Towards Tactical Lane Change Behavior Planning for Automated Vehicles

Simon Ulbrich and Markus Maurer

***Abstract*— Recently, automated driving has more and more been transformed from an exciting vision into hands on reality by prototypes. While drivers are used to assistance and maybe even automation for driving *within* a lane, it is exciting to dare a step ahead: Deciding and executing tactical maneuvers like lane changes in automated vehicles without any human interaction. In this paper, we present our approach for tactical behavior planning for lane changes. We present a way to tackle perception uncertainties and how to achieve provident, prediction-based behavior planning. For this, we introduce a novel framework to plan in high-dimensional, mixed-integer state spaces in real-time. Our approach is evaluated not only in simulation, but also in real traffic. The implementation has recently been demonstrated to the public in the *Audi A7 piloted driving concept* vehicle, driving from Stanford to the Consumer Electronics Show (CES) 2015 in Las Vegas.**

I. Introduction

Technology affine drivers are used to handing over some mainly stabilization level related driving tasks to driver assistance systems as an adaptive cruise control or lane keeping system. The next big step is to automate behavior planning on a tactical level. Among tactical behavior planning tasks are decision making, whether lane changes are beneficial and possible, if a vehicle should execute some cooperative driving behavior, or particular maneuvers in intersection handling. Tactical driving behavior has to consider the (perceived), current driving situation and addresses a planning horizon of something between 100 ms and 30 s. It is often attributed as the *intelligence* of the automated vehicle.

The central challenge for tactical behavior planning is the complexity of real world traffic situations and the uncertainty in its perception with todays imperfect sensor systems. Tactical behavior planning for automated driving requires rapidity, consistency, providentness and determinism. To achieve this, we present a framework for tactical behavior planning in uncertain, mixed-integer state spaces. We introduce a distinction into a measurement model, prediction model and reward model and present implementations for each of these.

For that purpose, this paper is structured as follows: In section II we define requirements and illustrate necessary scenarios to be handled by a tactical behavior planning for lane changes. Section III presents the framework for lane change behavior planning. This is followed by an evaluation in a simulation environment as well as in real traffic in section IV. Last of all, section V finalizes this paper with conclusions and a research outlook.

II. Background

Donges [1] defined different levels of driving tasks. He introduced the classification of driving tasks into navigation tasks (strategic level), guidance tasks (tactical level) and stabilization tasks (operational level). The focus of this paper are guidance tasks, which entail any tactical behavior planning of an automated vehicle. Figure 1 illustrates tactical behavior planning as a part of the overall architecture of an automated vehicle.

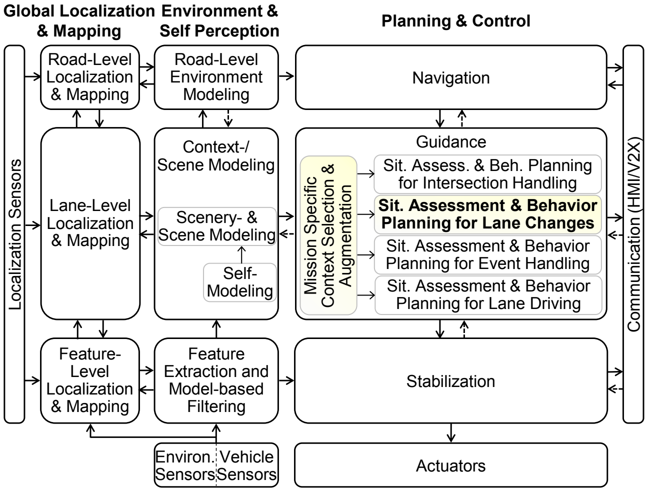
Global Localization & Mapping
Environment & Self Perception
Planning & Control
Localization Sensors
Road-Level Localization & Mapping
Road-Level Environment Modeling
Navigation
Lane-Level Localization & Mapping
Context-/ Scene Modeling
Scenery- & Scene Modeling
Self-Modeling
Guidance
Mission Specific Context Selection & Augmentation
Sit. Assess. & Beh. Planning for Intersection Handling
Sit. Assessment & Behavior Planning for Lane Changes
Sit. Assessment & Behavior Planning for Event Handling
Sit. Assessment & Behavior Planning for Lane Driving
Communication (HMI/V2X)
Feature-Level Localization & Mapping
Feature Extraction and Model-based Filtering
Stabilization
Environ. Sensors
Vehicle Sensors
Actuators

Fig. 1. Tactical behavior planning for lane changes as a part of the overall functional system architecture of an automated vehicle (cf. [2], [3], [4])

