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

A Generic Driving Strategy for Urban Environments

Constantin Hubmann[1], Michael Aeberhard[2] and Christoph Stiller[3]

Abstract— **Autonomous driving in urban environments depends on the ability to interpret the current situation and to react accordingly. This means to continuously make decisions for certain comfort-optimized maneuvers under the constraints of traffic rules and feasibility. This work presents a novel, longitudinal driving strategy formulated as a discrete planning problem. Instead of designing an algorithm for a single one of various potential subproblems, two interfaces are presented, called static and dynamic events, that are capable of representing any situation along the chosen lane of the autonomous vehicle. This allows fast, analytic calculation of Inevitable Collision States which are used as heuristic to realize a guided A* search. Instead of being limited to a small, finite set of maneuvers as rule-based driving strategies like state machines are, the algorithm selects the optimal of an infinite number of possible, implicit maneuvers. The presented algorithm has a worst-case runtime of 80 ms for a planning horizon of 13 seconds and is therefore capable of running online. The approach is evaluated on a simulator in a complex city scenario and on a prototype vehicle on the test track.**

I. INTRODUCTION

Autonomous driving will become a reality in the near future. Driver assistance systems in production vehicles have steadily increased their level of automation in the past few years. Level 2 partial automation systems, as defined by SAE J 3016 [1], such as a traffic jam assist or lane keeping assist, are already in series production. These systems, however, still have their limitations, where most systems can only be activated under certain conditions, such as on the highway, and additionally expect the driver to take over control at any given moment when a limit is reached. Level 3 automation and above marks the next step in vehicle automation, removing the requirement that the driver continuously monitors the system, as well as increasing the conditions under which an automated driving system can be activated. These Systems are currently still in the research phase.

In the past years, automated driving research has focused on highway or highway-like scenarios and many projects have successfully developed vehicles with the intent of Level 3 and above automation for the highway [2]–[4]. Current research starts to focus on automated driving in more complex and urban environments, where some projects have already demonstrated early prototypes in real traffic [5], [6]. Autonomous driving in reduced urban environments was pioneered by the DARPA Urban Challenge [7], where autonomous vehicles from various universities around the world coped with the challenge of various simple urban driving scenarios on a closed test track [8]. Despite this progress, autonomous driving in urban environments remains a challenge in robotics research.

The European Union project AdaptIVe [9] aims to advance the development of automated driving in various scenarios, including city driving. Compared to highway driving, urban scenarios pose many new challenges and situations which must be properly handled, including traffic lights, stop signs, pedestrian crossings, etc. These challenges increase the complexity of the longitudinal decision making of an autonomous vehicle within its own lane.

As the authors are convinced that complex scenarios in urban environments are not tractable by rule-based systems anymore, this work demonstrates, to our knowledge, the first realization of a driving strategy as a global, optimized planner. The key contribution of this paper is a generic longitudinal planning approach, which can handle many different types of situations, from cruise control to decision making at traffic lights, in one single algorithm in a combined and optimized manner. It is the first driving strategy which can decide between an infinite amount of possible, implicit maneuvers. By using Inevitable Collision States (ICS) as a heuristic for a guided A* search, the demands for real time application are easily met. The driving strategy provides dynamically feasible, safe, legal and comfortable long-term maneuvers. The developed algorithm is integrated into a prototype vehicle designed for basic automated driving in urban environments.

II. RELATED WORKS

Driving in urban environments requires the capability of an autonomous car to adapt the velocity to master the current situation. This is the case, as scenarios such as passing/approaching a traffic light, deciding to pass before or behind a merging vehicle or even braking for crossing pedestrians have to be solved while driving on a chosen lane. To steer the autonomous car along the chosen lane, popular algorithms have been proposed to generate a smooth trajectory [10]–[12]. While these algorithms solve the local trajectory planning problem efficiently, they lack the capability of generating long term trajectories ($t_{Hor} > 5s$), obeying traffic rules and for maneuver decisions (e.g. to pass or not to pass a traffic light). This is the case as decision making and complex constraints require a combinatoric problem formulation, which becomes computationally intractable

[1]Constantin Hubmann is with BMW Research and Technology, Munich, Germany and with Karlsruhe Institute of Technology (KIT), Institute for Control Theory, 76131 Karlsruhe, Germany. constantin.hubmann@bmw.de

[2]Michael Aeberhard is with BMW Research and Technology, Munich, Germany. michael.aeberhard@bmw.de

[3] Christoph Stiller is with Karlsruhe Institute of Technology (KIT), Institute for Control Theory, 76131 Karlsruhe, Germany. stiller@kit.edu

978-1-5090-1889-5/16/$31.00 ©2016 IEEE

under the high demand of combined lateral and longitudinal comfort optimization of the trajectory planner [13]. Directly calculating one trajectory for every combinatoric possibility (maneuver) as done in [14] is computationally infeasible and the needed generic constraint generation is difficult, as the authors notice.

