2010 13th International IEEE
Annual Conference on Intelligent Transportation Systems
Madeira Island, Portugal, September 19-22, 2010

A Probabilistic Model for Estimating Driver Behaviors and Vehicle Trajectories in Traffic Environments

Tobias Gindele, Sebastian Brechtel and Rüdiger Dillmann
Institute for Anthropomatics
Karlsruhe Institute of Technology
D-76128 Karlsruhe, Germany
Email: {gindele | brechtel | dillmann}@kit.edu

Abstract— **This paper presents a filter that is able to simultaneously estimate the behaviors of traffic participants and anticipate their future trajectories. This is achieved by recognizing the type of situation derived from the local situational context, which subsumes all information relevant for the drivers decision making. By explicitly taking into account the interactions between vehicles, it achieves a comprehensive situational understanding, inevitable for autonomous vehicles and driver assistance systems. This provides the necessary information for safe behavior decision making or motion planning. The filter is modeled as a Dynamic Bayesian Network. The factored state space, modeling the causal dependencies, allows to describe the models in a compact fashion and reduces the computational complexity of the inference process. The filter is evaluated in the context of a highway scenario, showing a good performance even with very noisy measurements. The presented framework is intended to be used in traffic environments but can be easily transferred to other robotic domains.**

I. Introduction

Autonomous vehicles need a comprehensive representation of their environment in order to make optimal behavior decisions or to conduct behaviors in a safe manner. The process of acquiring the necessary information is hindered by the fact that most of the environmental states are not directly observable and therefore need to be inferred. E.g. there exists no sensor which can perceive the plans of traffic participants. Today's sensory information is often limited to noisy measurements of pose, velocity and some basic geometric features.

While human drivers naturally put themselves into the position of other traffic participants to reason about their behaviors, machine tracking algorithms are usually limited to physical models and simplest heuristics. This is not sufficient for robust long term predictions — and consequently — anticipatory driving. The incorporation of semantics and context information is one key aspect when considering forward-looking driver assistance and autonomous vehicles.

The use of probabilistic methods helps to connect the symbolic and metric behavior representation, at the same time providing a decent way to cope with uncertainty and inaccuracy of a semantic formulation. The combination of both degrees of abstraction allows on the one hand to estimate the state of objects more exactly, because they are enriched with a complex understanding of situations and their context. On the other hand the system does not only obtain a symbolic situation classification, but also predicts it in the future, which is the basis for probabilistic decision making.

II. Related Work

The state estimation problem of traffic participants is often done with a standard filtering approach. Usually the pose of a vehicle is the only state of interest, which allows the use of classical tracking methods like Kalman Filters or Extended Kalman Filters. In the area of multi-target tracking, objects are often assumed to move independently to employ a process model that only models the systems dynamics. While this assumption induces some beneficial mathematical properties — since the objects can be tracked individually — it is definitely not valid for traffic scenarios, where transitions of vehicle states are highly coupled. To solve this problem it is necessary to model these interactions and different behaviors.

Dagli et al [1] [2] realized this problem and proposed a motivation based approach. Other than relying solely on the system dynamics, they conclude that it is necessary to infer the motivations and goals of driver in order to predict his behavior. Another approach is chosen by Zhang et al [3] which estimates the behaviors of vehicles in intersection scenarios. They choose an hierarchical approach that first estimates the likelihoods of possible paths through the intersection on the abstraction level of lanes to identify possible conflicts. In the second stage the behaviors are estimated with a Dynamic Bayesian Network (DBN). Forbes et al [4] already suggested the use of a DBN to estimate the states of traffic participants but their focus lied more on the following decision process of the autonomous vehicle, rather than the state estimation problem. The subproblem of situation recognition is addressed by Meyer et al [5] who use a relational hidden Markov model to recognize different classes of traffic situations based on a semantic representation. While this approach has the ability to predict situations on a high abstraction level, it can not estimate the quantiative properties like the pose of a vehicle. The problem of behavior estimation is often addressed by using Hidden Markov Models. A special variant are interacting multiple models filters (IMM) [6], which explain the observed behavior of an object as an interaction of several models corresponding to different behavior modes. This shows some similarities to our approach since we also use multiple models to describe different behaviors.