A. Related work

A review of relevant literature has been presented in Ulbrich & Maurer [5] already. Therefore, the focus is limited to some particular aspects of lane change planning and gap selection not yet covered in [5]. In simulated environments, the development of driving behavior models, especially car-following models for simulation environments has been a focus of research for several decades. Literature is vast and has extensively been reviewed by e.g. [6], [7], [8]. Extensions of these models exist to address lane changing and gap selection aspects, in which cooperative behavior is a central topic. Gipps [9] proposes a framework for lane change decision making in sub-urban driving situations. Hidas [10] extends the Gipps-model [9] by a gap quality evaluation. In contrast to real world implementations, most simulation models simplify the world by simulating “lane changing [...] as an instantaneous action” [11, p. 46]. Thus, any kind of options like aborting an already started maneuver are not

S. Ulbrich and M. Maurer are with the Institute of Control Engineering, Technische Universität Braunschweig, Hans-Sommer-Str. 66, 38106 Braunschweig, Germany {ulbrich, maurer}@ifr.ing.tu-bs.de

considered. Moreover, no kind of uncertainty is modeled or considered. Hence, all decisions are based on perfect knowledge about the other vehicle's states, intentions and willingness to cooperate.

Ardelt et al. [12], [13] present BMW's lane change approaches in its *ConnectedDrive*-project, focused on highly automated driving on highways. A hybrid, deterministic state machine is used to define the superordinate driving behavior and a decision tree is used as a hierarchical decision making process. The superordinate state is determined by traversing the decision tree, depending on the driving goal derived by the situation interpretation and the current feasibility of maneuvers.

Ruf et al. [14] develop the SPARC framework for behavior and trajectory planning. The prediction model is separated from the reward model. So far, it has only been applied on perfect, simulated data but seem to scale well with real, uncertain sensor data.

Brechtel et al. [15] showed a way of using Markov decision problems for lane change decision making. They based their decision process' state variables directly on measured values like relative distances and velocities toward surrounding vehicles. Brechtel et al. [16] used a learning algorithm to translate a value-continuous intersection merging problem into a discrete approximation of the same to be solved by a partially observable Markov decision process (POMDP). They evaluated it based on simulated data.

The authors presented and evaluated a lane change decision making approach based on a partially observable Markov decision process [5]. Here, the state space was discretized in order to render Branch and Bound based methods feasible. In [4], appropriate evaluation metrics and a detailed evaluation has been presented.

In this paper, we no longer discretize the state and observation space. We allow high-dimensional, mixed-integer state spaces with uncertainties as they occur in the real world. Only actions remain as value-discrete tactical choices (either do a lane change or don't).

B. Requirements

Figure 2 illustrates a typical decision making problem for performing lane changes in urban environments. Whether a lane change is *possible* depends on the relative distances, velocities and accelerations of other vehicles around the ego vehicle. Whether a lane change is *beneficial* depends on the road network, the mission and the behavior of other vehicles around the ego vehicle. As stated in [5], behavior planning for automated driving requires rapidity, coherency, providentness and determinism.

- *Rapidity:* Tactical behavior planning needs to be fast. Despite some strategic decisions (e.g. route recalculation) may take more time, at least tactical behavior planning needs to be taken fast (<100ms).
- *Consistency:* A decision should fit in the framework of previous decisions. Similar to a human driver, a behavior planning module should not constantly change its mind about the choice of driving maneuvers. All decisions should align well with a long term goal.
- *Providentness:* Behavior planning should have some foresight to predict how the situation will look like after a maneuver execution or the elapse of some time.
- *Determinism:* Last of all, decision making should be predictable in a sense that it can be tested and validated according to functional safety requirements.

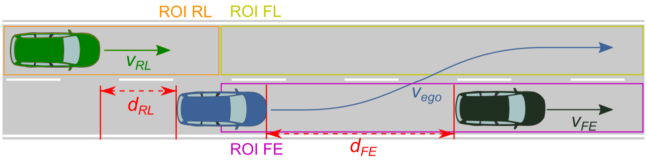

Fig. 2. Typical scenario for lane change decision making with two dynamic objects and three regions of interest rear left (RL), front left (FL) and front ego (FE)

C. MDPs, POMDPs and SN-MPCs

Markov decision processes (MDPs) are a general framework to model planning and decision making problems. Executing an action $u \in U$, given the system is in state $x \in X$, is what will be called as a part of a policy $\pi : x \to u$. The goal of such a planning problem is to find an optimal policy (sequence of actions) π^* that maximizes the expected reward r_τ with discount factor γ over the time horizon T:

$$R_T = E[\sum_{\tau=0}^{T} \gamma^\tau * r_\tau] \quad (1)$$

Anyhow, true system states are typically not observable. Partially observable Markov decision processes (POMDP) help to accommodate this issue by the introduction of the idea of a belief $bel(x_t)$ of being in a state x_t at time t.