Therefore, a high-level decision making process is needed to pick the right long-term maneuver in a complex situation to enable fast planning of the corresponding underlying trajectories. Typically, a selected maneuver is a discrete decision corresponding to a certain local minimum in a scenario with various local minima, i.e. various available options (e.g. to pass or not to pass a switching traffic light). The maneuver itself has often a longer duration than the trajectory planner's horizon. Therefore, its execution has to be steered and surveyed in a higher layer. This happens in the car's behavioral layer [15], which is also referred to as *driving strategy* in this work. It's task is to decide for a legal, safe and comfortable as well as dynamically feasible maneuver and to provide the maneuver's intermediary goal states to the trajectory planner for execution (see Fig. 1).

In literature, a variety of algorithms, solving certain specific situations under different priorities, have been proposed. For example, [16] solves the traffic light approach problem under an energy optimal point of view, while [17] selects the maneuver based on the minimum initial jerk. In [18] an anticipatory, energy efficient approach onto slower vehicles is demonstrated, while [19] demonstrates how human vehicle-following behavior can be learned. While these algorithms solve single subproblems like the approach of a traffic light or following behavior, these different submodules have to be combined at some point. The classic approach is to realize the selection process as a decision tree as in [20] or as a rule-based system as the winning Team of DARPA's Urban Challenge [8] did. Unfortunately, this requires tedious tuning of state transition parameters. While it is possible to model the number of maneuvers as a finite set for driving on highways, this becomes prohibitively complex in urban environments. Here, complex, tactical maneuvers like *"early braking during following behavior because of a traffic light switching to red in a larger distance"* have to be realized, which may lead to a virtually infinite number of explicit maneuvers. By modeling one state for each maneuver, this leads to an exponentially growing number of state transition rules. In addition, events which have not been considered at implementation time are not regarded in the state machine and may not be managed at all.

Another approach to combine different algorithms, is to run them in parallel and choose the minimum in terms of the generated acceleration [21], [22]. While a simple minimum operator has the advantage, that state transitions need not to be modeled explicitly, it also creates overhead by running several algorithms in parallel and does not guarantee to traverse smoothly from one state to the other. Finally, all these combination approaches (state machines, decision trees and minimum acceleration) have in common, that the decision does not consider several events simultaneously,

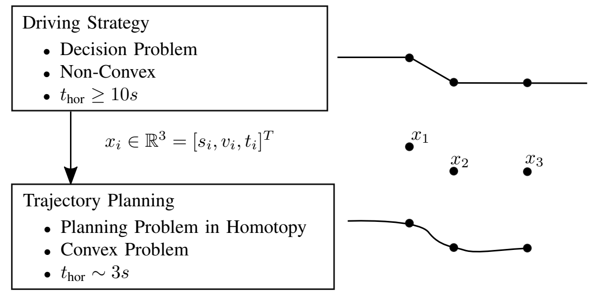


Fig. 1: The Driving Strategy solves the global decision problem and parameterizes the trajectory planner for finding the optimal trajectory in the global minimum.

but reaction is limited to a single event at any given time.

A different approach is to represent the situation with high-level, semantic states and search the generated graph, which describes possible transitions [23]. Nonetheless, the generated maneuver sequence may be dynamically infeasible and must be validated by planning it in the path-velocity-time state space [24], which makes the high-level planning obsolete. This approach requires a high amount of parameter tuning and preprocessing to translate the current situation into such a high level and abstract state space. Besides, every maneuver is described explicitly again, which requires a design process which thinks about every possible situation.

An algorithm which solves the maneuver decision problem in the path-velocity-time space is demonstrated in [25]. A velocity planner generates longitudinal and dynamically feasible trajectories for the following of a selected path which is temporarily occupied by dynamic objects. While the approach realizes fast planning, it is limited to plan only time-optimal trajectories for a simplistic motion model along the edges of pseudo-obstacles representing dynamic objects in the path-time space. Optimization in terms of changing speed limits (legal and curvature based), comfort (i.e. minimizing total acceleration) and deceleration to static events (e.g. a red traffic light) is not possible.

None of the presented algorithms provide a driving strategy for urban environments that generates maneuvers under the consideration of comfort, traffic rules and feasibility over a longer horizon and several different events.

III. Problem Statement

Driving along a predefined lane (e.g. provided by the mission plan of the knowledge representation layer defined in [15]) represents the standard problem of urban autonomous driving. This is the case as most situations may be formulated as driving along a predefined lane with arbitrary geometry. This also includes situations like turning at intersections with oncoming, intersecting traffic as well as passing pedestrians. In our formulation, the motion is calculated in the longitudinal direction only, i.e. we focus on the motion in a one dimensional direction. This approach is known as path-velocity decomposition [26].

Let $\mathbf{p_i} = [p_x, p_y]^T \in \mathbb{R}^2$ be a point in global coordinates on a lane's reference (center) line c, then $s(p_i) \in \mathbb{R}$ is the traveled distance along the curve in the interval $[p_0, p_i]$. The absolute velocity is bounded by $\dot{s} \in [0, v_{\text{max}}]$ with $v_{\text{max}}(s)$ being a function of the path's curvature κ at distance s, i.e. $v_{\text{max}}(s) = f(\kappa(s))$ and the vehicle's acceleration u being the system's bounded input $u \in [a_{\text{min}}, a_{\text{max}}]$. The movement of the vehicle may then be described by the following set of linear, differential equations:

$$\begin{bmatrix}\dot{s}\\ \ddot{s}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}s\\ \dot{s}\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}u. \tag{1}$$

Along the given lane c exists a finite set E of events E_i that occupies the lane for a time interval $\tau^{E_i} = [t^{E_i}_{\text{start}}, t^{E_i}_{\text{end}}]$ at a certain position $s^{E_i}(t)$ with $t \in \tau^{E_i}$. These events must not intersect with the ego vehicle's position s_{ego} at any time.

