

Probabilistic Long-Term Prediction for Autonomous Vehicles

Stefan Hoermann¹, Daniel Stumper¹, and Klaus Dietmayer¹

Abstract—Long-term prediction of traffic participants is crucial to enable autonomous driving on public roads. The quality of the prediction directly affects the frequency of trajectory planning. With a poor estimation of the future development, more computational effort has to be put in re-planning, and a safe vehicle state at the end of the planning horizon is not guaranteed. A holistic probabilistic prediction, considering inputs, results and parameters as random variables, highly reduces the problem. A time frame of several seconds requires a probabilistic description of the scene evolution, where uncertainty or accuracy is represented by the trajectory distribution. Following this strategy, a novel evaluation method is needed, coping with the fact, that the future evolution of a scene is also uncertain. We present a method to evaluate the probabilistic prediction of real traffic scenes with varying start conditions. The proposed prediction is based on a particle filter, estimating behavior describing parameters of a microscopic traffic model. Experiments on real traffic data with random leading vehicles show the applicability in terms of convergence, enabling long-term prediction using forward propagation.

I. INTRODUCTION

Towards autonomous and highly automated driving on public streets safe vehicle control requires an anticipatory consideration of the dynamic environment. Also, in the highly dynamic scene, a trajectory planning horizon of about 3 s can result in a rather reactive behavior [1]. Thus, to ensure a safe vehicle state at the end of the planned trajectory with higher speed a planning horizon of about 10 s was used in [2]. In order to keep the plan executable in the future, an estimate of the upcoming situation is important. However, a poor prediction forces re-planning when the assumed scene evolution differs from reality which leads to rather high planning cycles of about 10 Hz. On the other hand for such high prediction horizons a probabilistic treatment is unavoidable and should include uncertainty as well as reliability. The research of the EU project RobustSENSE includes a framework with holistic probabilistic processing [3]. Although a situation expected in 10 s might not be clearly known, a planning algorithm could generate a trajectory driving comfortably through a very likely future but can also overcome an unfavorable evolution that might be unlikely but possible. Also, the ego vehicle might influence the scene evolution emphasizing the need of a collaborative realization of prediction and planning.

Planning strategies considering uncertainty were already proposed [4]. While classical prediction in object tracking usually makes use of simple motion models, e.g. constant acceleration (CA) [5], their forward propagation quickly

results in high uncertainty. Long-term prediction, i.e. longer than 1 s, requires more complex models, designed explicit or learned. Machine learning approaches are popular for prediction tasks. The use case to predict maneuvers in terms of action classes is tackled in literature with great success. Examples for predicted action classes are *stopping or merging* at intersections [6], [7], as well as *breaking or accelerating* in car-following traffic using [8], [9]. However, for the further processing in trajectory planning not only the predicted maneuver, but also its trajectory including uncertainty is required. The authors of [10] proposed an approach to gain a prediction in form of a spatio-temporal probability density function (PDF) of future vehicle trajectories. A Gaussian mixture model is learned and used to infer future trajectory patterns from feature observations of the past 25 m which can result in a long initialization time. However, the approach requires measurements of the steering angle, pedal state and an inertial measurement unit (IMU) making it hard to predict other objects than the ego vehicle. Also, relations between objects are not considered in the already high-dimensional feature vector.

Explicit models can be seen as a complementing and more transparent alternative to learning based approaches, allowing for engineered models and assumptions like a cooperative driving style with multiple traffic participants. Microscopic car-following models [11] were intensively studied in the last decades to predict and simulate traffic flow. The work of [12] uses the Intelligent Driver Model (IDM) [13], to infer driver intent at intersections. Vehicle motion is compared to a learned intention conditioned velocity profile to infer maneuver classes. The influence of a preceding vehicle is successfully compensated by the car-following model where the observed trajectory of the past few seconds is compared with simulated data. Four alternative parameter sets are used in the model to cope with different driver styles. However, since we aim at trajectory prediction instead of maneuver classification we describe a driving style by continuous random variables rather than discrete manual defined classes.

The work in [14] also follows the strategy to use the known past trajectory to infer future motion. A particle filter samples patterns from a probabilistic motion model with a prediction horizon up to 3 s. Instead of a fixed time frame, a distance window of 30 m in the past is used for filtering. Influence of other objects, e.g. leading vehicles, is not modeled. However, the need for a known past trajectory leads to a long time between detecting an object and predicting its movement. Therefore, the approach is hardly applicable for trajectory planning. In contrast, our approach provides prediction results from the first detection.