A POMDP is represented by the tuple (X, U, T, R, Z, O) where:

- X is the set of all the environment states x_t at time t.
- U is the set of all possible actions u_t at time t.
- T is the $X \times U \times X \to [0, 1]$ is the transition function, where $T(x_t, u_{t-1}, x_{t-1}) = p(x_t|u_{t-1}, x_{t-1})$ is the probability of ending in state x_t if the agent performs action u_{t-1} in state x_{t-1}.
- R is the $X \times U \to \mathbb{R}$ is the reward function, where $r(x, u)$ is the reward obtained by executing action u in state x.
- Z is the set of all measurements or observations z_t at time t.
- O is the $X \times U \times X \to [0, 1]$ is the observation function, where $O(x_t, u_{t-1}, z_{t-1}) = p(z|u, x)$ give the probability of observing z if action u is performed and the resulting state is x.

Typically the set of states X, actions U and measurements Z are modeled *value-discrete*. This increases the computational complexity and therefore POMDPs are often avoided for real-time applications.

In model predictive control (MPC), the set of states X and actions U are typically assumed to be value-continuous *only* and the models R and T are assumed to be linear or quadratic. There are computationally efficient solution methods for finite, (receding) optimization horizons. Extensions to the general model exist to consider non-linear, e.g., mixed-integer state spaces and uncertain measurements Z. Mixed-integer POMDPs and stochastic, non-linear model predictive control (SN-MPC) are essentially two overlapping frameworks to model similar technical challenges.

The presented implementation will use the separated, non-linear, mixed-integer observation O, prediction T and reward R models of a POMDP to handle uncertainties and non-linearities. It will make use of a finite, receding optimization horizon as in an MDP to render online solution methods feasible.

III. Behavior Planning Framework

This section presents a novel framework for tactical behavior planning in uncertain, mixed-integer state spaces. The planning framework consists of four components components; it is illustrated in figure 3. A measurement model to account for uncertainties in the transformation of measurements into state estimates, a planning core to address the behavior planning and decision making and a situation prediction model and a reward model to support the planning core in its behavior planning.

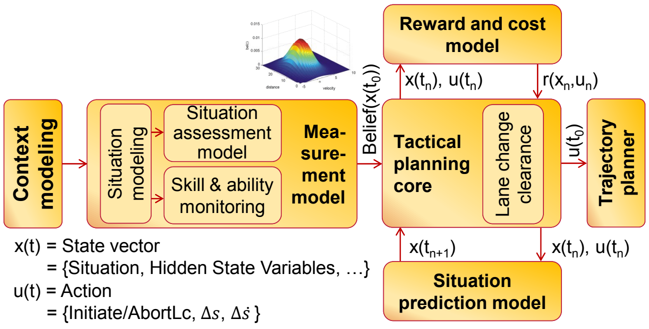

Fig. 3. Behavior planning framework

A. Measurement Model

The task of the measurement model is to translate observations about the driving situation into an aggregated belief on the system's state. Observations entail value-discrete (e.g., number of lanes) and value-continuous aspects (e.g., distance to a front vehicle). Some aspects of the system state are observable (e.g., how long the indicator has been switched on already), and hidden. Hidden state variables may contain information, whether a lane change seems possible or beneficial in the current situation, which gap is the best to head to, etc.

Figure 4 illustrates three different stages within the measurement model. The leftmost part of the image depicts a visualization of the context model as an abstract scene description of the vehicle itself and its environment. The next part visualizes a situation representation of lane change relevant information. A dynamic Bayesian network is used to obtain beliefs for the distributions of hidden state variables. The rightmost part of figure 4 depicts the dynamic Bayesian network.

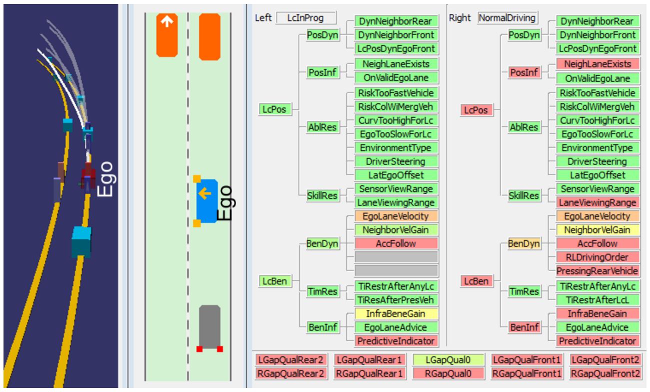

Fig. 4. Estimation of situation aspects by a Bayesian network. Left: scene, center: situation, right: visualization of dynamic Bayesian network

For planning lane changes, the four high-level hidden state variables are, whether a lane change is *possible* and *beneficial* to the *left* and *right* respectively. To calculate state estimates for those abstract hidden state random variables, several other underlying random variables need to be calculated:

1) Lane Change Possible Estimation: To estimate if a lane change is possible, we have to consider if it is possible due to the dynamic traffic situation, due to the infrastructure, due to ability induced skill restrictions and due to the system's current skill-level induced skill restrictions.

2) Lane Change Beneficial Estimation: To estimate if a lane change is beneficial, it we need to evaluate the dynamic traffic situation for relative *velocity gains* on neighbor lanes and if a lane change is beneficial due to infrastructure related information.