In addition, a finite set L of traffic laws $L_i(s)$ is imposed along the road and limits for example the absolute velocity. The goal of the driving strategy is to select and parameterize a valid maneuver in longitudinal direction. This may be formulated as a minimization problem over J:

$$\min_{u(t)}(J) = \min_{u(t)} J(s_{\text{ego}}, \dot{s}_{\text{ego}}, \ddot{s}_{\text{ego}}, \kappa(s), E, L). \tag{2}$$

As this problem does, in general, not only have one global minimum but different local ones (i.e. different maneuvers) as well as various constraints (e.g. speed limits), it is a constrained non-convex problem.

The driving strategy's task is hereby to find a global optimum, i.e. a long term solution ($t_{\text{hor}} \geq 10s$) for the non-convex optimization problem. This solution's subparts (i.e. the goal state in three seconds) are provided and executed by the trajectory planner presented in [10] under other optimization constraints (e.g. minimum combined lateral and longitudinal jerk). This local planner is able to solve the resulting convex optimization problem in an optimal way on a short horizon ($t_{\text{hor}} \sim 3s$). This combination of a global and a local planner is illustrated in Fig. 1.

IV. Approach

This work aims to plan driving maneuvers that consider traffic laws, long term comfort and human driving conventions. To overcome the limitations and the complexity of rule-based decision making, we formulate an explicit objective functional whose minimum defines the optimal driving maneuver in a given situation.

The optimization problem is formulated as a discrete planning problem [27] in the state space $\mathcal{X} \subseteq \mathbb{R}^3$ with states $x = [s, v, t]^T \in \mathcal{X}$ and solved by a A* graph search [28]. The state x_i is hereby defined as the state at planning step i. The graph is constructed online by sampling through the set of possible actions $\mathcal{A}$ for a sample time of Δt when expanding a node. As no states with a negative velocity, $v_i < 0$ are considered, the result is a directed, acyclic graph.

A. Transition Model

The discretized state transition model may be written as

$$x_{i+1} = \begin{bmatrix}s_{i+1}\\ v_{i+1}\\ t_{i+1}\end{bmatrix} = \begin{bmatrix}1 & \Delta t & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & \Delta t\end{bmatrix}\begin{bmatrix}s_i\\ v_i\\ t_i\\ 1\end{bmatrix} + \begin{bmatrix}\frac{1}{2}(\Delta t)^2\\ \Delta t\\ 0\end{bmatrix} a_i \tag{3}$$

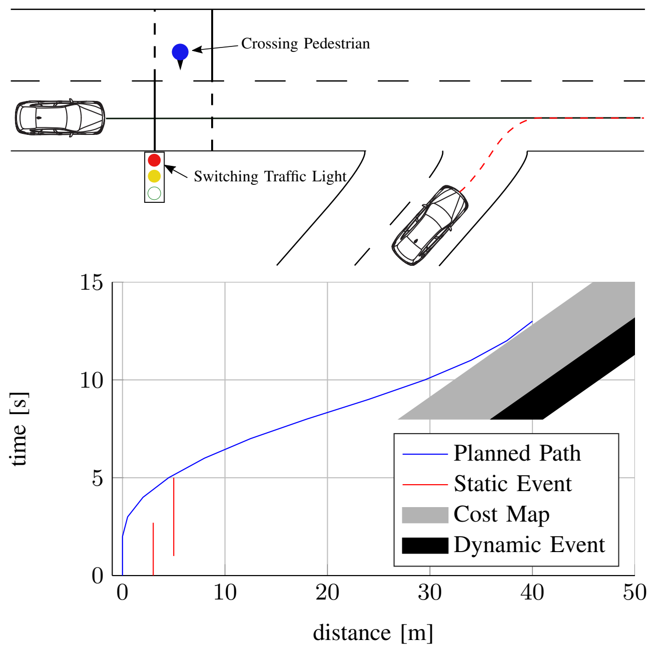


Fig. 2: Top figure: Typical situation in an urban environment: A traffic light is already switching while a pedestrian is still passing the road. Additionally, another car is merging on the autonomous car's lane. Bottom figure: Corresponding representation in the state space including the planned trajectory.

with a_i being the action, i.e. the acceleration, selected in state x_i and executed for Δt.

B. Cost functional

The step cost $c(x_i, a, x_{i+1}, E)$ is the cost for taking action a in state x_i to traverse to state x_{i+1}. The overall cost of this path is the sum of all costs along the path, $\sum_{x_i = x_{start}}^{x_{goal}} c(x_i, a, x_{i+1}, E)$ [28, p. 68].

The goal is to find the path from the start state to a goal state with minimal costs. To represent the different parameters of the optimization problem, a sum of different costs is used as step cost

$$c(x_i, a, x_{i+1}, E) = c_{\text{V}}(x_{i+1}) + c_A(a) + c_E(x_i, x_{i+1}, E) \tag{4}$$

with $c_{\text{V}}(x_{i+1})$ being the cost for any deviation to the desired speed, $c_A(a)$ being the cost for taking action a and $c_E(x_i, x_{i+1}, E)$ being the cost for a collision while traversing from x_i to x_{i+1}.

