Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, October 6-9, 2013

Probabilistic Online POMDP Decision Making for Lane Changes in Fully Automated Driving

Simon Ulbrich and Markus Maurer*

***Abstract*— The *Stadtpilot* project aims at fully automated driving on Braunschweig's inner city ring road. The TU Braunschweig's research vehicle "Leonie" is one of the first vehicles having the ability of fully automated driving in real urban traffic scenarios. This paper shows our decision making approach for performing lane changes while driving fully automated in urban environments. We apply an online Partially Observable Markov Decision Process (POMDP) to accommodate inevitable sensor noise to be faced in urban traffic scenarios. In this paper we propose a two step algorithm to keep the complexity of the POMDP low enough for real-time decision making while driving. The presented approach has been integrated in our vehicle and was evaluated in real urban traffic.**

I. INTRODUCTION

In the past two decades the driving abilities of fully automated driving vehicles have progressed rapidly. Especially the DARPA Grand Challenges put fully automated driving into the focus of many research groups around the world. After the participation in the DARPA Urban Challenge in 2007 [1], the Technische Universität Braunschweig continued its effort in the automated driving field with the *Stadtpilot* project. The goal of the *Stadtpilot* project is to drive fully automated on Braunschweig's inner city circular ring road. First accomplishments of driving in heavy inner city traffic have already been successfully demonstrated to the public [2]. In this paper we present our approach for probabilistic decision making for performing lane changes in urban environments.

Inevitable sensor noise is a major challenge for coherent decision making. Tracked sensor object's positions and velocities are to a certain degree noisy due to physical limits of the sensor systems and unavoidable limitations for low latency object tracking algorithms. Furthermore the de facto viewing range of the LIDAR sensor setup being used in our vehicle is limited to maybe 70 meters or less. Decision making for lane changes is particularly difficult in urban environments because there are many choices of actions in inner city traffic - for the automated vehicle itself as well as for other traffic participants. Moreover, drivers change their driving intent quite often. Although they might have left a big gap in front of them for several seconds they could suddenly decide to accelerate because they changed their strategy to achieve their individual driving target.

This paper is structured as follows: First of all we introduce the reader to the problem of tactical lane change decision making in the overall context of decision making and maneuver planning for lane changes in automated driving. Then we review other approaches for tactical lane change decision making. Thereafter we shortly introduce the reader toward Markov decision problems and approximation strategies for real world applications. Section III provides details about the lane change decision making model developed by the authors. Section IV discusses our evaluation metrics and evaluates our approach while driving in urban traffic. In the end, section V concludes this paper and points out further research directions.

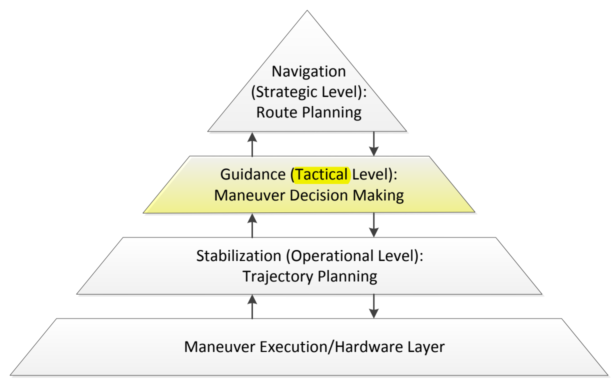

Fig. 1. Tactical decision making as a sub-problem of overall decision making for lane changes in automated driving

II. BACKGROUND

A. *Integration of tactical driving maneuver decision making in the overall process*

Different levels of decision making exist for automated driving. Donges [3] introduced the classification of driving tasks into navigation tasks (strategic level), guidance tasks (tactical level) and stabilization tasks (operational level). Figure 1 illustrates the different decision making levels. All of them are necessary for successful execution of a lane change. In this paper we focus on the tactical level. However, we briefly introduce the reader to the overall problem of decision making for lane changes and provide a short hint how they are approached in the *Stadtpilot* project.

- *Strategic level:* A standard a-star graph search algorithm is employed to find the optimal sequence of roads and even driving lane segments to reach the driving destination. This a-priori route planning also encompasses the planning of *necessary lane changes* to reach the

* All authors are with the Institute of Control Engineering, Technische Universität Braunschweig, 38106 Braunschweig, Germany. `ulbrich, maurer at ifr.ing.tu-bs.de`

978-1-4799-2914-613/$31.00 ©2013 IEEE

driving destination, e.g. for reaching a left turning lane while driving on the rightmost lane of a multiple lane road. The remaining distance along the route to the last possible point of such a *necessary lane change* can be used as an infrastructure related measure for the urgency of a lane change.

- *Tactical level:* The tactical decision making level is responsible for modifying the a-priori-planned lane-level route in such a way that it fits well with the driving maneuvers of other traffic participants. Hence abstract *necessary lane changes* are rendered concrete into collision-free *decided lane changes* at a certain spatial location and point in time. Moreover, slow vehicles might necessitate additional lane changes to incorporate overtaking maneuvers. This tactical decision making used for rendering an abstract pre-calculated route into palpable driving maneuvers while considering the dynamically changing environment is the focus of this paper.
- *Operational level:* Operational tasks subsume everything that is necessary to translate the maneuvers from the tactical decision making into values for the steering, acceleration and braking actuators. Among those operational aspects are control engineering tasks like velocity control and steering control as well as low-level trajectory planning for jerk-free comfortable driving.