3) Gap Quality Assessment: High traffic densities necessitate to adjust the automated vehicle towards a cost-optimal gap. Therefore, the most appropriate gap for a lane change is determined.

4) Calculating and Propagating Uncertainties: Among the key challenges for tactical lane change behavior planning is the inherent uncertainty from any kind of environment perception modules. State estimates from perception modules come along with an uncertainty already. Based on this, hidden state variable estimates are calculated by the dynamic Bayesian network. To obtain a variance estimate for those as well, we use an unscented transform with a minimal set of sigma points in the same way it is used in an Unscented Kalman filter [17, p. 65].

Details about the measurement model are provided in Ulbrich & Maurer [18].

B. Tactical Behavior Planning Core

The behavior planning core uses the state believes to derive behavior decisions. It utilizes the reward and situation prediction models as illustrated in figure 3. It decides about -to the reward model- optimal tactical behavior actions and commands those to a subordinate trajectory planning module.

POMDP-based and stochastic non-linear model predictive control approaches are often not applicable, due to their computational complexity. Luckily, a lot of domain knowledge can be incorporated in the action selection process to tailor the decision process to the particular issue:

- Planning horizons are relatively short: It is typically not possible to make long-term predictions of more than maybe 10 s, anyway. Hence, there is no need for a high, possibly infinite planning depth T.
- Action alternatives are sparse: Typically several variations of the same maneuver exist, but only very few mutually exclusive, discrete action alternatives.
- Mixed observability: Some internal states (e.g., if a lane change is in progress) are free of uncertainty. Hence, they reduce the model complexity by a lot and may even rule out some action alternatives at all.
- Limited planning accuracy needed in the far future: After all, only the immediate, next action will be propagated to the subsequent modules, e.g., for trajectory planning. Hence, there is no need for a detailed plan for the far future.

The authors apply a tree-based policy evaluation to make use of this domain knowledge. Tree-based policy evaluation methods uses tree of beliefs b and actions u that can be obtained and executed in each time step. Figure 5 illustrates such a belief tree [19], while simplifying that one action might result in several beliefs. Based on a current belief b_0 at a time step t_0 several actions $u^{(i)} \in U$ can be executed and result in a reward $r(b_0, u^{(i)})$. Given a certain action will be executed, new observations will occur $z^{(i)} \in O$ resulting in a new, future beliefs $b_1, ..., b_5$ at time step $t_{0+\tau}$. These new beliefs are once more root nodes of some subtrees for this future time slice. The total reward R_T is calculated along each path to a fringe node representing the planning horizon T. A path represents a sequence of future actions. The path with the highest reward is selected.

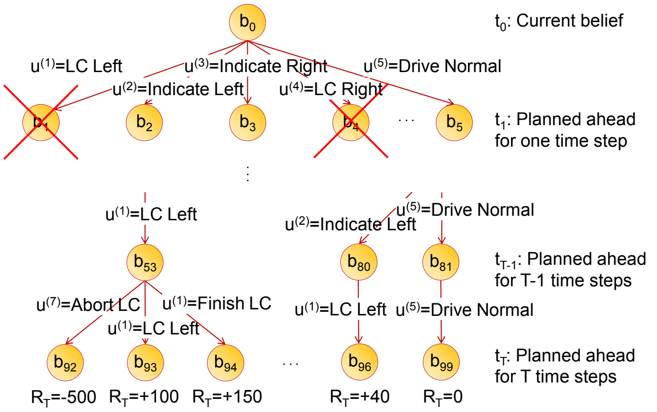

Fig. 5. Policy tree of (predicted) state beliefs and actions; LC=lane change

The tree size grows with the number of value-discrete actions and orthogonal observations. As an approximation, it is possible to consider only the most likely parametric description of a measurement variable distribution and its subsequent belief distribution. By this, measurement variables probability distributions can be considered, as long as they can be described by *one* parametric distribution. Therefore, it is for instance possible to model that cooperative breaking intensities of traffic participants vary by a probability distribution, but not that some driver might be inattentive, gradually approaches and finally crashes into the automated vehicle without reacting at all. For the real world, this limitation is not severe, as the authors are not aware of any way to estimate those behavior likelihoods in such a mixture-of-probability-distributions random processes, anyway.

Figure 5 shows a tree simplification for not pursuing action $u^{(1)}$ or $u^{(4)}$ while having belief b_0. This is to show that some actions can be ruled out because of not being allowed (changing lanes without indicating, given belief b_0 represents normal driving). In fact, it is possible to prune the tree even further by ruling out actions would result in unreasonable policies (sequence of actions). E.g., if a lane change was decided it will be an unreasonable policy to abort a lane change and reinitiate a second lane change in two consecutive time steps. This helps to reduce the tree complexity a lot. Every path to a fringe node at end of the planning horizon at time step $t_0 + T$ will be a possible -to a certain degree- reasonable policy. Both tree simplifications reduce the number of policies to be evaluated by a lot as demonstrated in section IV-C.