C. Desired Speed

The desired speed $v_{\text{des}}(s)$ at position s is the desired speed under the assumption that no events exist. It is a combination of the current legal speed limit $v_{\text{law}}(s)$ and $v_{\text{curve}}(s)$, i.e. the limit introduced by the road's curvature. The maximum curve speed v_{curve} is defined as in [22] via a maximum allowed lateral acceleration, $a_{lat,curve}$, in the curve which is defined by its radius $r_{curve}(s)$ along the path:

$$v_{\text{curve}}(s) = \sqrt{a_{\text{lat,curve}} r_{\text{curve}}(s)}. \tag{5}$$

The desired speed v_{des} is defined by the minimum of v_{law} and v_{curve}, followed by a processing step, which adds an approach phase to every upcoming curve. This is shown in Fig. 3.

The velocity-dependent costs, c_V, are defined by the deviation to the desired velocity v_{des}. While too high velocities are punished quadratically, too low velocities are punished linearly to allow lower velocities during decelerating upon events (as red traffic lights). $c_V(x_{i+1})$ is then defined as follows:

$$c_V(x_{i+1}) = \begin{cases} (v_{i+1} - v_{des}(s_{i+1}))^2, & v_{i+1} > v_{des}(s_{i+1}) \\ 0, & v_{i+1} = v_{des}(s_{i+1}) \\ \frac{1}{2}(v_{des}(s_{i+1}) - v_{i+1}), & v_{i+1} < v_{des}(s_{i+1}) \end{cases} \quad (6)$$

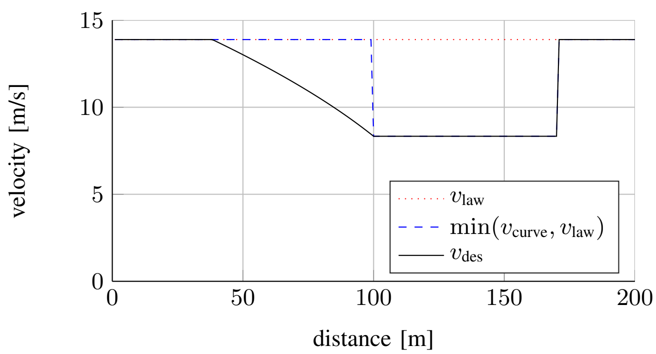


Fig. 3: Along the road s, the velocity is limited by speed limits and the curvature. The curve approach velocity is also part of the desired velocity.

D. Representation of the Environment

Along the path, various events like cars, traffic lights and passing pedestrians may occur. The idea of this work is to create two simple but powerful representations for events, which allow the representation of any possible happening in the longitudinal direction. With this approach, the algorithm may be easily extended by adding further events without changing runtime or resolvability. The two classes are:

- static events E_S
- dynamic events E_D

Static events block the road at a certain position s during a time interval $\tau \in [t_{start}, t_{end}]$. The same applies for dynamic events, with the difference that their position $s(t)$ is not constant but changes with time. In addition, a specific dynamic event $E_{D,i}$ has a defined length of the corresponding object, $l^{E_{D,i}}$, and occupies therefore an area in the distance-time plane of the state space $\mathcal{X}$ (see Fig. 2). It also allows for the definition of a following distance $d^{E_{D,i}}$, which defines a temporal-spatial cost map $M^{E_{D,i}}$, to realize a smooth following behavior of the autonomous car. The cost map $M^{E_{D,i}}$ is realized as an increasing linear function, defined by $s^{E_{D,i}}(t), d^{E_{D,i}}, l^{E_{D,i}}$ in the interval $[t_{Start}^{E_{D,i}}, t_{End}^{E_{D,i}}]$.

The overall costs are defined as

$$c_E(x_i, x_{i+1}, E) = c_{E_S}(x_i, x_{i+1}) + c_{E_D}(x_{i+1}), \text{ with} \quad (7)$$

$$c_{E_S} = \begin{cases} \text{inf}, & \text{if } \exists E_{S,i} \in E_S : \{\overrightarrow{x_i x_{i+1}}\} \cap E_{S,i} \neq \emptyset \\ 0, & \text{otherwise} \end{cases}, \text{ and} \quad (8)$$

$$c_{E_D} = \begin{cases} \text{inf}, & \text{if } \exists E_{D,i} \in E_D : x_{i+1} \in E_{D,i} \\ M^{E_{D,i}}(x_{i+1}), & \text{if } \exists E_{D,i} \in E_D : x_{i+1} \in M^{E_{D,i}} \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

$\{\overrightarrow{x_i x_{i+1}}\}$ is hereby defined as the set of points on the vector between x_i and x_{i+1}.
The classification into a small amount of classes allows a consistent representation of events in the state space which guarantees a short planning time due to heuristics.