B. Tactical decision making for lane changes and cognate problems

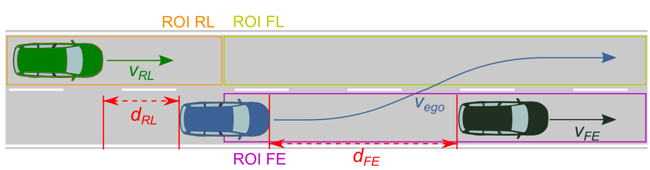

Fig. 2. Typical scenario for lane change decision making with two dynamic objects and three regions of interest rear left (RL), front left (FL) and front ego (FE)

Figure 2 illustrates a typical decision making problem for doing 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 and the behavior of other vehicles around the ego vehicle. Decision making for automatic driving requires rapidity, coherency, providentness and predictability.

- *Rapidity:* Decision making needs to be fast. Despite some strategic decisions (e.g. route re-calculation) might be allowed to take some more time, at least the tactical decisions for driving in urban environments need to be taken fast (<100ms).
- *Coherency:* A decision should fit in the framework of the decisions being taken so far. Similar to a human driver, a decision making unit should not constantly change its mind about the driving maneuvers to be taken. All decisions should align well with a long term goal. However, this does not necessarily imply entirely greedy decision making or not reconsidering a previously taken decision at all.
- *Providentness:* Decision making should have some foresight to predict how the situation will look like after execution of some maneuvers or simply after some time has elapsed.
- *Predictability:* Last of all decision making should be predictable in a sense of acting like a human driver is supposed to act in the given situation as driving on an urban road implies direct or indirect interaction with several other (human) drivers.

The DARPA Grand and Urban Challenges stimulated a lot of research efforts in the automated driving field. Every participating team had some kind of decision making instance. Some of them also implemented complex driving maneuvers requiring tactical decision making tasks. Most teams used some variant of a state machine. For the decision making for lane changes and intersections the CMU's team "Boss" analyzed the gaps between cars by a set of analytic equations and used thresholding and binary decisions [4]. Team Junior used some cost based approaches for global path planning and a finite state machine for switching between different maneuvers [5]. Team Odin from Virginia Tech [6] used a set of behavior-based driver models with some special arbitration method. Team Carolo from TU Braunschweig [1] used a hybrid approach of a traditional rule-based decision making engine and a behavioral model. In our experience, the behavioral model did not provide the required predictability needed for driving on public roads. Nevertheless we built upon the idea of a cost based evaluation of decision alternatives.

Pellkofer [7] and Naranjo et al. [8] used fuzzy logic for modeling lane change decision making problems. The advantage of such a fuzzy logic approach is its simplicity and computational efficiency. Schubert et al. [9] used a Bayesian network for situation assessment and decision making for lane changes. *Deceleration to safety time* (DST) was used as a central criterion for lane change situation assessment. Wei et al. [10] used an analytic dynamic environment prediction model for driving and doing lane changes on freeways. Their model focused on the cooperative behavior with the vehicles around the automated vehicle. They used a set of analytic cost functions for decision making. Their approach did not draw particular attention toward uncertainties in sensor data and their evaluation was limited toward a simulation environment. Wei et al. [11] extended on this by modeling the task of single-lane automated driving under uncertainty using a Markov decision problem approach. Brechtel et al. [12] showed a way of using Markov decision problems for lane change decision making. They based their decision process's state variables directly on measured values like relative distances and velocities toward surrounding vehicles. On the one hand this helped to keep the decision to be based on physical quantities, on the other hand it made the overall decision process more complex and thus hard to be extended toward uncertain measurement data. Bandyopadhyay et al. [13] applied mixed observability Markov decision processes for the recognition and appropriate motion planning while considering human agents' intentions. However, they provided a more general framework rather than focusing on lane change decision situations in particular.

Other related research problems can be found in the field of maneuver prediction. E.g., Friedman et al. [14] trained Bayesian belief networks for the prediction of lane changes. They showed the general feasibility of the idea of learning the structure and parameters of belief networks by testing the belief network's performance on simulated data. One of the points for further evaluation is the applicability of the approach toward real uncertain and inaccurate sensor data. Similarly, Gindele et al. [15] used a dynamic Bayesian network for behavior and trajectory prediction.

C. MDPs and POMDPs

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 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 over the time horizon T:

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

A commonly applied approach to find an optimal policy is using *value iteration* see, e.g., ~~[16, p. 499]~~.

Anyhow, true system states are typically not observable. Partially observable Markov decision processes 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.

In real-time applications, POMDPs are often avoided because of their computational complexity. Significant research efforts have been spent on extending POMDP models and finding approximation methods to solve POMDPs.