978-1-4244-7659-6/10/$26.00 ©2010 IEEE

III. Bayesian Model

A. Overview

Our approach to the estimation problem of the desired properties is to formulate it as a time discrete filtering problem and to apply an appropriate probabilistic inference method to reason about the hidden states.

The main idea is a chain of reasoning steps that result in the prediction of the next state of the global situation. Beginning with the extraction of implicit context information from the vehicle states – like which vehicle drives in front of another – one is able to classify the type of situation each vehicle resides in. Each situation class implies certain behaviors that are likely to be conducted by a driver in this situation. Based on the behavior and context information the metric realization in form of a desired trajectory can be inferred. This ultimately closes the loop by enabling the prediction of the next vehicle states.

As can be seen, we use different levels of abstraction to achieve the goal, thereby combining symbolic and subsymbolic representations. The following sections describe the necessary models and representations in detail.

B. vehicle model

For representing a vehicle state we apply a kinematic one-track model which virtually replaces the front and the rear wheels by one in the middle. Neglecting any slipping effects, the car rotates around its instantaneous center of rotation (ICR). Therefore the car's state depends directly on the orientation of the wheels and the length of the vehicle as depicted in Figure 1.

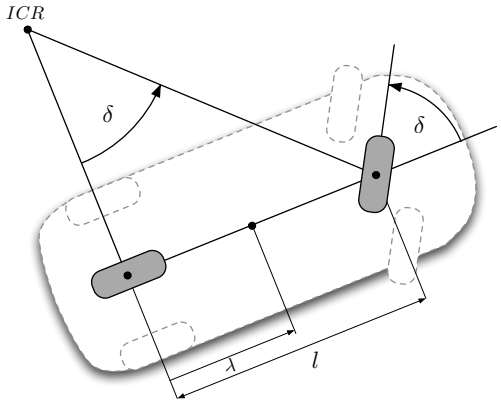

Fig. 1. One-track model of a non-holonomic car

The set of vehicle states $\mathbb{X} \subseteq \mathbb{R}^5$ comprises the position $\left(x_1, x_2\right)^\intercal$ of the vehicle together with the gear angle ψ, the longitudinal velocity v in direction of the gear angle and the steering angle δ. A vehicle state is therefore defined as $\boldsymbol{x} = \left(x_1, x_2, \psi, v, \delta\right)^\intercal \in \mathbb{X}$. The length of the vehicle is given by l and assumed constant.

The basic controls that a driver can directly influence are the acceleration a and steering rate ω of the vehicle. So a control $\boldsymbol{u}$ is defined as: $\boldsymbol{u} = \left(a, \omega\right)^\intercal \in \mathbb{U}$. The total system equations are provided by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\psi} \\ \dot{v} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} v\cos\psi \\ v\sin\psi \\ v\frac{\tan\delta}{l} \\ a \\ \omega \end{bmatrix} \tag{1}$$

C. Trajectories

A trajectory is a time ordered set of states of a dynamical system. In our case a trajectory defines a vehicle state over a time interval. Given a function of the vehicle position

$$y(t) : [t_s, t_e] \rightarrow \mathbb{R}^2$$

together with its first and second derivative and the systems equations (1) all dimensions of the vehicle state $\boldsymbol{x} \in \mathbb{X}$ are defined.

The trajectories we need to represent are realizations of actual behaviors and therefore have to fulfill the non-holonomic constraints. To get a compact representation we further state that all trajectories are entirely defined by their start and end states and the corresponding time interval.

$$\boldsymbol{y} = \left(\boldsymbol{x}_s, \boldsymbol{x}_e, t_s, t_e\right)^\intercal \in \left\{\mathbb{X} \times \mathbb{R}\right\}^2 = \mathbb{T}$$

This constrains the complexity of behaviors, since the space of possible trajectories is restricted.

The intermediate values are obtained by interpolation. We chose a Bézier interpolation scheme. Higher order Bézier curves have proven to be useful in the context of trajectory representation [7]. Besides their compact representation they produce jerk continuous trajectories which obey the non-holonomic constraints of the vehicle. Other interpolation mechanisms like spline interpolation that generate sufficiently smooth curves could be used as well.