1) Modeling of Traffic Lights: As the handling of traffic lights motivated different other publications [17], [22], the realization as a static event is shown exemplary. A static event $E_{S,i}$ is defined by the triple $E_{S,i} = [s, t_{Start}^{E_{S,i}}, t_{End}^{E_{S,i}}]^T$. The interval $[t_{Start}^{E_{S,i}}, t_{End}^{E_{S,i}}]$ defines the forbidden (red phase) time interval. If no further CAR2X information is available, the current green and red phase is assumed to last forever. During a yellow phase, the legal length of the yellow phase is used to predict the traffic signal. That way, the algorithm implicitly handles the decision to pass or not to pass a recently switched traffic light. While the algorithm's event is therefore independent to potentially available CAR2X information that information can be easily added if available by a different event initialization.

2) Leading/Merging Vehicles: A Dynamic Event $E_{D,i}$, which is in front of the ego vehicle is parameterized by the vector $[s(t), d(t), l, t_{Start}^{E_{D,i}}, t_{End}^{E_{D,i}}]^T$. As the focus of this work is on the planning algorithm but not the predicted longitudinal behavior of other vehicles, the velocity of dynamic events is assumed to be constant. Nonetheless, a better prediction function can be easily included by replacing the linear function $s^{E_{D,i}}(t)$. In addition, other vehicles are predicted to leave/enter the lane by a simple rule-based classifier to enable foresight decision making as well as cooperative behavior.

E. Actions

The set of actions $\mathcal{A}$ represents different accelerations during a discrete planning step Δt. Punishing accelerations quadratically reduces the duration and intensity of acceleration which increases the driver's comfort.

F. Graph Search and Heuristics

The use of the A* algorithm requires a discretization of actions. Besides, an admissible and consistent heuristic reduces the computational burden of the algorithm. The idea of this work is to use the concept of Inevitable Collision States (ICS) [29] as a heuristic. An ICS is a state from which at least one collision is inevitable in the future given the available system input. When a new state is generated, it is tested for being an ICS. If this is the case, the remaining estimated costs, $h_{x,i}$ are at least the collision costs. By setting the heuristic $h_{x,i}$ to the collision costs, an admissible and consistent heuristic is found which furthermore allows to react to upcoming events which are currently ahead of the planning horizon. In the case of a movement in a one dimensional direction, the test for an ICS can be done analytically and is therefore fast enough to be used as heuristic. Formally written, a newly generated state x_i may be labeled as an ICS if and only if

$$\forall a \in \mathcal{A}, \exists E_i \in E : \{s_i + v_i \cdot t + \frac{1}{2} a t^2 | t \in [0 \ \ \infty[\} \neq \emptyset. \quad (10)$$

The concept is demonstrated in Fig. 4.

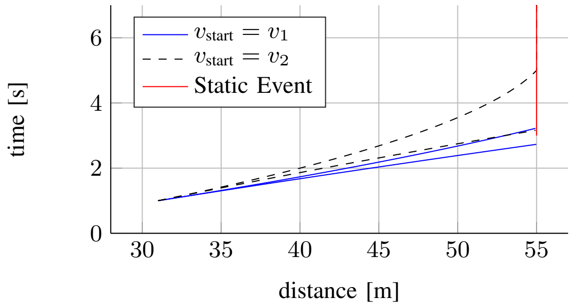


Fig. 4: Analytic calculation of the Inevitable Collision States [29], which are then used to estimate the upcoming costs in the A* graph search algorithm. It is illustrated for two different start states, with $v_1 > v_2$. It can be seen, that a collision cannot be avoided for x_2, while the static event can be avoided for state x_1.

G. Goal Formulation

A state x_i is defined as a goal state x_G if $t_G < t_{Hor}$. Nonetheless, by setting the goal state condition to a more complex equation, advanced problems may be tackled. For example, the long planner may use the moving distance to a chosen gap for a lane change as the goal state and is thereby easily capable of realizing complex gap approach behavior.

V. Evaluation

A. Implementation

The planning algorithm is implemented in BMW's Highly Automated Driving Framework and tested on a simulator as well as on a prototype vehicle on a test track. The implementation and its parameters are outlined in the following.

The driving strategy runs with a frequency of 10 Hz to be able to react quickly to changing environments. The sample step size $\Delta t = 1\,\text{s}$ and the planning horizon $t_{\text{Hor}} = 13\,\text{s}$. The set of actions $\mathcal{A}$ is defined as [-2 -1 0 1]. As the trajectory planner and the low level controller are not able to follow the goals of the driving strategy exactly, the new start state x_{start} may not lie on the last planned trajectory. This violates Bellman's Principle of Optimality which states that a optimal solution consists of optimal partial solutions [28]. Therefore, starting at a varying state may lead to a different solution under constant environment conditions which leads to unwanted driving behavior i.e. switching between maneuvers. Therefore, instead of planning from the actual, measured state $x_{\text{meas}}(t_0)$ the currently desired state, retrieved from the previous planning step, $x_{des}(t_0)$ is used as the start state x_{start}. Furthermore, to fulfill Bellman's Principle of Optimality in a discrete planning problem, the actions must be sampled at the same absolute points in time. This is impossible when sample steps of $\Delta t = 1\,\text{s}$ are used with a planning frequency $f = 10\,\text{Hz}$. Therefore, the first sampling step is not executed with Δt but with the temporal difference to the last solution's first state, i.e. $t_1^{last} - t_{\text{now}}$.