D. Solving POMDPs by approximations

For real world decision making problems it is necessary to calculate a best possible strategy iteratively to incorporate new measurements. Typically the latest measurements provide the best hints for the immediate actions that are supposed to be taken. Hence decision making needs to be done in real-time. Publications in the POMDP community typically focus on solving POMDP models *offline*. In this context offline means that the focus is typically not to calculate the best possible action for the *current* belief state but rather for *every* imaginable belief state. Hence they provide a policy - prior to the execution - of the best action u to execute in any possible situation. POMDP problems are PSPACE-complete and thus getting computationally intractable for bigger state spaces [17]. Even for relatively small POMDP problems it takes several minutes to hours to calculate approximate offline solutions. On the contrary, for decision making in urban environments decisions need to get updated, e.g., every 100ms. One way to apply current offline POMDP algorithms to real world robotics problems is used, e.g., by Bai et. al. [18]. They run an offline POMDP algorithm to calculate best policies for a finite set of belief distributions over the state space. For the online task, nearest neighbor search is applied to find the best fitting belief distribution in that finite set to match the current belief distribution in real-time.

In this paper we apply an online decision making algorithm [19], [20], [21] to avoid the complexity of computing a sophisticated, long-term policy by planning online only for the current belief state $bel(x_t)$. This is typically done by a look ahead search in the belief state space to explore only those belief states that are actually reachable from the state right now.

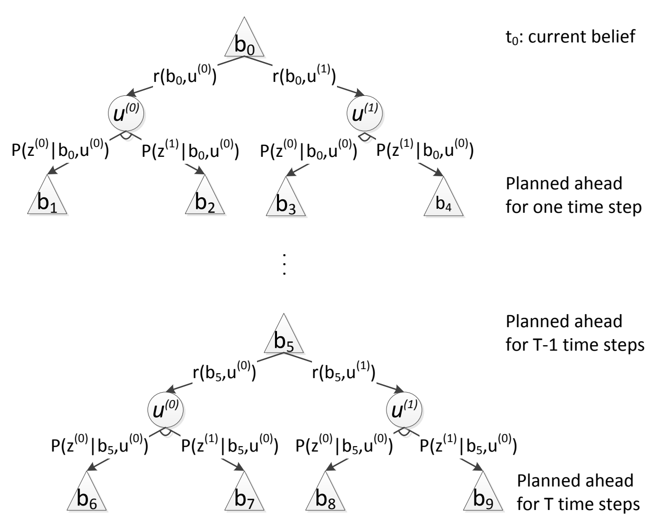

Fig. 3. AND-OR policy tree for a T step plan ahead policy with Branch-and-Bound tree search [22]

We use a variant of Paquet et al. [20] real-time belief space search (RTBSS) approach. Similar as in Ross et al.

[22] we use a blind policy (BP) [22], [23] to obtain a lower bound for $V^*(x)$ and the QMDP algorithm [24] to obtain an upper bound for $V^*(x)$. A blind policy is a policy, where the same action is always executed regardless of the belief state. We modified the blind policy approach slightly by exploiting special knowledge about the problem structure itself. Policies with numerous begins and abortions of lane changes every few time steps are not likely to provide a tight bound. Our blind policy for initiating/aborting a lane change will only initiate/abort one lane change; only the drive ahead-policy is a regular blind policy. After calculating a lower and upper bound for a specific belief state, Branch-and-Bound tree search is executed on the so called policy tree. Similar to Ross et al. [22] and Paquet et al. [20] we expand actions in descending order of their upper bound to maximize pruning.

Figure 3 shows a simple example with two possible actions $|U| = 2$, two measurements $|Z| = 2$, a planning range of T and a discount factor of γ. The policy tree is an AND-OR tree. Belief nodes are represented by a triangular OR-node. At every OR the system has to choose a single action. Every action is represented by a round AND-node. For every action it has to evaluate every subbranch. The edges under an action node denote the probability that this action will result in the child belief state. Vice versa the branches under a belief node, reflect the reward/cost of executing a particular action in that state.

The goal is to find that particular action that will result in the highest expected overall reward for the current belief state b_0. The higher the search depth T is, the smaller the gap between the lower bound and the upper bound will get. The upper and lower bounds for a particular belief state are calculated by a QMDP and BP approximation as introduced above. Those values propagate according to the following rules in the policy tree. $F(Tree)$ denotes the set of fringe nodes in the AND-OR tree. Ross et al. [22] provide a more extensive discussion.

$$V_{lo}(b) = \begin{cases} V_{BP}(b), & \text{if } b \in F(Tree) \\ \max_{u \in U} V_{lo}(b,u), & \text{otherwise} \end{cases} \quad (2)$$

$$V_{lo}(b,u) = r(b,u) + \gamma \sum_{z \in Z} p(z|b,u) \cdot V_{lo}(B(b,u,z)) \quad (3)$$

$$V_{up}(b) = \begin{cases} V_{QMDP}(b), & \text{if } b \in F(Tree) \\ \max_{u \in U} V_{up}(b,u), & \text{otherwise} \end{cases} \quad (4)$$

$$V_{up}(b,u) = r(b,u) + \gamma \sum_{z \in Z} p(z|b,u) \cdot V_{up}(B(b,u,z)) \quad (5)$$

III. MODELING LANE CHANGE DECISION MAKING

This section will describe the approach developed for decision making for lane changes. Based on the situation representation consisting of the dynamic objects in front and behind the ego vehicle on its own lane, its left neighbor lane and its right neighbor lane, our implementation follows a simple two step procedure as in figure 4. The situation is evaluated by a signal processing network. A signal processing network is a graph, in which each node represents a computation block and each edge a signal flow from one block into another. The outputs of the signal processing networks, whether a lane change is possible or not, and whether a lane change is beneficial or not, are sent to the POMDP decision making algorithm. The signal processing networks are used to simplify the POMDP model.