The set of actions U contains the 13 discrete action alternatives of *DoLc, FinishLc, PrepareLc, IndicateLc* and *AbortLc* to the left and right and action alternatives for *NormalDriving, AbortLcIndication* and *AbortLcPreparation*. Further more it entails a *targetGapIndex* and continuous dimensions for gradual deviations like a longitudinal delta target pose and sampling variation as well as a commanded velocity deviation from the current ego velocity.

C. Multi-resolution planning

Another key idea to render online solution methods tractable for the presented behavior planning is to incorporate the concept of multi-resultion planning.

Earl et al. [20] use an iterative refinement pattern for robot path planning with mixed integer linear programming. Culligan [21, p. 26] extends this idea by using a variable time discretization for a trajectory planning application in unmanned aerial vehicles.

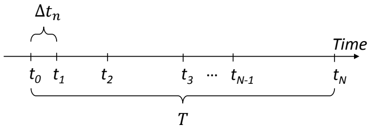

Fig. 6. Variable planning time steps

A common way to discretize the planning horizon T is to use a *fixed* time step $\Delta t_n = \Delta t = T/N$. However, it is only necessary to plan with a high temporal accuracy in the immediate future. In this paper, the authors use a time resolution pattern as illustrated in figure 6. At first, a time step of $\Delta t = 500\,ms$ is used. After the first prediction/action selection step, a time step of $2 \cdot \Delta t = 1\,s$, $3 \cdot \Delta t = 1.5\,s$, ..., is used. The overall planning horizon T is $14\,s$.

D. Situation prediction model

The situation prediction model facilitates a prediction of the entire situation as a function of the former situation and an action: $bel(x_{t+\tau}) = p(bel(x_t), u_t)$. The situation prediction as a whole constitutes of the prediction of several aspects of that situation. Among them are simple dynamic models for the prediction of object movements, behavior models to imitate the interaction between vehicles and simplified models to predict the ego behavior. A plethora of situation prediction models have been proposed. The focus of this paper is not to find the best one, but rather to present a framework to make use of any of them for actual behavior planning.

We use an improved intelligent driver model as in Shen et al. [22]. It is based on predicting a longitudinal acceleration and by this calculating new longitudinal velocity and position of each object based on its environment. For the lateral prediction, we assume that vehicles will maintain their lateral offset towards their lanes. Lanes are predicted to continue the way they are perceived based on a floating clothoid model.

Planned actions may change the situation itself. For instance, if the abortion of a lane change is commanded to the prediction model, it will change the prediction of the ego vehicle as well as of the rest of the situation.

E. Reward model

The reward model calculates a reward $r(bel(x_t), u_t)$ as a function of the system state belief $bel(x_t)$ and an action u_t. In a traditional POMDPs and MPCs, such a reward is a single numeric value. However, it proved useful to return a vector. By this, it is possible to aggregate the lane change possible reward dimension separate from the lane change beneficial or the gap selection reward dimension over time. Thus, the decision making algorithm may distinguish if a positive overall reward results from whether a lane change is possible and/or beneficial.

Moreover, the reward model incorporates a decision hysteresis for several aspects of the system's state belief $bel(x_t)$. Therefore, similar to the measurement and prediction model it yields a non-linear mapping.

IV. Evaluation

This section presents our evaluation results. After a short presentation of an evaluation in a simulation environment, we present an evaluation in real traffic.

A. Evaluation in a Simulation

During the development process and for validation of the algorithms, a simulation environment is a crucial part. We use a tool chain of *Virtual Test Drive (VTD)*[1] and *Automotive Data and Time-triggered Framework (ADTF)*[2] to test the presented algorithms. Figure 8 illustrates a scenario where a lane change is executed to overtake a slow vehicle.

B. Evaluation in Real Traffic

The best proof for the feasibility of a concept is its evaluation in real traffic. The algorithms have been tested and tweaked in the *Audi A7 piloted driving concept* vehicles for about 60.000 km in public traffic. The lane changing behavior has recently been demonstrated to the public in the 550 mile drive from Stanford to Las Vegas to the Consumer Electronics Show 2015[3] and on a German highway[4]. The focus of our efforts has been on highways, but it has also been tested on (sub-)urban multilane streets.

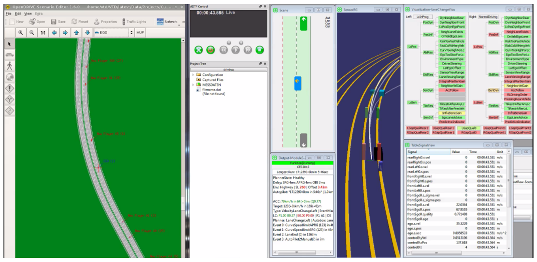

Fig. 8. Screenshot of a scenario-based closed loop testing in *VTD*

The first two images in figure 9 illustrate driving a 20 km stretch of the A9 from Ingolstadt, Germany northbound. The first diagram depicts the longitudinal ego velocity of the automated vehicle. There is no speed limit on this stretch of a 3+3 lane highway. The target velocity is set to $40\,m/s$. Occasionally, the automated vehicle gets slowed down by traffic in front, if it is not able to perform a lane change due to traffic from behind on the neighbor lanes.