To prevent the generated graph from expanding too many nodes, only the cheapest state of two very close states is expanded. Closeness between state A and B is defined by

$$(t_A - t_B)^2 + (s_A - s_B)^2 + (v_A - v_B)^2 < 1. \quad (11)$$

B. Performance

The algorithm's performance is evaluated on a simulated round course containing four different intersections with traffic lights, various road curvatures and randomly generated traffic. The system runs on a Intel Core i7-4900MQ CPU with 2.8 Ghz. The runtime of the algorithm depends strongly on the length of the planning horizon t_{hor} and the micro traffic situation. Fig. 5 shows the runtime for a worst case scenario during driving on a evaluation circuit with four traffic lights and intersections. The average runtime is lower than the worst-case runtime by an approximate factor of 10.

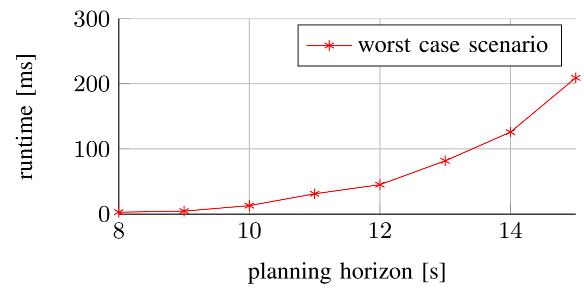
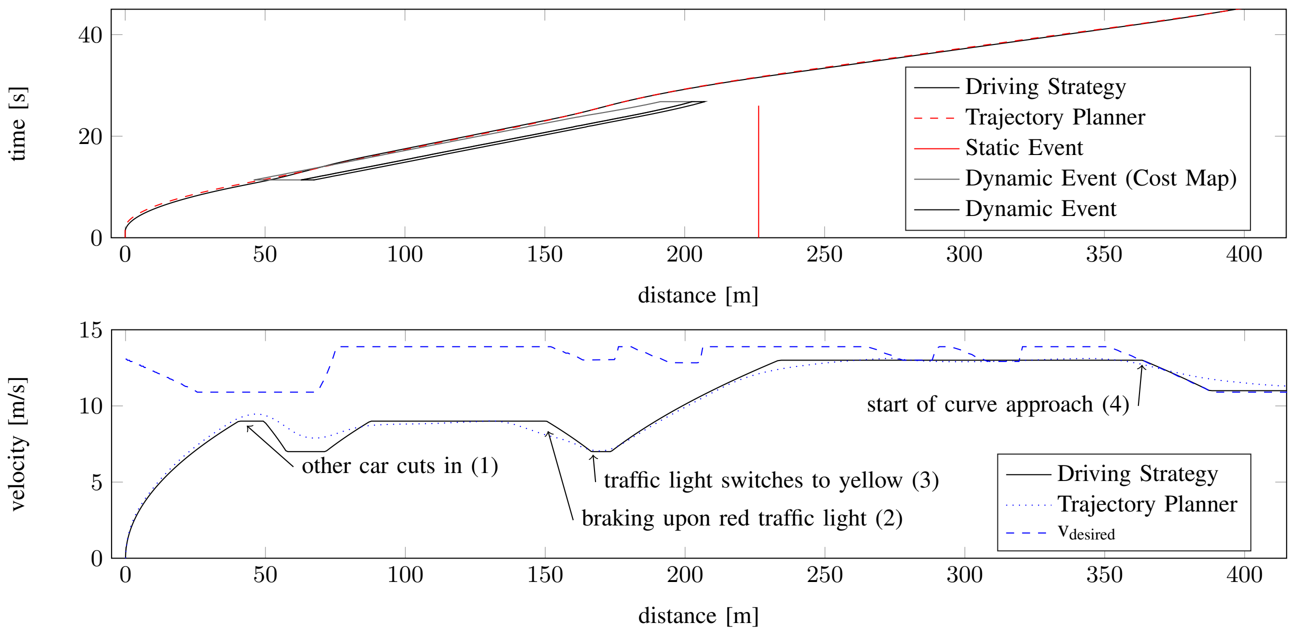


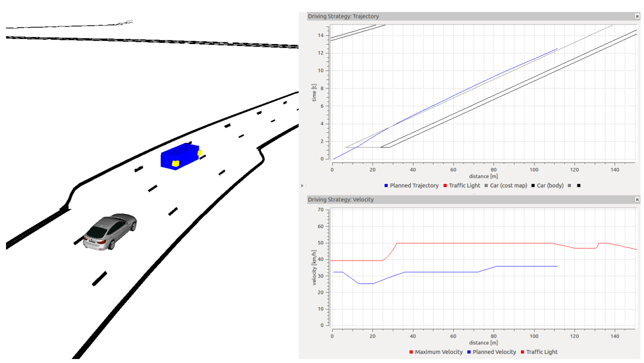
Fig. 5: Worst Case performance of the algorithm for different planning horizons t_{hor}.

The algorithm's complexity is linear regarding the number of events.