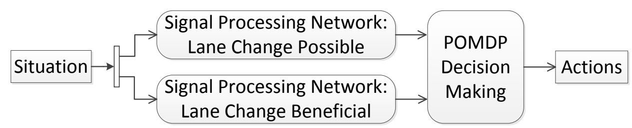

Fig. 4. Proposed steps for lane change decision making

The signal processing networks consider relative distances, relative velocities and time to collisions with objects around the automated vehicle. Figure 5 shows the signal processing network used to find an aggregated measure if a lane change is possible. Our signal processing network is a rather straight-forward series of mathematical operations being applied to aggregate some of the situation's state variables into a single number between zero and one to express if a lane change is possible. We do a separate evaluation for every of the three relevant regions of interest (ROIs) in figure 2. The region of interest behind the ego vehicle on the neighbor lane is the most critical for collision avoidance while doing lane changes. Hence most fine-tuning has been dedicated to it. If there is no object in the ROI it is assumed that a lane change is possible regarding that particular ROI. If there is an object we use the cumulative distribution function of a Gaussian distribution Φ to translate the numeric value of the objects's distance into a random variable between zero and one. Figure 5 shows the parameters μ and σ of the Gaussian's cumulative distribution function. A similar operation is applied to the object's time gap ($tg = dist/v_{ego}$) and its time to collision ($ttc = |dist/(v_{obj} - v_{ego})|$) where $dist$ is the distance between the ego vehicle and the object and v_{ego} the ego vehicle's velocity. Depending on the ego vehicle's velocity as a parameter we use two different sets of Gaussian distributions. In the end we use the minimum of the different random variables to aggregate all of them into one measure for the particular region of interest. The minimum operation was selected due to the similarity of the overall process toward describing the process with fuzzy logic rules. However, other than in pure fuzzy logic we needed additional operations like averaging and weighting by factors to get an acceptable behavior.

All in all, we are convinced that there are other, theoretically more profound ways to compute a random variable describing if a lane change is possible or not. Anyhow, the whole signal processing network can be replaced by any other module, e.g. a Bayesian network or a soundly formulated fuzzy logic while keeping the second POMDP decision making stage unchanged. We used this signal processing network because it performed well in practical driving in real urban scenarios.

A second signal processing network was developed to determine if a lane change is beneficial due to the dynamic aspects in a traffic situation (e.g. slow car ahead). However, due to space constraints and not having direct relevance for the evaluation in section IV we do not discuss this second signal processing network in detail.

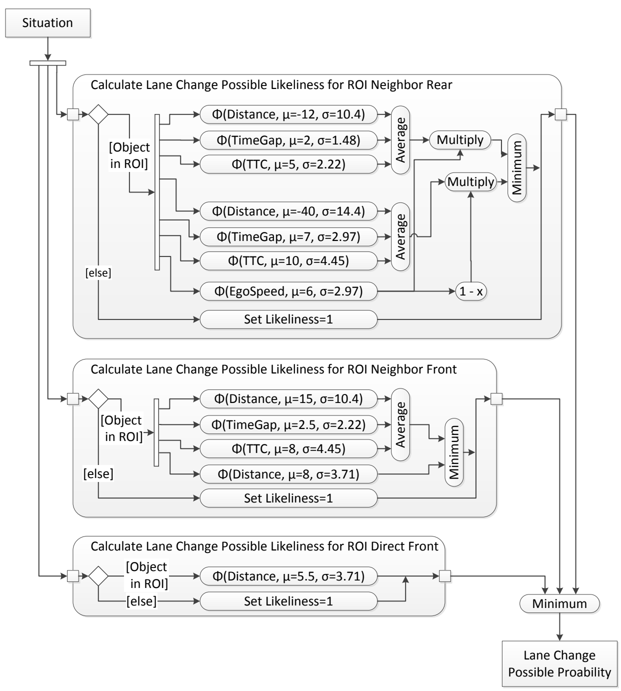

Fig. 5. Example of the signal processing network for lane change possible

For the moment our model does not consider any direct state transitions from a lane change left toward a lane change right state. Hence the POMDP model only need to have eight states:

$$\begin{aligned} X = \{&(\neg LcPos, \neg LcInProg, \neg LcBen), \\ &(LcPos, \neg LcInProg, \neg LcBen), \\ &(\neg LcPos, LcInProg, \neg LcBen), \\ &(LcPos, LcInProg, \neg LcBen), \\ &(\neg LcPos, \neg LcInProg, LcBen), \\ &(LcPos, \neg LcInProg, LcBen), \\ &(\neg LcPos, LcInProg, LcBen), \\ &(LcPos, LcInProg, LcBen)\} \end{aligned} \quad (6)$$

$LcPos$ is a binary state variable describing whether a lane change is possible, $LcBen$ is a binary state variable to state if lane change is beneficial and $LcInProg$ is a binary state variable to state if a lane change is currently in progress.