A Bézier curve of degree n with control points $\{P_0, \ldots, P_n\}$ is given by

$$P_{[t_s,t_e]}(t) = \sum_{i=0}^{n} B_i^n(t) P_i \tag{2}$$

where the Bernstein polynomials B_i^n are defined as

$$B_i^n(t) = \binom{n}{i} \left(\frac{t - t_s}{t_e - t_s}\right)^i \left(\frac{t_e - t}{t_e - t_s}\right)^{n-i}.$$

The derivative itself is again a Bézier curve with modified control points D_i of degree $n-1$:

$$\dot{P}_{[t_s,t_e]}(t) = \sum_{i=0}^{n-1} B_i^{n-1}(t) D_i \tag{3}$$

with

$$D_i = \frac{n}{t_e - t_s}(P_{i+1} - P_i) \tag{4}$$

To fully define the cubic Bézier trajectory function $y_B(t)$ given $\boldsymbol{y}$ the two inner control points need to be defined. They can be derived from the velocity and gear angle constraints

of the start and end state together with the first derivative of the Bézier curve ($n = 3$):

$$P_1 = P_0 + \begin{pmatrix} v_s \cos\psi_s \\ v_s \sin\psi_s \end{pmatrix} \frac{t_e - t_s}{3}$$

$$P_2 = P_3 - \begin{pmatrix} v_e \cos\psi_e \\ v_e \sin\psi_e \end{pmatrix} \frac{t_e - t_s}{3}$$

D. Behaviors

Behaviors are symbolic representations of context dependent motion primitives that a vehicle is able to conduct. We assume that a finite set of behaviors is sufficient to describe all possible maneuvers of a real driver. We chose the classes based on semantic equivalents that humans use to describe the maneuvers of a vehicle. This is not an optimal choice but has the advantage that the estimations can be interpreted by humans.

For the evaluation of the approach we modeled several basic behaviors for highway driving situations. Naturally the set of behaviors can be extended to other domains like urban traffic where e.g. behaviors for intersection handling are needed. It is important that the situation description comprises all features necessary to discriminate behavior classes.

The set of behaviors $\mathbb{B}$ consists of {*free_ride*, *following*, *acceleration_phase*, *sheer_out*, *overtake*, *sheer_in*} which are sufficient to describe the most common maneuvers in highway scenarios. Examples of the different behavior patterns together with corresponding trajectories are shown in Figure 2.

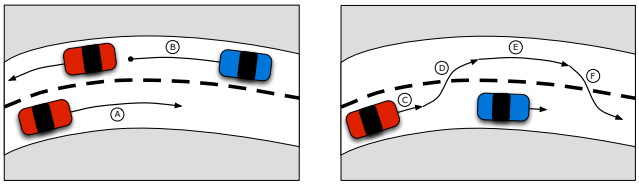

Fig. 2. Behaviors with trajectories. A *free_ride*, B *following*, C *acceleration_phase*, D *sheer_out*, E *overtake*, F *sheer_in*

a) free_ride: This behavior is applied if there are no other vehicles or obstacles ahead. The main aspects a driver has to consider are speed limits, sight conditions and the curvature of the road.

b) following: A vehicle is in following mode if another vehicle drives ahead on the same lane. Unlike the *free_ride* behavior the driver has to accommodate to the velocity of the up front vehicle to keep the necessary safety distance.

c) acceleration_phase: The complete overhauling process is separated in four phases. The first phase is the acceleration phase where the driver accelerates to become fast enough to drive past the up front vehicle.

d) sheer_out: In the sheer out phase the car is often still accelerating. The car moves smoothly in a S-shaped trajectory to the opposite lane. When it reaches the center of the opposite lane the *overtake* phase begins.

e) overtake: In the *overtake* phase the overhauling car drives on the opposite lane until it is far enough ahead of the other vehicle to sheer back in.

f) sheer_in: By sheering in the car changes back from the opposite lane to the original lane. As soon as the car reaches the original lane it changes back to *free_ride* or *following*.

All the different behaviors induce constraints on the corresponding trajectories. A requirement for a trajectory of the *sheer_out* behavior is e.g. that the end point lies on the left opposite lane and that the velocity is significantly faster than that of the overhauled vehicle. These constraints are expressed by a conditional probability distribution function (cdf) over the trajectory space depending on the behavior and the situational context.

E. Situation context

The situational context describes the properties of the vehicle surroundings. It consists of relations between objects, which are in our case mainly distances. The relations are extracted by analyzing the implicit features of the situation, thereby making them explicit. These relations are the direct basis for the drivers decisions. E.g. if the distance between two car falls below the safety distance, the the driver of the following car will decelerate.