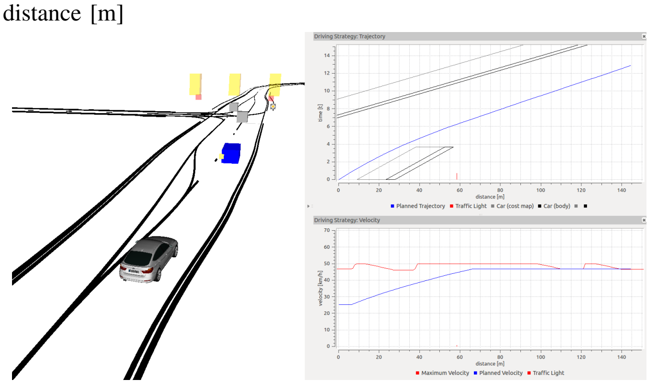
C. Qualitative Evaluation in the Simulator

To evaluate the different capabilities of the algorithm, a complex situation is evaluated in the simulation. The situation with its recorded data is shown in Fig. 6 and described in the following. The autonomous car drives along a road, when a cut-in maneuver happens (Sit. 1). Nonetheless, the other car is predicted to enter the ego vehicle's lane in two seconds, such that the ego vehicle reacts anticipatory and cooperatively by starting to decelerate already before the other car enters the lane. After that, the ego car follows the other car with ACC behavior, realized by the spatio-temporal cost map. At (2), the ego vehicle starts to brake upon a red traffic light, which is in front of the vehicle running ahead. At (3), the traffic light switches from red to yellow and the forbidden passing time (ongoing yellow period) of the traffic light is predicted to last for another second. In addition, it is detected that the preceding vehicle changes its lane to the left and it is predicted to actually leave the lane in two seconds. It can be seen in Fig. 6b, how the driving strategy optimizes its behavior over these different events. While the vehicle ahead is braking during its lane change, the ego vehicle already starts to accelerate to v_{des} as it can predict when the vehicle ahead leaves the lane and that the currently red-yellow traffic light will have switched at arrival time. It can





(a) Situation 1: A sudden cut in maneuver is detected by the autonomous car. The arrival time on the ego lane is predicted and therefore it can be reacted early.



(b) Situation 3: The traffic light ahead switches from red to yellow. Therefore, the current red-yellow phase is predicted to last for one second (duration of red-yellow phase). As the vehicle ahead is predicted to leave the ego lane in 4 seconds, the autonomous car can already start to accelerate.

Fig. 6: Evaluation situation: The autonomous car drives along a road and has to react to several events (see section V-C).

also be seen, that the maximum velocity is constrained by the slight road curvature, such that v_{des} is lower than v_{law}. In this situation, the algorithm optimizes its behavior under consideration of other vehicles, a currently switching traffic light and the road's curvature.

D. Qualitative Evaluation on the test track

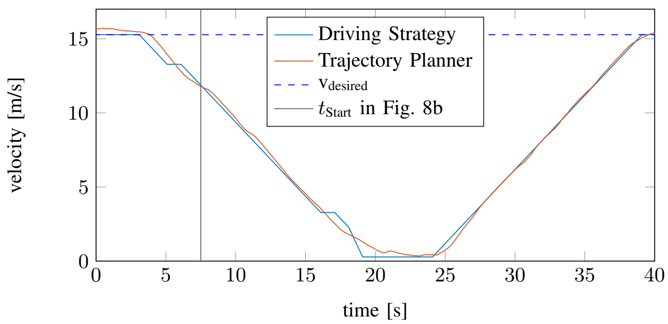
In addition to the very complex scenario in the simulator, the presented algorithm was also evaluated on the test track in a prototype vehicle of BMW. The demonstrated situation

Fig. 7: The prototype vehicle and a screenshot of the visualization during the drives on the test track.

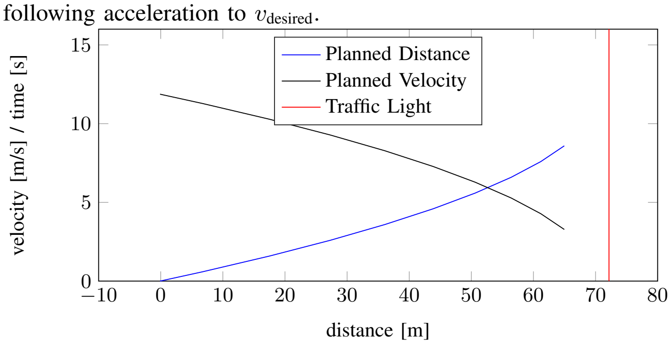
is as follows. The autonomous car is driving with constant velocity and has to react to a simulated, red traffic light. After full deceleration and standstill, the traffic light switches to green again and the acceleration behavior of the vehicle can be seen. This situation is demonstrated in Fig. 8. While Fig. 8a shows the velocity during the whole evaluation process, Fig. 8b shows exemplary the found optimal maneuver during planning at $t = 7.5s$. While it can be seen how reliable the planned and desired behavior of the driving strategy is, another aspect is also of hight interest. While the static event is beyond the planning horizon ($t_{\text{Hor}} = 10\,\text{s}$ in this illustrative case), the deceleration maneuver is nonetheless already started as the Inevitable Collision State heuristic leads the A* search to a deceleration maneuver.

VI. Conclusion

This work presents a novel driving strategy based on a globally comfort-optimized planner under constraints of traffic rules, curvatures and different events along the road. The result is an optimal speed profile which parameterizes



(a) Velocity during the approach to a red traffic light and the following acceleration to $v_{desired}$.



(b) Optimal selected maneuver during the approach at $t = 7.5\,\mathrm{s}$.