Moreover three actions have been modeled, regular driving (straight) ahead, initiating a lane change to the neighbor lane ($InitiateLC$) and aborting a lane change being currently in progress ($AbortLC$):

$$U = \{Drive, InitiateLC, AbortLC\} \quad (7)$$

All elements of the reward matrix are set to zero except for:

$$\begin{aligned} &r(u = InitiateLC, x = (:, \neg InProg, :)\ ) = -100 \\ &r(u = InitiateLC, x = (:, InProg, :)\ ) = -10000 \\ \\ &r(u = AbortLC, x = (:, \neg InProg, :)\ ) = -10000 \\ &r(u = AbortLC, x = (:, InProg, :)\ ) = -200 \\ \\ &r(u = Drive, x = (:, \neg InProg, \neg LcBen)\ ) = +5 \\ &r(u = Drive, x = (:, \neg InProg, LcBen)\ ) = -5 \\ &r(u = Drive, x = (LcPos, InProg, \neg LcBen)\ ) = -5 \\ &r(u = Drive, x = (\neg LcPos, \neg InProg, :)\ ) = -60 \\ &r(u = Drive, x = (LcPos, InProg, LcBen)\ ) = +50 \end{aligned} \quad (8)$$

Every ":" denotes $\forall x_{sub} \in X_{sub}$ for the specific subsection of the state space X. The state transition matrix for $p(x'|x, u)$ was initialized such that state transitions roughly fit with observed state transitions in real world driving scenarios. For the status quo the state transition matrices were set by using expert knowledge. The authors do expect much room for improvements when these matrices are learned from real world scenarios, e.g., with reinforcement learning.

IV. EVALUATION

As a first step we used Virtual Test Drive (VTD)[1] for a purely virtual simulation of the lane change decision situation against perfect noise free data. After all algorithms were working correctly with simulated, noise free data, we implemented the algorithms in our research vehicle "Leonie" and re-enacted a lane change situation as in figure 2 on a test facility with two lanes. We used the setup in figure 2 to fine-tune the reward parameters in equation 8. Due to the lack of space we do not provide details about the evaluation results in the simulation or on the test track. However, we focus our discussion for this evaluation section toward the evaluation in urban traffic. After the algorithms have passed our required test procedure for being used on a public road, we started to do first lane changes in urban traffic.

Compared to regular driving on a single lane, lane changes are comparably dangerous maneuvers for being decided only by the automated vehicle itself, especially because they are hard to predict by a safety driver and because of the short amount of time being available for overriding/aborting a lane change once it is initiated. Therefore we currently only announce a planned lane change to the safety driver and do not execute it until the vehicle got a clearance from the safety driver for that lane change. Later on we plan to simplify that procedure to just announcing a lane change and not explicitly

[1] www.vires.com

asking for a confirmation, but just aborting the lane change as soon as an overriding is detected.

Our resources did not allow an extensive field study, whether a large group of human (safety) drivers judge the lane change feeling as natural and comfortable. Hence we reduce our evaluation toward more technical metrics in state estimation rather than judging the overall feeling of appropriate action selection. Therefore we limited our evaluation to automated driving without actually executing lane changes. Hence our evaluation only evaluates the lane changes our algorithms are planning without actually executing them during evaluation. We recorded the raw data from the sensors as well as the decision algorithm's results. Since there is no way of getting hard ground truth data for abstract state variables like $LcPos$ we manually labeled our training data by averaging the results of three human experts judging the lane change situation based on a video in a range from zero to one. For evaluation we used a stretch of inner city driving on a two lane street. We assumed that a lane change was necessary during all the time. Hence the $LcPos$ belief of the POMDP directly translated into lane change announcements/suggestions. We drove a stretch of 2.5 km in urban traffic as in [2]. We excluded situations where an evaluation would not be meaningful, e.g., during the U-turn in figure 6, etc. All in all we used 300 seconds of driving in urban traffic for evaluation.

Fig. 6. Track used for evaluation of algorithm

Choosing the right policies depends on two things: On the one hand the correct rewards/penalties for conducting the specific actions in specific system states and on the other hand the correct beliefs of the system state. For the sake of simplicity, the authors will limit the quantitative performance analysis on the system state estimates and using only subjective driving feeling for parametrization of the action selection.