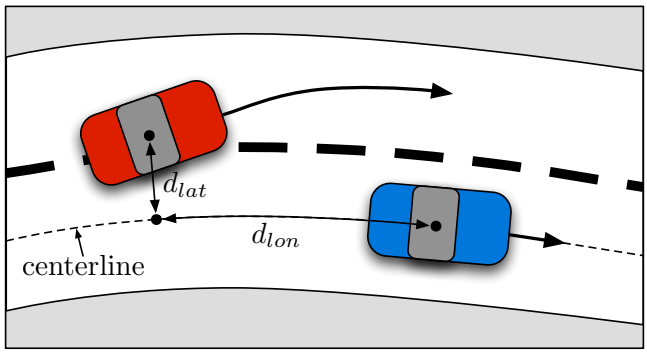

Fig. 3. Distance relations between two cars. The dashed line represents the baseline for the distance calculations

Since the situational context is only needed to infer the behaviors, only those relations need to be considered that are relevant for the drivers decision making. They are evaluated for every car in the scene. Figure 3 illustrates the distance relations between two cars. For the modeled set of behaviors $\mathbb{B}$ the following distance relations are relevant:

longitudinal distance: The longitudinal distance $d_{lon,f}$ describes the distance of a vehicle to the next vehicle ahead. The distance is measured between the projected positions of the vehicles on the centerline of the lane. If there is no vehicle ahead in the observable area then the distance is set to infinity. The analog distance relation $d_{lon,b}$ measures the distance to the next vehicle behind. Since the centerline follows the curvature of the road it is a better measure than the Euclidean distance.

lateral distance: The lateral distance d_{lat} measures the displacement between the position of a car and the centerline. Again it's calculated by projection of the position onto the centerline.

relative velocity: Regarding a vehicle and the next vehicle ahead, the relative velocity $d_{dv,f}$ describes the difference between their velocities. Analogous $d_{dv,b}$ describes the relative velocity to a vehicle behind.

The basic set of context relations is defined as $\mathbb{C} = \{d_{lat}, d_{lon,f}, d_{lon,b}, d_{dv,f}, d_{dv,b}\}$. Since for some driving maneuvers, like lane changing or overtaking, the objects on adjacent lanes are also relevant, similar relations are included in the overall set of context features. If the set of behaviors is extended for other applications, $\mathbb{C}$ needs to be extended to contain all relevant context features.

F. Statespace

The state space of the filter consists of several random variables that describe the different aspects of the situation. We assume a finite number of n vehicles in the scene. The random variables are defined as follows:

X: Vector containing the vehicle states.

$$X = \begin{pmatrix} X_1 & \dots & X_n \end{pmatrix}^\intercal, \quad X_i \in \mathbb{X}$$

C: Vector containing the features of local situational context, primarily consisting of relevant distances between the traffic participants and other objects, e.g. the distance to next vehicle ahead.

$$C = \begin{pmatrix} C_1 & \dots & C_n \end{pmatrix}^\intercal, \quad C_i \in \mathbb{C}$$

S: Vector containing the recognized situations. $\mathbb{S}$ is the set of all situations

$$S = \begin{pmatrix} S_1 & \dots & S_n \end{pmatrix}^\intercal, \quad S_i \in \mathbb{S}$$

B: Vector of behaviors. $\mathbb{B}$ is the set of all behaviors

$$B = \begin{pmatrix} B_1 & \dots & B_n \end{pmatrix}^\intercal, \quad B_i \in \mathbb{B}$$

T: Vector of the trajectories, representing the realization of the behaviors.

$$T = \begin{pmatrix} T_1 & \dots & T_n \end{pmatrix}^\intercal, \quad T_i \in \mathbb{T}$$

Z: Measurement vector providing information about the pose. The actual form depends on the sensor configuration. We assume the sensor signals can be linearly mapped onto a subspace of the vehicle pose to ease the formulation of the measurement model. A reasonable assumption is the observability of the position (x_1, x_2), orientation ψ and speed v of the vehicles.

$$Z = \begin{pmatrix} Z_1 & \dots & Z_n \end{pmatrix}^\intercal, \quad Z_i = \begin{pmatrix} x & y & \psi & v \end{pmatrix}$$

All variables marked with a minus like X^- correspond to the random variables at time index $t-1$. Since we assume that the first order Markov assumption is met, we only need to consider random variables with time index t and $t-1$.