The authors are with:

¹ Institute of Measurement, Control, and Microtechnology, Ulm University, Germany {firstname.lastname}@uni-ulm.de

Additionally, the use of context information, like road topology, can improve the precision of prediction. Map information is used in [15] to predict the succeeding control input function of the vehicle considered as a dynamic system. The non-linear bicycle model is forward propagated using pseudo measurements for steering angle and acceleration extracted from lanes. However, relations between objects were not considered in this approach.

In this paper we present a probabilistic motion prediction applicable for long-term trajectory planning. We follow the strategy to continuously estimate the driving style parameters of the IDM and the relational motion between objects. These parameters are considered as continuous random variables, rather than discrete behavior profiles. The relational features between vehicles are considered as random input variables of the dynamic system enabling to cope with hidden object. By measuring vehicle accelerations with radar we can provide a driving style estimation from the first detection without the need of long observation time before performing the predict. The use of a particle filter enables to cope with continuous behavior changes inter and intra driver with arbitrarily shaped parameter distribution. The particles are used for forward propagation in the prediction step to gain occupancy probability in time intervals requested by trajectory planning. However, like most approaches we assume legal and cooperative driving behavior according to our models and context knowledge. A performance assessment for monitoring the validity of the used assumptions is therefore provided to allow for safe long-term trajectory planning and vehicle control.

The remaining paper is organized as follows. Section II gives an overview of the considered scenario and the used model. Section III explicates the prediction framework which handles the input datasets and estimates the behavior models. Section IV depicts the experimental setup and evaluates the presented approach. Section V summarizes the preceding sections and presents possible future work.

II. CAR-FOLLOWING MODEL

The car-following scenario considered in this paper is illustrated in Fig. 1. The ego vehicle measures leading vehicles enumerated by α , while the possibility that a leading vehicle is not measured due to a large distance is considered in the algorithm. The IDM is used to model the behavior of vehicles (α) with driving style parameters $\mathbf{x}^{(\alpha)}$.

The IDM was invented by M. Treiber [13] to model a realistic traffic behavior. Compared to other microscopic traffic models, an important advantage of the IDM is its robustness of the parameters. The IDM is comparably insensitive to small changes of the parameters, as shown in [11]. These

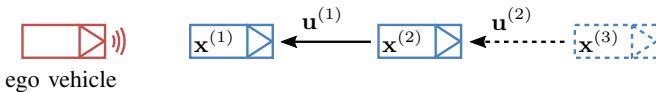


Fig. 1: Vehicles with driving style parameters $\mathbf{x}^{(\alpha)}$ are tracked ahead of the ego vehicle. A third vehicle may be present, but is not detected. The influence of leading objects is modeled with relative kinematics in $\mathbf{u}^{(\alpha)}$.

TABLE I: Variables of the IDM with their physical interpretation.

Behavior Parameter	Interpretation
$a_0^{(\alpha)}$	Maximum acceleration
$b_0^{(\alpha)}$	Desired deceleration
$v_0^{(\alpha)}$	Desired velocity
$s_0^{(\alpha)}$	Minimum distance
$T_0^{(\alpha)}$	Desired time gap
System Input	Interpretation
$v^{(\alpha)}$	Object velocity
$s^{(\alpha)}$	Distance to leading object
$\Delta v^{(\alpha)}$	Approaching rate to leading object
System Output	Interpretation
$y^{(\alpha)}$	Object acceleration $\dot{v}^{(\alpha)}$

advantages made the IDM the model of choice since in this approach it is used to estimate the driving style of real traffic participants, rather than simulating traffic with constantly defined parameters.

From a system theoretical perspective, the IDM describes the dynamics of an object α by

$$y^{(\alpha)} = \dot{v}^{(\alpha)} = \text{IDM}(\mathbf{x}^{(\alpha)}, \mathbf{u}^{(\alpha)}) \quad (1)$$

with a behavior describing parameter vector

$$\mathbf{x}^{(\alpha)} = [a_0^{(\alpha)}, b_0^{(\alpha)}, v_0^{(\alpha)}, s_0^{(\alpha)}, T_0^{(\alpha)}]^T \quad (2)$$

and an input vector

$$\mathbf{u}^{(\alpha)} = [v^{(\alpha)}, \Delta v^{(\alpha)}, s^{(\alpha)}]^T. \quad (3)$$