The second and third diagram of figure 9 present the lateral offset of the automated vehicle to the center of the ego lane. As the ego lane jumps to another lane, every time a lane change is performed, the lateral offset jumps from positive to negative (lane change left) or negative to positive (lane change right). The third till sixth plot of figure 9 illustrate a single overtaking maneuver during the $20\,km$ drive in more detail.

The maneuver is visualized by a sequence of images from the lane tracking camera and a situation visualization widget in figure 7. Initially, the automated vehicle drives on the rightmost lane of a highway. In front of it appears a slow truck (green). As overtaking on a highway in Germany is only allowed on the left, and the grey vehicle from behind on the left neighbor lane is sufficiently far away (arrow pointing backwards) and slow enough (grey color), the automated vehicle (blue) activates the indicator (yellow indicator lights) and initiates a lane change to the middle lane (visualized by the yellow arrow). The automated vehicle speeds up slightly and overtakes the truck. While driving on the middle lane, the automated vehicle gets overtaken by a very fast vehicle on the leftmost lane (red). After the truck has been passed and no other truck appears on the horizon on the rightmost lane, the automated vehicle changes back to the rightmost lane to obey the right lane driving order in German traffic regulations.

The last three diagrams in figure 9 illustrate internal state variables of the lane change decision making. First of all, the state estimates for the hidden state variables *lane change possible* and *lane change beneficial* (c.f. section III-A) are

[1] VTD: http://www.vires.com

[2] ADTF: https://automotive.elektrobit.com/ products/eb-assist/adtf/

[3] http://www.audi.com/content/com/brand/en/vorsprung_durch_technik/content/2014/10/piloted-driving.html

[4] http://www.stern.de/auto/news/jack-das-selbstfahrende-auto-von-audi-erstmals-auf-einer-deutschen-autobahn-2174446.html

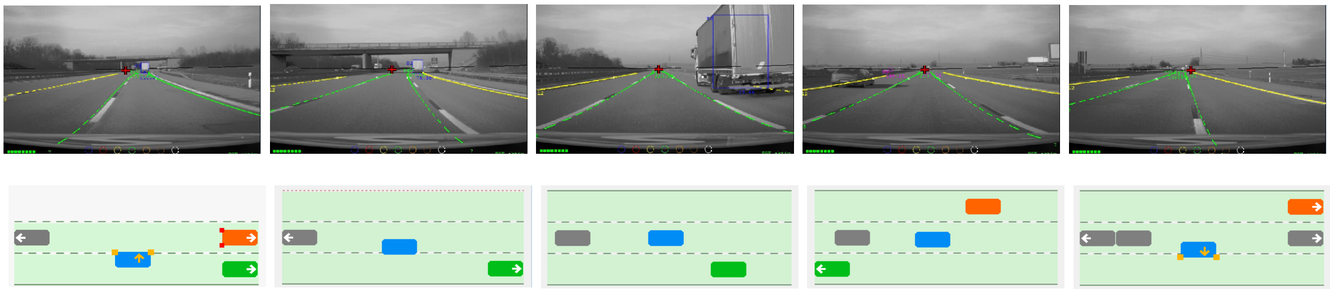

Fig. 7. Video and situation visualization for an overtaking scenario.

illustrated. The closer the automated vehicle approaches the slow, green truck on the rightmost lane, the more beneficial a lane change to the left becomes. As a lane change is also possible (threshold is around 0.5), a lane change is indicated (*lane change state* changes to $+9$). The indication phase is followed by the actual lane change to the left itself (*lane change state*$= +1$). The lateral offset to the center of the ego lane $d_{pos,ego}$ increases until the ego lane switches to the middle lane. The automated vehicle gets overtaken by a fast vehicle on the leftmost lane. Hence, the *lane change possible* state estimate decreases between $t = 238\,s$ till $t = 243\,s$. Due to object mismatches, *lane change possible left* decreases at $t = 248\,s$. Directly after the first lane change to the left, *lane change possible left* temporarily drops because a lane change is only considered possible after a proper re-centering to a lane to avoid challenges for the lane tracking modules. After the lane change to the left has been completed and the green truck has been overtaken *lane change beneficial right* rises from $t = 240\,s$ onwards. At $t = 245\,s$ a lane change to the right is first indicated (*lane change state* changes to -9) and then executed (*lane change state* changes to -1).

C. Runtime

The algorithm runs in real time on an Intel i7 4800MQ CPU sharing resources with trajectory planning, situation modeling and visualization modules. Typical peak loads for any of the cores are below 20%. Per cycle, we evaluate on average about 80 pathes in the tree. At worst, the number of evaluated pathes will grow to 200 and it will still take less than in peak $4\,ms$ to evaluate them.

