will remain within the corridor while not violating kinematic restrictions, constraints 16 and 17 are added.

VI. Algorithmic Overview

The following overview gives a further understanding of what is done and in which order: At first all traffic participants are predicted and the forbidden zones are determined. Afterwards an initial trajectory is simulated using a path tracking and a velocity controller. To maneuver around further traffic participants during the simulation the tracked center lines of the adjacent lanes are switched in the right moments. Next, the trajectory is checked for intersections against the forbidden zones. In case no collisions occur, the method breaks up and overtaking is assessed as unsafe. Otherwise, the simulated trajectory is smoothed using quadratic programming in the final step.

Overview

```
Predict each vehicle except the ego vehicle
Create probabilistic forbidden zones from predictions
Simulate an initial trajectory
Place the corridor around the trajectory, select circle
radii by:
for r_max, ..., r_min do
    Check corridor for collisions
    if collision detected then
        try next radius
    else
        use current radius
    end if
end for
if no radius found then
    return overtaking unsafe
end if
Solve the optimal control problem
return Overtaking trajectory
```

TABLE I
Parameters for simulation

N	v_0	T	k_p	k_I	a_{max}
61	$13.89\,\mathrm{m\,s^{-1}}$	$20\,\mathrm{s}$	5	0.1	$5\,\mathrm{m\,s^{-2}}$

VII. Evaluation

To validate the approach the traffic evolution during an overtaking maneuver with an oncoming vehicle on the opposite lane is investigated. The method is implemented within C++, the non-linear optimization problem is solved using the SQP-method of WORHP [13]. Figure 8 shows snapshots of the scene at diverse time instants, the ego vehicle is drawn in black and other vehicles in red. Corresponding velocities and accelerations are shown in figures 9 and 10 respectively. Before and after overtaking when the vehicle needs to move in longitudinal traffic only, the decision layer switches to the planner in [4].
Parameters for simulation are chosen as in table I.

The number of trajectory support points is indicated by N. The target speed is v_0, the temporal planning horizon T has to be chosen large to make sure the maneuver can be completed within the simulation phase of the algorithm. Controller gains for path tracking and approaching the target speed are k_p and k_I for proportional gain and integral gain.

As can be seen in figure 8 the ego vehicle leaves the original lane and aims to track the left center line in the very beginning of the scene. To approach the target speed and to make sure that passing will succeed sufficiently fast, the ego vehicle accelerates and thus reaches the target speed v_0 after $t \gtrsim 4\,\mathrm{s}$. Merging back into the original lane is not performed until the ego vehicle has a small buffer in front of the overtaking target to not merge back too close. Speed and acceleration values are smooth and thus trackable by the vehicle controller, furthermore comfortability to the vehicle passengers is increased contrary to non smooth values.

Figures 8, 9 and 10 show how the vehicle behaves if the overtaking trajectory is tracked perfectly by the vehicle controller. Of course it is of interest to see the planned trajectory over the complete temporal horizon a well. Therefore figures 11, 12 and 13 show a snapshot of the planned trajectory at $t = 0\,\mathrm{s}$.

The planned path starts towards the opposite lane as soon as the maneuver is triggered. The acceleration takes large values $\gtrsim 3\,\mathrm{m\,s^{-2}}$ at the very beginning of the horizon. Thus, the velocity increases rapidly and the target speed is planned to be reached after $\approx 2\,\mathrm{s}$. The planned merge back takes place after $\approx 6\,\mathrm{s}$ with a sufficient safety distance to the overtaking target and the oncoming vehicle as well. Regarding figure 10 one can see a small "dogleg" in the acceleration at $\approx 7\,\mathrm{s}$, this is because the planner switches back to longitudinal planning as soon as the maneuver is finished. Nevertheless, velocity