G. Decomposition of the Joint Distribution

The filter is modeled as a first order Markov process. The process satisfies the Markov property since the state space subsumes all relevant information needed to infer the next system state. This property can be achieved for all higher order Markov processes [8]. To exploit the conditional independencies between random variables it is beneficial to model the process as a Dynamic Bayesian Network (DBN) [8].

Assuming several conditional independencies the joint distribution can decomposed as follows:

$$\begin{aligned} P(X, X^-, C, S, B, B^-, T, T^-, Z) = \\ P(X^-)P(B^-)P(T^-)P(X|T^-)P(C|X) \\ P(S|C)P(B|B^-, S)P(T|T^-, B, C, X)P(Z|X) \end{aligned} \quad (5)$$

The cdfs are explained in section III-H.

Figure 4 shows the resulting DBN. Continuous nodes are depicted by circles and discrete nodes by rectangles. Solid lines represent direct dependencies within a time frame while dashed lines state dependencies between time steps. The behaviors e.g. depend on the current situations and the last conducted behaviors.

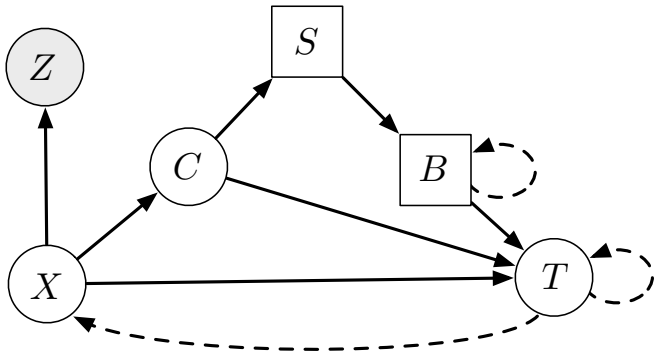

Fig. 4. Bayesian Network. Dashed Lines represent temporal dependencies while solid lines represent causal dependencies within a time frame

H. Filtering Models

The factors of the decomposition of the joint distribution (5) are referred to as the models of the filter. They describe the dependencies between the random variables in form of cdfs. Detailed descriptions and interpretations of the models, as well as the definitions of the corresponding cdfs, are given in the following sections.

1) Dynamics Model - $P(X|T^-)$: The dynamics model describes the change of the vehicles states after a time step Δt, given their last planned trajectory. This model is often referred to as the motion model. Since we neglect the explicit handling of collisions between vehicles in the dynamics model, we are able to regard the individual motions as conditionally independent. This simplifies the cdf to

$$P(X|T^-) = \prod_{i=1}^{n} P(X_i|T_i^-)$$

Since the planned trajectory already accounts for the non-holonomic motion constraints, the new vehicle state can be derived from the planned trajectory and current time t using (2). Allowing some small deviations from the intended motion, represented by the error term $\boldsymbol{e_x}$, the cdf can be stated as follows:

$$P(X_i|T_i^- = \boldsymbol{y}) = g_t(\boldsymbol{y}) + \boldsymbol{e}_x$$

where $g_t(\boldsymbol{y}) : \mathbb{T} \rightarrow \mathbb{X}$ calculates the new vehicle state given the curve parameters $\boldsymbol{y}$. We assume $\boldsymbol{e}_x$ to be normal distributed with $\boldsymbol{e}_x \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_x)$

2) Context Model - $P(C|X)$: The context model describes how the local situational context is derived from the vehicle states. The context describes all relevant information needed by a driver to choose an appropriate behavior. This result in a local view for each vehicle, allowing us to decompose the model as follows:

$$P(C|X) = \prod_{i=1}^{n} P(C_i|X)$$

The distance relations are evaluated with the helper function $d_{map}(\boldsymbol{x})$ as described in III-E. Therefore a virtual map of the road network is employed, which allows to match the vehicles to corresponding lanes, as well as finding the next vehicles ahead or behind. It also provides the centerlines needed for the projection. The cdf for a specific $\boldsymbol{x}$ is defined as a Dirac distribution

$$P(C|X = \boldsymbol{x}) = \delta_{d_{map}(\boldsymbol{x})}.$$

Since the distances can be calculated without uncertainty, given the exact states of vehicles and assuming a perfect map, no error term is needed.