A common way to evaluate decision making algorithms is using receiver operator curves (ROC). This is a plot of the true positive rate $P(T_p)$ as a function of the false positive rate $P(F_p)$. A true positive is a situation where a lane change is possible, a false positive one where a lane change is assumed to be possible, but in reality it is not. The area under the ROC curve (AUC) can be used as an aggregated measure of the classifier's performance [26]. We sorted the data by the lane change possible state estimate for different situations

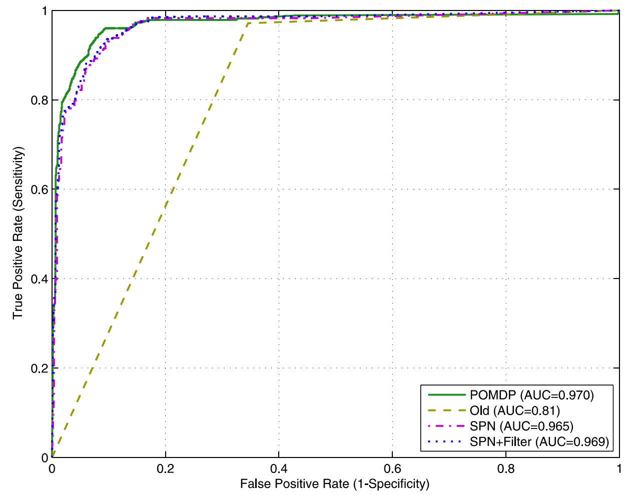

Fig. 7. ROC curves for different decision making approaches

while driving on the route shown in figure 6. In the lower left corner of the ROC curve we find all situations where a lane change is easily possible. The upper right area shows all situations where a lane change is clearly not possible. Hence the effect of this sorting is the same of varying the decision threshold for a binary decision making for the lane change possible state estimate. Figure 7 shows the resulting ROC curve for different decision making approaches. In our performance analysis we compared an existing manually crafted decision making algorithm *(Old)* [25], directly using the estimates from our signal processing networks *(SPN)*, the low pass filtered output of the signal processing network *(SPN+Filter)* and the state estimate of the POMDP. The curves indicate that the POMDP-based as well as the SPN-based decision making algorithms outperform the old handcrafted decision making algorithm. Moreover, they point out that the POMDP-based approach outperforms the bare signal processing network and the filtered signal processing network output at least slightly.

Anyhow, ROC curves assume binary decision making situations. Performing lane changes is typically not a binary decision. There is typically no hard threshold until which a lane change is still perfectly possible and vice versa. It is more a gradual decision. Hence we suggest using Pearson's linear correlation coefficient [27, p. 636 ff.] between the algorithm's state estimate X and the manually labeled ground truth data Y

$$\rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)} \cdot \sqrt{\mathrm{Var}(Y)}} \tag{9}$$

or the normalized mutual information $I_{norm.}(X,Y)$. The normalized mutual information is based on Shannon's definition of the information entropy. It measures the mutual information between two random variables:

$$\begin{aligned} I(X,Y) &= I(Y,X) \\ &= H(X) + H(Y) - H(X,Y) \\ &= \sum_{x \in X} \sum_{y \in Y} p(x,y) * log_2 \left(\frac{p(x,y)}{p(x) * p(y)} \right) \end{aligned} \quad (10)$$

For the ease of use, it should be normalized. We followed the suggestion of Kvalseth [28], Press et al. [27] and Witten/Frank [26, pp. 290 ff.]. They used $0.5 * (H(X) + H(Y))$ as a normalization factor and thus they obtain $I_{norm.}(X,Y) = 2 \cdot \frac{I(X,Y)}{H(X)+H(Y)} = U(X,Y)$ also named symmetric uncertainty by [26, pp 291].

TABLE I
PERFORMANCE EVALUATION AGAINST HUMAN OPERATOR WITH PERSON CORRELATION AND NORMALIZED MUTUAL INFORMATION

Method	Pearson correlation	norm. mutual information
Old	0.620	0.27
SPN	0.876	0.38
Filtered SPN	0.884	0.39
POMDP	0.889	0.48

In table I it is obvious that all decision making methods are better than the existing handcrafted decision making method. Considering the absolute decision performance, the mutual information or the Pearson correlation indicate that the POMDP algorithm is at least on par or even better than the other decision criteria. However, absolute decision performance is only one of the criteria. So far we did not consider the coherency of decisions. Figure 8 shows a representative sample of driving in urban traffic. It shows the number of decision changes of any of the algorithms compared with a human expert. Traffic situations necessitated 16 decision changes. Although the signal processing network provided a good overall decision performance it changed its decisions more than 4 times more often than a human driver would have done. Low-pass filtering and a hysteresis improved the coherency of the lane change decisions. However, the POMDP decision making outperformed by far the other approaches and resulted in much more human like coherency of lane change decision making.

Summarizing the results, they show that the POMDP based decision making approach is superior for our application. Revisiting the criteria in section II-B, decision making should be fast, coherent, provident and predictable. POMDP based decision making fulfills all of them. If approximation methods like QMDP and Branch-and-Bound based online evaluation is used, decision making is fast (<10ms) and hence possible in real time. As shown in figure 8 it exhibits a good coherency; and by its problem definition with considering the discounted future reward it will act with foresight. The high decision correlation toward those of a human driver will result in, for other traffic participants predictable, human-like behavior.

Another nice benefit of the POMDP decision making is that the model can be fully separated from the algorithms.