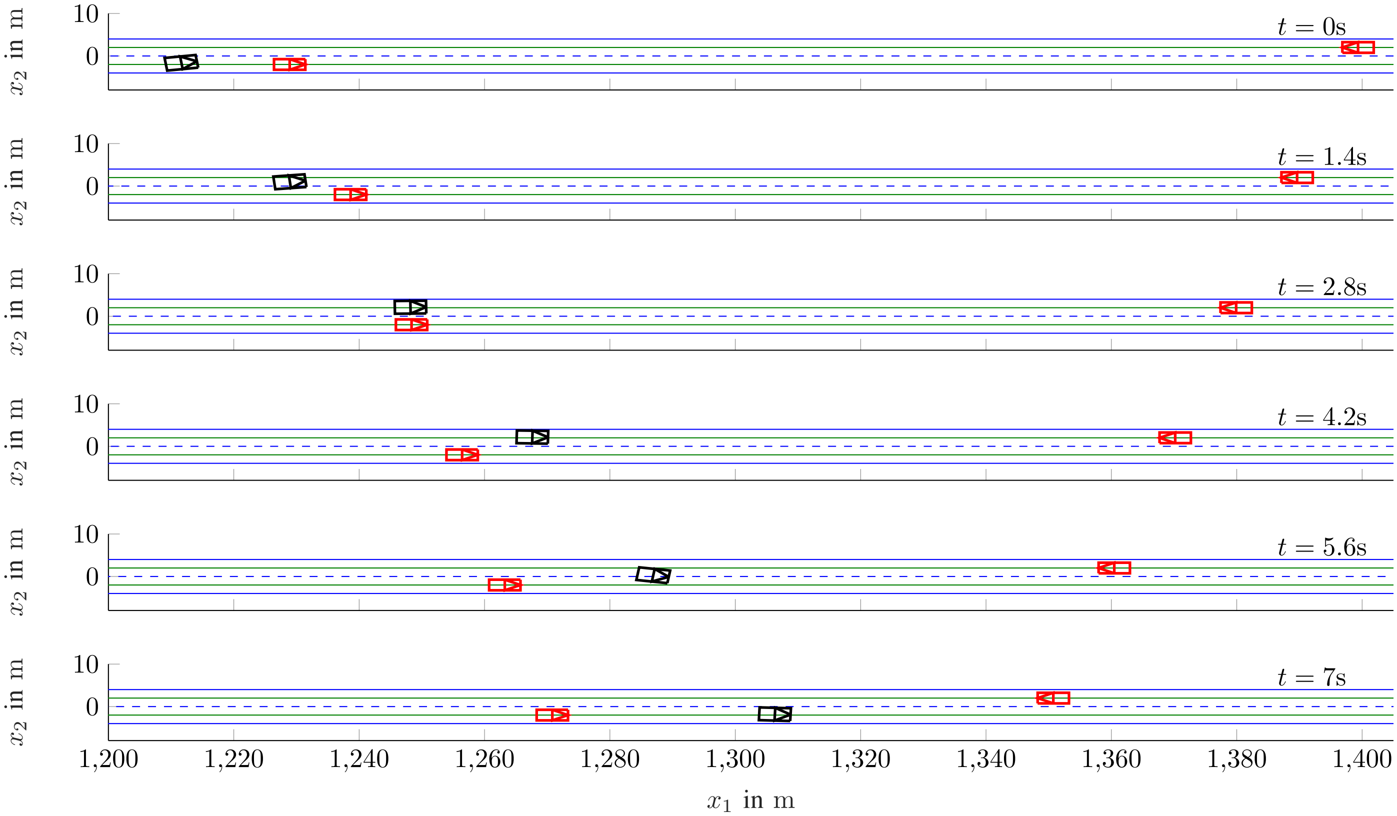

Fig. 8. Evolution of traffic during an overtaking maneuver. The ego vehicle is depicted in black, the leading as well as the oncoming vehicle in red.

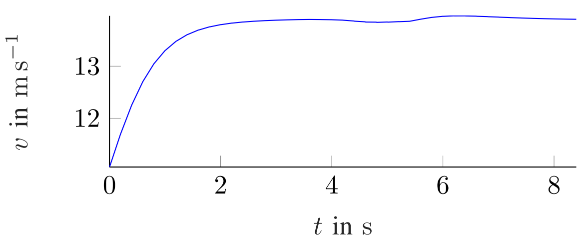

Fig. 9. Velocity of the ego vehicle.

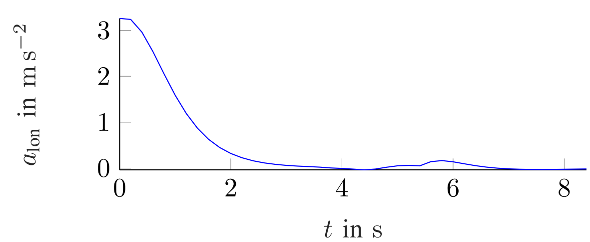

Fig. 10. Acceleration of the ego vehicle.

and acceleration are smoothed properly by the optimizer and no constraints are violated at any time. Thus, the trajectory is trackable by the vehicle controller and promising to be comfortable to the vehicle passengers.
Trajectory planning cycles were repeated for 31 times and during the overtaking maneuver, the average computation time measured is $t_{\text{meas}} = 43.68\,\text{ms}$.