This model couples all vehicle states together. By extracting all relevant information from the global object state and mapping it to the local context states, one is able to formulate the remaining models for every object individually. This simplifies the formulation of the other models drastically and enables fast convergence of machine learning methods and better generalization, since the models need not to be learned for different numbers of objects.

3) Situation Model - $P(S|C)$: If the local context of a vehicle is known, it is possible to distinct several classes of situations. The matching of contexts to the finite set of possible situations $\mathbb{S}$ is done by the situation model. This matching is in fact a discretization that transfers the knowledge about the situation to a symbolic level. All contexts that are grouped in a situation class should share the characteristic, that they imply similar distributions over behaviors. This maximizes the information a situation holds about the drivers behavior decision.

Since the type of situation is recognized individually for each vehicle on the basis of its local context, we can decompose the situation model to:

$$P(S|C) = \prod_{i=1}^{n} P(S_i|C_i)$$

To define the likelihood function for a situation class, we use a weighted multivariate normal pdf, so all likelihood functions together can be interpreted as a finite normal mixture. Figure 5 illustrates the relationship between a context and situation likelihoods. In this example the context only consists of the distance to a vehicle driving ahead and two situation classes are distinguished, namely that the vehicle is close and that the vehicle is far away.

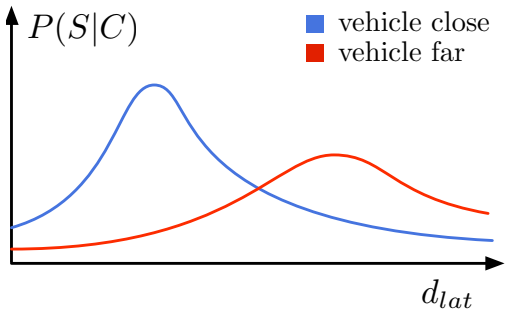

Fig. 5. Exemplary likelihood functions of a simplified situation model with two classes.

4) Policy Model - $P(B|S, B^-)$: On the basis of the last conducted behaviors and considering the type of situation that a vehicle currently resides, the policy model describes the behavior decision of a driver in a probabilistic way. For each situation class it defines a distribution over the behaviors likely to be chosen, which can be seen as set of probabilistic rules covering every combination of behavior and situation.

To simplify the model, we decompose the cdf:

$$P(B|S, B^-) = \prod_{i=1}^{n} P(B_i|S_i, B_i^-)$$

Regarding the behavior decisions as independent is in general a valid assumption, because we can presume the drivers do not communicate to arrange their behaviors. But expecting that the current behavior decision is not influenced by the last behaviors of the other traffic participants is a strong assumption which is not always true. This is only done for complexity reasons and analyzing the impact of this decision will be part of future research.

Currently the set of rules is defined by a human expert. An even better way to obtain the model – in contrast to engineering – is to use machine learning methods or to employ a planner. This would ensure that the model fits the real policies of drivers.

5) Behavior Realization Model - $P(T|B, X, C, T^-)$: The behavior realization model makes the actual step from abstract symbolic behaviors to concrete trajectories, thereby taking into account the overall situation. The cdf of trajectories is also biased by the planned trajectories of the last time step.

In order to reduce the model complexity the cdf is decomposed like the policy model, stating

$$P(T|B, X, C, T^-) = \prod_{i=1}^{n} P(T_i|B_i, X_i, C_i, T_i^-).$$

Neglecting the planned trajectories of the other traffic participants is a critical choice, since the model isn't able to enforce non-overlapping trajectories directly. This is accomplished only indirectly by means of the policy model which favors behavior decisions that do not result in overlapping trajectories.

The most likely trajectories are the ones that optimize several criteria set by the driver. The most important are the length of a path to reach a goal, how economic and comfortable the trajectory is and of course the safety properties. These facts need to be considered in the cdf, e.g if a vehicle drives at high speeds the trajectory on an overtaking maneuver has a flatter shape than one driven at low speeds.