The system output $y^{(\alpha)}$ is the acceleration of the vehicle α . An overview of physical interpretations of the variables is given in Table I. The IDM model is defined by

$$\text{IDM}(\mathbf{x}^{(\alpha)}, \mathbf{u}^{(\alpha)}) := a_0^{(\alpha)} \left[1 - \left(\frac{v^{(\alpha)}}{v_0^{(\alpha)}} \right)^4 - \left(\frac{s^*(v^{(\alpha)}, \Delta v^{(\alpha)})}{s^{(\alpha)}} \right)^2 \right] \quad (4)$$

with

$$s^*(v^{(\alpha)}, \Delta v^{(\alpha)}) = s_0^{(\alpha)} + v^{(\alpha)} T_0^{(\alpha)} + \frac{v^{(\alpha)} \Delta v^{(\alpha)}}{2 \sqrt{a_0^{(\alpha)} b_0^{(\alpha)}}}. \quad (5)$$

Due to the intuitive interpretation of the behavior parameters $\mathbf{x}^{(\alpha)}$, their influence on the model behavior in (5) allows a physical explanation and as a consequence intuitive limits of a reasonable parameter space. However, the influence of the used parameters highly depends on the situation. For example, mainly in congested traffic ($v^{(\alpha)}$ is small), $s_0^{(\alpha)}$ determines s^* while in free traffic ($v^{(\alpha)} \approx v_0^{(\alpha)}$), the vehicle

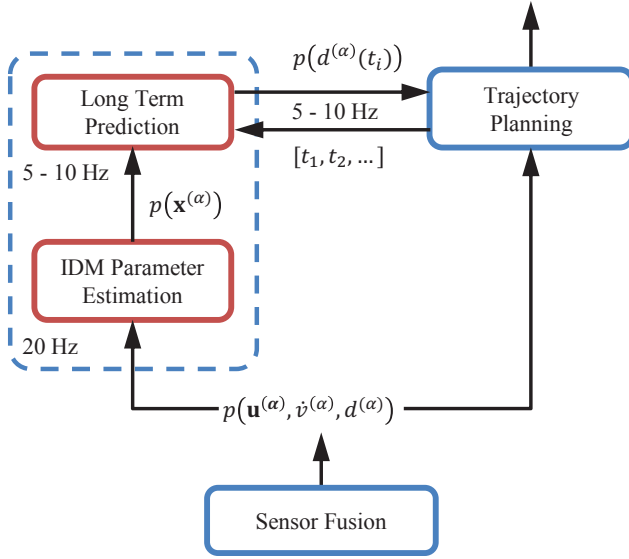


Fig. 2: System overview; The algorithm consists of two steps: identifying behavior describing parameters of the IDM and predicting the scene using forward propagation.

dynamic is dominated by $T_0^{(\alpha)}$. Experiments in Section IV-A show this effect and also the applicability of the IDM for prediction of real traffic scenarios.

III. PREDICTION FRAMEWORK

The overall system framework is illustrated in Fig. 2. The used trajectory planning algorithm requests probabilistic information of occupied areas at target times $[t_1, t_2, \dots]$, while t_0 denotes the current system time. This output is generated in two steps: First, estimating a probabilistic representation of behavior defining IDM parameters $\mathbf{x}^{(\alpha)}$ and second, forward propagating the dynamic scene using these parameters as initial conditions. The input for both steps is a probabilistic environment model as provided by sensor fusion algorithms. Among others, the environment model contains object kinematics which can be considered as estimates of the input vector $\mathbf{u}^{(\alpha)}$ and the system output $y^{(\alpha)} = \dot{v}^{(\alpha)}$ in (1). Following the strategy of a holistic probabilistic treatment, all system in- and output variables are considered as random variables.

A. Estimating behavior parameters

A particle filter is used to estimate the behavior parameters $\mathbf{x}^{(\alpha)}(k)$ at time step k . A particle ρ represents one object α by $\{\mathbf{x}^{(\alpha,\rho)}, \mathbf{u}^{(\alpha,\rho)}, y^{(\alpha,\rho)}\}$ denoting the object behavior parameter set, the system input vector, and the resulting vehicle accelerations, respectively. In the following, the time index k is omitted to simplify notation wherever all variables in an equation refer to the same time step. The time between k and $k+1$ is denoted by dT .

1) *Filter initialization:* A prior distribution of parameter sets $\mathbf{x}^{(\alpha,\rho)}$ should contain only reasonable parameter sets. This can be achieved in two steps. Initially $\mathbf{x}^{(\rho)}$ and $\mathbf{u}^{(\rho)}$

are sampled equally distributed while $y^{(\rho)}$ is calculated from (1). In a second step, the particles ρ are weighted according to a prior distribution of reasonable vehicle acceleration \dot{v} , followed by resampling. The obtained $p(\mathbf{x}, \mathbf{u})$ is used as a prior distribution in the following. Since no object specific information is used, the prior is independent of the object α . Parameter estimation is performed as soon as a first estimate of the corresponding vehicle acceleration is provided by sensor fusion.