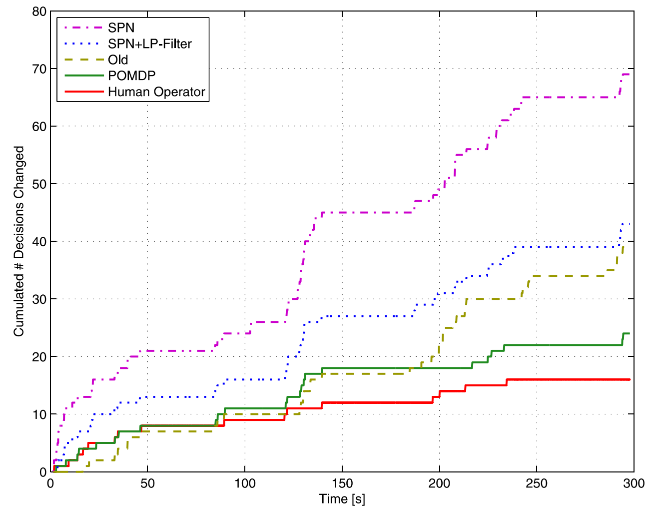

Fig. 8. Number of decision changes

Hence it is easily possible to re-parameterize the decision making to be, e.g., more conservative and comfortable.

V. CONCLUSIONS

In this paper we presented our approach for performing lane changes for automated driving in urban traffic scenarios. Evaluating the algorithm in real world inner city traffic we showed that we gained a significantly higher level of decision coherency while maintaining a high level of decision quality. In this paper we did not focus on driver in the loop scenarios. We asked the driver for a confirmation of the lane changes solely for safety purposes. However, for transferring our approach to the wide field of driver assistance systems, this human-machine-interaction becomes more relevant. Anyhow, our approach scales remarkably well toward these applications by introducing additional states like "prepare lane change" or "announce lane change". Particularly for human-machine-interaction the decision consistency which we gained by this approach is a major aspect to address the warning dilemma for human-machine-interaction. Another point which did not receive particular attention in the paper is, how to handle extraordinary scenarios (snowy, slippery roads or foggy weather). We see two ways to tackle those issues: First, we could model those conditions as a part of the state space and hence do an online estimation as with any other hidden state variable. Though, this comes to the cost of state space complexity. Second, and most likely more viable is a modification of the reward matrix $r(u,x)$ according to those weather conditions. Apart from these application oriented points, we will try to improve the decision making further on the modeling side. In our future work we will try on improving our decision making model in a way that we allow continuous state spaces [18] and make use of simplifying the POMDP problem by exploiting mixed observabilities of the state variables [29].

ACKNOWLEDGMENT

The authors thank the *Stadtpilot* team, namely Dr. Bernd Lichte, Tobias Nothdurft, Falko Saust, Andreas Reschka, Sebastian Ohl, Jaebum Choi and Sven Böhme for their support in improving hard- and software of the research vehicle "Leonie" and spending days and weeks on the test track and public roads to tune the algorithms until they reached the status quo.

REFERENCES

[1] Rauskolb, F. W., Berger, K., Lipski, C., Magnor, M., Cornelsen, K., Effertz, J., Form, T., Graefe, F., Ohl, S., Schumacher, W., Wille, J.-M., Hecker, P., Nothdurft, T., Doering, M., Homeier, K., Morgenroth, J., Wolf, L., Basarke, C., Berger, C., Gälke, T., Klose, F., Rumpe, B., "Caroline: An autonomously driving vehicle for urban environments," in: *Journal of Field Robotics*, vol. 25 (9), pp. 674-724, 2008.

[2] Saust, F., Wille, J.M., Lichte, B., Maurer, M., "Autonomous Vehicle Guidance on Braunschweig's inner ring road within the Stadtpilot Project," in: *Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, pp. 169-174, June 2011.

[3] Donges, E.: "A Conceptual Framework for Active Safety in Road Traffic," in: *Vehicle System Dynamics*, vol. 32, no. 2-3, pp. 113-128, Aug. 1999.

[4] Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M. N., Dolan, J., Duggins, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T. M., Kolski, S., Kelly, A., Likhachev, M., McNaughton, M., Miller, N., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Salesky, B., Seo, Y.-W., Singh, S., Snider, J., Stentz, A., Whittaker, W., Wolkowicki, Z., Ziglar, J., Bae, H., Brown, T., Demitrish, D., Litkouhi, B., Nickolaou, J., Sadekar, V., Zhang, W., Struble, J., Taylor, M., Darms, M., Ferguson, D., "Autonomous driving in urban environments: Boss and the Urban Challenge," in: *Journal of Field Robotics*, 25(8), pp. 425-466, 2008.

[5] Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., Haehnel, D., Hilden, T., Hoffmann, G., Huhnke, B., Johnston, D., Klumpp, S., Langer, D., Levandowski, A., Levinson, J., Marcil, J., Orenstein, D., Paefgen, J., Penny, I., Petrovskaya, A., Pflueger, M., Stanek, G., Stavens, D., Vogt, A. and Thrun, S., "Junior: The Stanford entry in the Urban Challenge," in: *Journal of Field Robotics*, 25(9), pp. 569-597, 2008.

[6] Bacha, A., Bauman, C., Faruque, R., Fleming, M., Terwelp, C., Reinholtz, C., Hong, D., Wicks, A., Alberi, T., Anderson,, D., Cacciola, S., Currier,, P., Dalton, A., Farmer, J., Hurdus, J., Kimmel, S., King, P., Taylor, A., Covern, D. V., Webster, M., "Odin: Team VictorTango's entry in the DARPA Urban Challenge," in: *Journal of Field Robotics*, vol. 25, pp. 467-492, 2008.