Since we use a Bézier curve representation for the trajectories based on control points as described in III-C, the cdf defines a conditional distribution over the control point parameter space. To define the cdf we use deterministic functions k_b for each behavior type b together with additive white noise $\boldsymbol{e}_b \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_b)$. This results in

$$P(T|B=b, X=\boldsymbol{x}, C=\boldsymbol{c}, T^-=\boldsymbol{y}) = k_b(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{y}) + \boldsymbol{e}_b.$$

The function k_b calculates the control points parameters for a specific behavior b. Like the context model this function also makes use of the virtual road map to generate control points according to the centerlines of the lanes. The main difficulty is to find the appropriate end control point, since the values of the start control point are constrained by the vehicle state in order to produce smooth change-overs. E.g. for the *sheer_in* behavior, the function picks a position and orientation of the end control point based on the centerline of the opposite lane together with an appropriate desired velocity.

These relationships can become rather complex, so for more complex domains, like inner city environments, it is a good idea to utilize a motion planner that generates distributions over likely trajectories. Using machine learning methods is another way to generate the model

6) Observation Model - $P(Z|X)$: The observation model describes the coherence between the overall system state and the likelihood of an observation. We assume that we receive independent observations z_i for each vehicle individually, i.e. we don't have to cope with the data association problem [9] and are able to state the cdf as

$$P(Z|X) = \prod_{i=1}^{n} P(Z_i|X_i).$$

We assume a linear mapping H between vehicle state and expected observation to stay sensor agnostic. The sensor noise $\boldsymbol{e}_z$ is assumed to be normally distributed with $\boldsymbol{e}_z \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_z)$. The cdf of the vehicle observation model can then be stated as

$$P(Z|X_i = \boldsymbol{x}_i) = H\boldsymbol{x}_i + \boldsymbol{e}_z.$$

IV. Filter Equations

Given the joint distribution of all variables (5), the recursive update formulas can be derived using Bayes' rule. The distribution of the posterior state is estimated by marginalizing over the prior state and weighting with the likelihood of the measurement z. Thereby the update can be separated in two steps – classically called *Prediction* and *Correction*. The resulting update equation is

$$P(X, C, B, T, U|z) \propto P(z|X) \int_{B^-,T^-} P(X, C, S, B, T|B^-, T^-) P(B^-, T^-) \quad (6)$$

where the integral corresponds to the prediction and the other part to the correction.

The integral of the prediction step is not solvable in its general form. The existence of an analytical solution depends strongly on the distribution types and the cdfs. Classical representatives of this kind are e.g. Kalman filters and histogram filters [10].

Since we are dealing with a mixed state space covering discrete behaviors and continuous vehicle states in addition to non-linear cdfs, we use a particle filter framework for inference. Particle filters approximate the true posterior and have been used successfully in many robotics applications [10]. The inference is achieved with likelihood weighting [8] where samples are drawn for all unobserved random variables according to their parent states. The overall particles are weighted with the evidences according to the corresponding cdfs. Since the number of samples is finite, the integral of the prediction is replaced with a finite sum over all prior particles.

V. Experiments

To evaluate the presented filter we implemented a slightly simplified version of the model, using only a subset of the named behaviors. The behaviors used are *free_ride*, *following*, *sheer_out* and *overtake*. These behaviors are sufficient to show the principal properties of the filter. The traffic scenario we used for the evaluation was a two lane road scenario with two vehicles, as depicted in Figure 6. In the simulated runs, the vehicle coming from behind is approaching the slower vehicle on the right and changes the lane, so it can pass the other vehicle. For the evaluation we

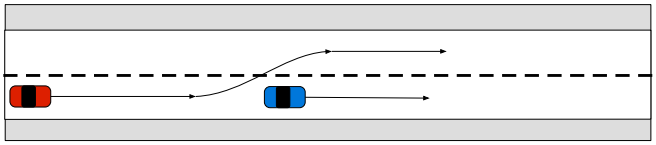

Fig. 6. Scenario of the experiment. Two lane with two cars, where on is overtaking the other

assumed the measurability of the position, gear angle and velocity of the vehicles. The measurements come at a frequency of 10 Hz. To have a valid ground truth, we simulated the scenario several times and added white Gaussian noise to the measurements. The accuracy of the filter is measured by estimating the average root mean square error (RMSE) of the position of the overtaking vehicle. As position estimation

we used the expectation of the particle distribution. The expectation shows a slightly more accurate and more stable performance than the use of the best particle. The particle inference employed 500 particles.