Fig. 8: Recorded data during evaluation in the prototype car.

the trajectory planner by providing feasible, law conform and comfortable goal states. The algorithm is real-time capable and has been demonstrated to handle various situations by providing optimized solutions.

In the future, the algorithm will be extended in two ways. On the first hand, the idea is to include lateral actions such that lane changes and/or passing objects, which extend into the lane, can be realized. The longitudinal approach may be extended by providing a third event type which represents lane areas which should not be occupied for a given time. This is e.g. the case for cycle lanes, on which the autonomous car should not stand during stop and go traffic situations.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Commission Seventh Framework Programme (FP/2007-2013) under the project AdaptIVe [9], grant agreement number 610428. Responsibility for the information and views set out in this paper lies entirely with the authors. The author(s) would like to thank all partners within AdaptIVe for their cooperation and valuable contribution.

REFERENCES

[1] SAE International, "Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems," SAE International, Tech. Rep. SAE J 3016, 2014.

[2] M. Aeberhard, S. Rauch, M. Bahram *et al.*, "Experience, Results and Lessons Learned from Automated Driving on Germany's Highways," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 42–57, 2015.

[3] Audi USA. (2015) Mission Accomplished: Audi A7 Piloted Driving Car Completes 550-mile Automated Test Drive. [Online]. Available: https://www.audiusa.com/newsroom/news/

[4] Bosch Group. (2015) Cars that Drive Themselves - Highway Pilot Technically Viable in Five Years. [Online]. Available: http://www.bosch-presse.de/

[5] J. Ziegler, P. Bender *et al.*, "Making Bertha Drive - An Autonomous Journey on a Historic Route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[6] Google Official Blog. (2014) Just Press Go: Designing a Self-Driving Vehicle. [Online]. Available: http://googleblog.blogspot.com/2014/05/just-press-go-designing-self-driving.html

[7] M. Buehler, K. Iagnemma, and S. Singh, Eds., *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, ser. Springer Tracts in Advanced Robotics, vol. 56. Springer, 2009.

[8] C. Urmson, J. Anhalt *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," vol. 25, no. 8. Wiley Online Library, 2008, pp. 425–466.

[9] AdaptiVe. [Online]. Available: http://www.adaptive-ip.eu/

[10] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 987–993.

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

[12] D. González, J. Pérez, V. Milanés, 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.

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

[14] P. Bender, Ö. Ş. Taş, J. Ziegler, and C. Stiller, "The combinatorial aspect of motion planning: Maneuver variants in structured environments," in *IEEE Intelligent Vehicles Symposium*, 2015, pp. 1386–1392.

[15] C. Stiller, G. Färber, and S. Kammel, "Cooperative cognitive automobiles," in *IEEE Intelligent Vehicles Symposium*, 2007, pp. 215–220.

[16] A. Lawitzky, D. Wollherr, and M. Buss, "Energy optimal control to approach traffic lights," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013, pp. 4382–4387.

[17] A. Mazzalai, F. Biral, M. D. Lio, M. Darin, and L. D'Orazio, "Automated crossing of intersections controlled by traffic lights," in *IEEE Int. Conf. on Intelligent Transportation Systems*, 2015, pp. 1928–1933.

[18] R. Kohlhaas, T. Schamm, D. Nienhser, and J. M. Zöllner, "Anticipatory energy saving assistant for approaching slower vehicles," in *IEEE Int. Conf. on Intelligent Transportation Systems*, 2011, pp. 1966–1971.

[19] D. Zhao, "Supervised adaptive dynamic programming based adaptive cruise control," in *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 2011, pp. 318–323.

[20] M. Ardelt, C. Coester, and N. Kaempchen, "Highly automated driving on freeways in real traffic using a probabilistic framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1576–1585, 2012.

[21] M. Werling and L. Gröll, "Low-level controllers realizing high-level decisions in an autonomous vehicle," in *IEEE Intelligent Vehicles Symposium*, 2008, pp. 1113–1118.

[22] R. Kohlhaas, T. Schamm, D. Lenk, and J. M. Zöllner, "Towards driving autonomously: Autonomous cruise control in urban environments," in *IEEE Intelligent Vehicles Symposium*, 2013, pp. 116–121.

[23] R. Kohlhaas, T. Bittner, T. Schamm, and J. M. Zöllner, "Semantic state space for high-level maneuver planning in structured traffic scenes," in *IEEE Int. Conf. on Intelligent Transportation Systems*, 2014, pp. 1060–1065.

[24] R. Kohlhaas, D. Hammann, T. Schamm, and J. M. Zöllner, "Planning of high-level maneuver sequences on semantic state spaces," in *IEEE Int. Conf. on Intelligent Transportation Systems*, 2015, pp. 2090–2096.

[25] J. Johnson and K. Hauser, "Optimal acceleration-bounded trajectory planning in dynamic environments along a specified path," in *IEEE Int. Conf. on Robotics and Automation*, 2012, pp. 2035–2041.

[26] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 72–89, 1986.

[27] S. M. LaValle, "Planning algorithms," 2004.

[28] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009.

[29] T. Fraichard and H. Asama, "Inevitable collision states. a step towards safer robots?" in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2003, pp. 388–393 vol.1.