D. Current Limitations

Although many guests in our vehicle judged the presented algorithms to perform very well already, there are still several limitations. A central performance determinant is the traffic density. The denser the traffic is, the more likely is the automated vehicle to not finding a sufficiently large gap, or to abort an already initiated lane change due to false object detections or track-to-lane associations.

Another challenge for our implementation are the current sensor systems to the side. Other than e.g. [13] or [23] our vehicle is just equipped with low-cost, series production radar sensors to detect vehicles directly at its side. Thus, many false positive and false negative detections are caused by these sensors. A lidar, camera or better radar system would clearly help to reduce the number of aborted lane changes and to merge into smaller gaps as a whole.

Also very challenging is the gap adjustment. Since the object detection to the front and rear is currently based on lidar sensors it is hard to differentiate trucks from cars as typically only the wheels or undercarriage is seen. Frequently, there seems to be an appropriate gap between the reflection of a truck's trailer and the driver compartment while still being, e.g., 60 m away. Approaching such an assumed gap, the contour estimation of the neighbor vehicle improves and the front and rear of the truck are perceived as one truck without a gap in between. Thus, a gap approach needs to be overthrown and a better gap has to be found. A passenger judges this as an illogical gap approach, because based on his perception, he does not face the same perceptual limitations as the automated vehicle.

Another limitation are currently highway interchanges and on- and off-ramps. As our vehicle is designed to work without highly accurate GPS-systems and detailed, highly accurate maps, we are limited by what a camera and the lane detection algorithms perceive. Thus, weaving and merging areas often appear to be shorter than for a human driver. If some interaction with other vehicles in those weaving areas does not turn out to work smoothly, it may happen, that we do not have sufficient remaining maneuver space for lane changes.

V. Conclusions

In this paper, the authors presented a novel framework for tactical lane change behavior planning for automated vehicles. It allows to plan a sequence of actions in uncertain, high-dimensional, mixed-integer state spaces in real-time. A measurement model allows to handle uncertain perception information, a prediction model yields consistent and provident behavior planning. A fixed model results in predictive, deterministic behavior.

The paper evaluates the algorithms' performance in a simulation environment and online in real traffic.

Despite demonstrating a solid performance already, several areas of improvement exist. So far, our prediction model is yet relatively crude. Particularly in complex, highly interactive traffic situations it does not provide accurate predictions of the other drivers' tactical maneuvers. The uncertainty estimation is not yet fully considered in the reward model and is still lacking regarding an objects' uncertainty of existence or especially modeling the perception uncertainties of the

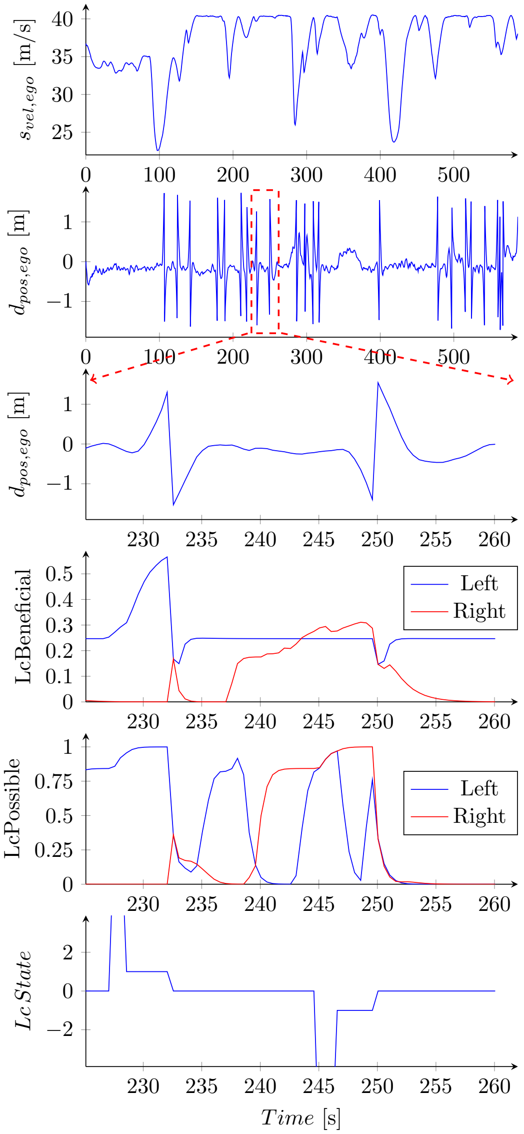

Fig. 9. State variables during an overtaking scenario

sensor systems. The gap adjustment is still limited, mainly by the persistent detection of objects and gaps.

Acknowledgment

The authors thank Volkswagen Group Research, Volkswagen of America and Audi for their financial and/or operational support. Special thanks to our colleagues Simon Grossjohann, Christian Appelt, Kai Homeier and Bernd Rössler for spending hours and months on the test track and on public roads to tune the algorithms.