VIII. Conclusion and Future Work

The goal of this paper was to present a practical method for planning overtaking trajectories in real-time. The main concept is to firstly predict all traffic participants located on the road using probabilistic Kalman-filtering method to obtain forbidden zones over the prediction horizon. Subsequently an aggressive designed path following controller to compute a trajectory for overtaking the leading vehicle is simulated. In the next step a corridor consisting of circles around each support point of the initial trajectory is created to describe the free-space. Finally an optimization problem is stated to relax the initial calculated trajectory within the corridor. There are 2 main features of our approach: Firstly, it is ensured that the end of the overtaking trajectory is located on the original lane again and therefore the maneuver is completed at the end of the planning horizon. Therefore the decision whether overtaking should be performed or not can be made on a trajectory level. The second feature is, that the corridor is guaranteed to have at least one solution that is feasible in the sense that is satisfies kinematic constraints.
To make the method more robust and thus applicable in multiple scenarios the simulation of the reference trajectory can be repeated with various sets of parameters to increases the quality of the reference trajectory. To enlarge the corridor circles radii can be determined locally instead of choosing one global radius for every circle. This will reduce the costs

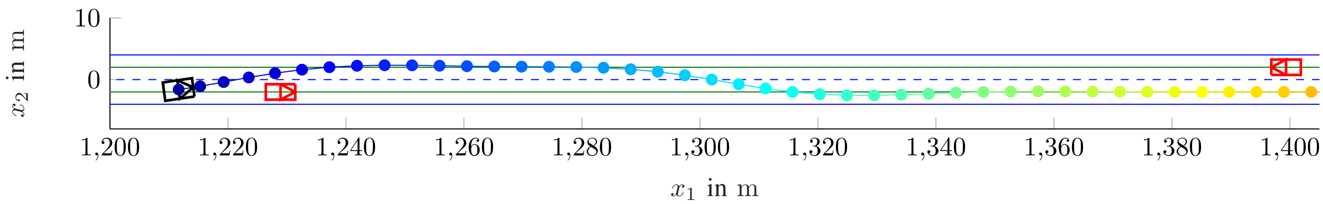

Fig. 11. Snapshot of an overtaking scenario.

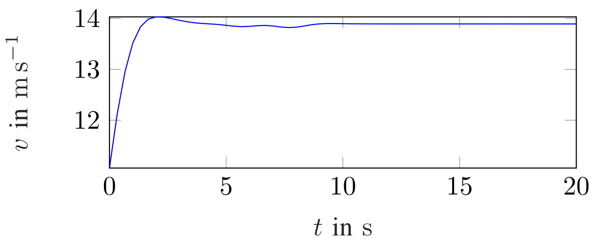

Fig. 12. Trajectory velocity.

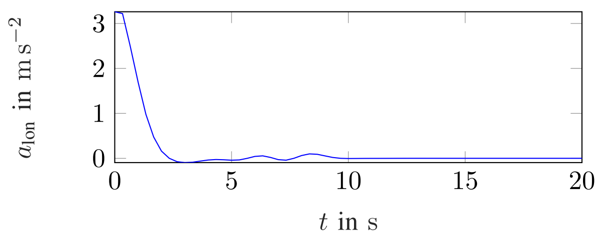

Fig. 13. Trajectory acceleration.

according to the optimal control problem and thus the final trajectory will be smoother and therefore more comfortable for the vehicle passengers.

Acknowledgment

This paper has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 69075.

References

[1] M. Werling *et al.*, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 987–993.

[2] 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.

[3] J. Ziegler *et al.*, "Trajectory planning for bertha-a local, continuous method," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 450–457.

[4] M. Graf *et al.*, "Trajectory planning for automated vehicles using driver models," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 1455–1460.

[5] T. Shamir, "How should an autonomous vehicle overtake a slower moving vehicle: design and analysis of an optimal trajectory," *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 607–610, April 2004.

[6] S. Zhang *et al.*, "Trajectory planning of overtaking for intelligent vehicle based on x-sin function," in *2014 IEEE International Conference on Mechatronics and Automation*, Aug 2014, pp. 618–622.

[7] N. Murgovski and J. Sjberg, "Predictive cruise control with autonomous overtaking," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 644–649.

[8] P. Bender *et al.*, "The combinatorial aspect of motion planning: Maneuver variants in structured environments," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, June 2015, pp. 1386–1392.

[9] X. Qian *et al.*, "Optimal trajectory planning for autonomous driving integrating logical constraints: An miqp perspective," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 205–210.

[10] S. Hoermann, D. Stumper, and K. Dietmayer, "Probabilistic long-term prediction for autonomous vehicles," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 237–243.

[11] D. Petrich *et al.*, "Map-based long term motion prediction for vehicles in traffic environments," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, Oct 2013, pp. 2166–2172.

[12] Y. Kuwata *et al.*, "Motion planning in complex environments using closed-loop prediction," 2008.

[13] C. Büskens and D. Wassel, "The esa nlp solver worhp," in *Modeling and optimization in space engineering*. Springer, 2012, pp. 85–110.