A. Filtering Results

The results of the average estimation error of the vehicles position for different settings of the noise parameters is shown in Tab. I. The filter yields a small average estimation

TABLE I

AVG. RMS ERROR OF THE MEAN POSITION FOR DIFFERENT NOISE PARAMETERS

σ_{x_1,x_2}	σ_ψ	σ_v	avg. RMS position error
0.050	0.025	0.250	0.06
0.100	0.050	0.500	0.09
0.150	0.075	0.750	0.13
0.200	0.100	1.000	0.18
0.250	0.125	1.250	0.21

error of 6 cm of the vehicles position for low measurement noise. As can be seen, the estimation error increases slower than the applied noise, showing the smoothing effect of the process model. This means that the filter is able to estimate the vehicle states even under heavy noise due to the accuracy of his models. Tab. II shows the average likelihood of the true

TABLE II

AVG. LIKELIHOOD OF THE TRUE CONDUCTED BEHAVIOR FOR DIFFERENT NOISE PARAMETERS

σ_{x_1,x_2}	σ_ψ	σ_v	avg. behavior likelihood
0.050	0.025	0.250	0.9168
0.100	0.050	0.500	0.9226
0.150	0.075	0.750	0.9059
0.200	0.100	1.000	0.8934
0.250	0.125	1.250	0.8748

conducted behavior, measuring the quality of the behavior estimation. As expected the reliability of the estimation decreases slowly with increasing noise. In all considered sets of noise parameters a maximum likelihood estimator would classify the behavior correctly in almost every case.

VI. CONCLUSION

In this paper we presented a filter that allows to estimate simultaneously the current pose and behavior as well as the anticipated trajectory of traffic participants. These estimations are needed by an autonomous vehicle for decision making and motion planning. By first recognizing the type of situation based on the local situational context, the filter is able to infer the likely behaviors.

The filter is modeled as a Dynamic Bayesian Network, thereby exploiting the conditional independencies, which allows to describe the models in a compact way and to calculate the inference more efficiently. The probabilistic approach makes the estimation robust under noisy measurements. A distinctive feature is that the expected trajectories are estimated via filtering without enrolling the DBN into the future. This saves a lot of computational effort since it scales linearly with the number of enrolled time slices.

The evaluation showed a good performance of the filter in the domain of highway scenarios. For an application of this approach in an urban environment the models have to be more complex to deal with the variety of possible situations. An interesting direction to explore, is the estimation of plans of traffic participants. We expect that this will improve the accuracy of the estimation of the other states. On the computational side, the use of Rao-Blackwellized methods [11] could reduce the complexity of the inference.

ACKNOWLEDGMENT

The authors gratefully acknowledge the contribution of the German collaborative research center "SFB/TR 28 – Cognitive Cars" granted by Deutsche Forschungsgemeinschaft.

REFERENCES

[1] I. Dagli, M. Brost, and G. Breuel, "Action recognition and prediction for driver assistance systems using dynamic belief networks," *Lecture Notes in Computer Science*, pp. 179–194, 2003.

[2] I. Dagli and D. Reichardt, "Motivation-based approach to behavior prediction," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2002.

[3] J. Zhang and B. Rössler, "Situation Analysis and Adaptive Risk Assessment for Intersection Safety Systems in Advanced Assisted Driving," in *Proceedings of the 21th Fachgespräch Autonome Mobile Systeme*, 2009, pp. 249–258.

[4] J. Forbes, T. Huang, K. Kanazawa, and S. Russell, "The batmobile: Towards a bayesian automated taxi," in *International Joint Conference on Artificial Intelligence*, vol. 14, 1995, pp. 1878–1885.

[5] D. Meyer-Delius, C. Plagemann, G. Von Wichert, W. Feiten, G. Lawitzky, and W. Burgard, "A Probabilistic Relational Model for Characterizing Situations in Dynamic Multi-Agent Systems," in *Proceedings of the 31st Annual Conference for Data Analysis, Machine Learning and Applications*, 2007.

[6] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: a survey," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 103–123, 1998.

[7] J. Choi and G. Elkaim, "Bézier Curve for Trajectory Guidance," in *World Congress on Engineering and Computer Science*, San Francisco, CA, Oct. 22-24 2008.

[8] K. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning," Ph.D. dissertation, University of California, 2002.

[9] I. Cox, "A review of statistical data association techniques for motion correspondence," *International Journal of Computer Vision*, vol. 10, no. 1, pp. 53–66, 1993.

[10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. MIT press, Cambridge, Massachusetts, USA, 2005.

[11] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp. 176–183, 2000.