References

[1] E. Donges, “A Conceptual Framework for Active Safety in Road Traffic,” *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, vol. 32, no. 2-3, pp. 113–128, Aug. 1999.

[2] R. Matthaei and M. Maurer, “Autonomous Driving - A Top-Down-Approach,” *at - Automatisierungstechnik*, vol. 63, no. 3, pp. 155–167, 2015.

[3] A. Reschka, J. R. Böhmer, J. Gačnik, F. Köster, J. M. Wille, and M. Maurer, “Development of software for open autonomous automotive systems in the stadtpilot-project,” in *8th International Workshop on Intelligent Transportation (WIT2011)*, Hamburg, Germany, 2011.

[4] S. Ulbrich and M. Maurer, “Evaluation einer taktischen Verhaltensentscheidungsfindung für Fahrstreifenwechsel beim vollautomatisierten Fahren in Städten (English title: Evaluation of tactical behavior decision making for lane changes in fully automated driving in cities),” in *9. Workshop Fahrerassistenzsysteme FAS 2014*, B. Färber, Ed., Walting, 2014, pp. 147–158.

[5] S. Ulbrich and M. Maurer, “Probabilistic online POMDP decision making for lane changes in fully automated driving,” in *Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, The Netherlands, Oct. 2013, pp. 2063–2067.

[6] T. Muller and H. van Zuylen, “A Driving Behavior Model for Microscopic Online Simulation based on Remote Sensing and Equipped Vehicle Data,” in *Transportation Research Board Annual Meeting*, Washington, DC, 2006, p. w/o page numbers.

[7] T. Toledo, “Integrated Driving Behavior Modeling,” Dissertation, MIT, 2003.

[8] V. Kanagaraj, G. Asaithambi, C. H. N. Kumar, K. K. Srinivasan, and R. Sivanandan, “Evaluation of Different Vehicle Following Models Under Mixed Traffic Conditions,” *Procedia - Social and Behavioral Sciences*, vol. 104, pp. 390–401, Dec. 2013.

[9] P. Gipps, “A model for the structure of lane-changing decisions,” *Transportation Research Part B: Methodological*, vol. 20, no. 5, pp. 403–414, Oct. 1986.

[10] P. Hidas, “Modelling lane changing and merging in microscopic traffic simulation,” *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 56, pp. 351–371, Oct. 2002.

[11] P. Hidas, “Modelling vehicle interactions in microscopic simulation of merging and weaving,” *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 1, pp. 37–62, Feb. 2005.

[12] M. Ardelt and P. Waldmann, “Hybrides Steuerungs- und Regelungskonzept für das hochautomatisierte Fahren auf Autobahnen (English title: Behavior decision making for autonomous vehicles with gaze control),” *at - Automatisierungstechnik*, vol. 59, no. 12, pp. 738–750, 2011.

[13] 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, Dec. 2012.

[14] M. Ruf, J. R. Ziehn, D. Willersinn, B. Rosenhahn, J. Beyerer, and H. Gotzig, “A Continuous Approach to Autonomous Driving,” in *Proceedings of the Conference on Vehicle and Infrastructure Safety Improvement in Adverse Conditions and Night Driving (VISION 2014)*, Versailles, France, Dec. 2014.

[15] S. Brechtel, T. Gindele, and R. Dillmann, “Probabilistic MDP-behavior planning for cars,” in *14th Internat. IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, oct 2011, pp. 1537–1542.

[16] S. Brechtel, T. Gindele, and R. Dillmann, “Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs,” in *2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC) October 8-11, 2014. Qingdao, China*, Oct. 2014, pp. 392–399.

[17] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, 2nd ed. Cambridge, MA, USA: MIT Press, 2000.

[18] S. Ulbrich and M. Maurer, “Situation Assessment in Lane Change Behavior Planning for Automated Vehicles,” in *Proceedings of the 18th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2015)*, Gran Canaria, Spain, Sept 2015, accepted.

[19] S. Ross, J. Pineau, S. Paquet, and B. Chaib-Draa, “Online Planning Algorithms for POMDPs,” *The journal of artificial intelligence research*, vol. 32, no. 2, pp. 663–704, July 2008.

[20] M. Earl and R. D’Andrea, “Iterative MILP methods for vehicle-control problems,” *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1158–1167, Dec. 2005.

[21] K. F. Culligan, “Online Trajectory Planning for UAVs Using Mixed Integer Linear Programming,” Dissertation, University of London, 2006.

[22] J. Shen and X. Jin, “Detailed traffic animation for urban road networks,” *Graphical Models*, vol. 74, no. 5, pp. 265–282, Sept. 2012.

[23] A. Broggi, P. Cerri, S. Debattisti, M. Laghi, P. Medici, M. Panciroli, and A. Prioletti, “PROUD-Public road urban driverless test: Architecture and results,” in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, June 2014, pp. 648–654.