[7] Pellkofer, M.: "Verhaltensentscheidung für autonome Fahrzeuge mit Blickrichtungssteuerung," Ph.D. dissertation, Universität der Bundeswehr München, Munich, Germany, 2003.

[8] Naranjo, J. E., Gonzalez, C., Garcia, R., de Pedro, T.: "Lane-Change Fuzzy Control in Autonomous Vehicles for the Overtaking Maneuver," in: *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 438-450, 2008.

[9] Schubert, R., Schulze, K., Wanielik, G.: "Situation Assessment for Automatic Lane-Change Maneuvers," in: *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 607-616, 2010.

[10] Wei, J., Dolan J. M., Litkouhi, B., "A Prediction- and cost function-based algorithm for robust autonomous freeway driving," in: *IEEE Intelligent Vehicles Symposium*, San Diego, USA, pp. 512-517, 2010.

[11] Wei, J., Dolan J. M., Snider, J. M., Litkouhi, B., "A point-based MDP for robust single-lane autonomous driving behavior under uncertainties," in: *IEEE International Conference on Robotics and Automation*, Shanghai, China, pp. 2586-2592, 2011.

[12] Brechtel, S., Gindele, T., Dillmann, R.: "Probabilistic MDP-behavior planning for cars," in: *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, USA, pp. 1537-1542, 2011.

[13] Bandyopadhyay, T., Won, K. S., Frazzoli, E., Hsu, D., Lee, W. S., Rus, D.: "Intention aware motion planning," in: *Springer Tracts in Advanced Robotics, Algorithmic Foundations of Robotics X*, vol. 86, pp. 475-491, 2013.

[14] Friedman, N., Murphy, K., Russell, S., "Learning the structure of dynamic probabilistic networks," in: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, Madison, USA, pp. 139-147, 1998.

[15] Gindele, T., Brechtel, S., Dillmann, R.: "A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments," in: *13th International IEEE Conference on Intelligent Transportation Systems*, Madeira, Portugal, pp. 1625-1631, 2010.

[16] Thrun, S., Burgard, W., Fox, D., *Probabilistic Robotics*, The MIT Press, Cambridge, 2005.

[17] Papadimitriou, C., Tsitsiklis, J. N., "The complexity of Markov decision processes," in: *Mathematics of Operations Research*, vol. 12 (3), pp. 441-450, 1987.

[18] Bai, H., Hsu, D., Kochenderfer, M.J., Lee, W. S.: "Unmanned Aircraft Collision Avoidance using Continuous-State POMDPs," in Durrant-Whyte, H. F., Roy, N. Abbeel, P.: *Robotics: Science and Systems VII*, pp. 1-8, The MIT Press, Cambridge, 2012.

[19] Washington, R., "BI-POMDP: bounded, incremental partially observable Markov model planning," in: *Proceedings of the 4th European Conference on Planning*, Toulouse, France, pp. 440-451, 1997.

[20] Paquet, S., Tobin, L., Chaib-draa, B., "An online POMDP algorithm for complex multiagent environments," in: *Proceedings of The fourth International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-05)*, Utrecht, Netherlands, pp. 970-977, 2005.

[21] Shani, G., Brafman, R., Shimony, S.: "Adaptation for changing stochastic environments through online POMDP policy learning," in: *Proceedings of the Workshop on Reinforcement Learning in Non-Stationary Environments, 16th European Conference on Machine Learning*, Porto, Portugal, pp. 61-70, 2005.

[22] Ross, S., Pineau, J. Paquet, S. Chaib-draa, B., "Online planning algorithms for POMDPs," in: *Journal of Artificial Intelligence Research*, vol. 32, pp. 663-704, 2008.

[23] Hauskrecht, M., "Value-Function Approximations for Partially Observable Markov Decision Processes," in: *Journal of Artificial Intelligence Research*, 13, pp. 33-94, 2000.

[24] Littman, M. L., Cassandra, A. R., Kaelbling, L. P., "Learning policies for partially observable environments: Scaling up," in: *Proceedings of the 12th International Conference on Machine Learning*, Tahoe City, USA, pp. 362-370, 1995.

[25] Scheide, T.: "Aufbau erster Module einer Situationsanalyse des autonomen Straenfahrzeugs Leonie," Diploma thesis, Institut für Regelungstechnik, TU-Brauschweig, Germany, 2009.

[26] Witten, I. H., Frank, E., *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, Amsterdam, p. 291, 2005.

[27] Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., *Numerical Recipes in C - The Art of Scientific Computing*, 2nd edition, Cambridge University Press, Cambridge, 1992.

[28] Kvalseth, T. O., "Entropy and Correlation: Some Comments," in: *IEEE Transactions on Systems, Man and Cybernetics*, vol.17(3), pp.517-519, 1987.

[29] Ong, S., Png, S., Hsu, D., Lee, W., "Planning under Uncertainty for Robotic Tasks with Mixed Observability," in: *International Journal of Robotics Research*, pp. 1053-1068, 2010.