2) *Prediction step:* It is assumed that behavior parameters change only slowly compared to the measurement process. Experiments support this assumption. The process model is therefore

$$\mathbf{x}^{(\alpha,\rho)}(k+1) = \mathbf{x}^{(\alpha,\rho)}(k) + \mathbf{n}_{\mathbf{x}}^{(\alpha,\rho)}(k) \quad (6)$$

with additive white Gaussian noise (AWGN) $\mathbf{n}_{\mathbf{x}}^{(\alpha,\rho)}$, drawn element wise from $\mathcal{N}(\mathbf{0}, \sigma_{\mathbf{x}})$. It denotes the normal distribution with mean vector $\mathbf{0}$ and standard deviation $\sigma_{\mathbf{x}}$, constant model parameters determined in preliminary experiments.

In case a vehicle $\alpha+1$, which leads vehicle α , is not detected and therefore not tracked by the environment perception due to a great distance, the system input $\mathbf{u}^{(\alpha)}(k+1)$ must be estimated for the update step. The process model for the input is then

$$\mathbf{u}^{(\alpha,\rho)}(k+1) = f(\mathbf{u}^{(\alpha,\rho)}(k), n_v^{(\alpha,\rho)}(k), n_a^{(\alpha+1,\rho)}(k)) \quad (7)$$

with $n_v^{(\alpha,\rho)}$ denoting AWGN of the velocity of a tracked object α and $n_a^{(\alpha+1,\rho)}$ denoting AWGN with respect to the acceleration of a possibly undetected object $\alpha+1$. The elements in $\mathbf{u}^{(\alpha,\rho)} = [v^{(\alpha,\rho)}, \Delta v^{(\alpha,\rho)}, s^{(\alpha,\rho)}]^T$ are predicted with

$$v^{(\alpha,\rho)}(k+1) = v^{(\alpha,\rho)}(k) + n_v^{(\alpha,\rho)}(k) \quad (8)$$

for the tracked vehicle α 's velocity,

$$\begin{aligned} \Delta v^{(\alpha,\rho)}(k+1) &= v^{(\alpha,\rho)}(k+1) - v^{(\alpha+1,\rho)}(k+1) \\ &= v^{(\alpha,\rho)}(k+1) \\ &\quad - \left(v^{(\alpha,\rho)}(k) - \Delta v^{(\alpha,\rho)}(k) + a^{(\alpha+1,\rho)}(k) \cdot dT \right) \end{aligned} \quad (9)$$

for the approaching rate to the untracked vehicle $\alpha+1$ and

$$\begin{aligned} s^{(\alpha,\rho)}(k+1) &= s^{(\alpha,\rho)}(k) \\ &\quad + (v^{(\alpha+1,\rho)}(k) - v^{(\alpha,\rho)}(k)) \cdot dT \\ &\quad + 0.5 \cdot (a^{(\alpha+1,\rho)}(k) - a^{(\alpha,\rho)}(k)) \cdot dT^2 \\ &= s^{(\alpha)}(k) \\ &\quad - \Delta v^{(\alpha,\rho)}(k) \cdot dT \\ &\quad + 0.5 \cdot (a^{(\alpha+1,\rho)}(k) - a^{(\alpha,\rho)}(k)) \cdot dT^2 \end{aligned} \quad (10)$$

for the distance to the leading object. Using these equations, the influence of the untracked vehicle can be obtained with a single random variable $n_a^{(\alpha+1,\rho)}(k)$ in

$$a^{(\alpha+1,\rho)}(k+1) = a^{(\alpha+1,\rho)}(k) + n_a^{(\alpha+1,\rho)}(k) \quad (11)$$

Here, $n_a^{(\alpha+1,\rho)}$ is sampled according to the process noise of vehicle motion, while in (8) $n_v^{(\alpha,\rho)}$ is sampled according to

the known velocity uncertainty provided by sensor fusion as described for the update step.

The simple design of the random process can lead to parameter sets $\mathbf{x}^{(\alpha,\rho)}$ reaching invalid areas in the parameter space. Examples are parameters < 0 or values for \mathbf{x} and \mathbf{u} outside a plausible range. For this reason, before the update step, invalid particles are replaced with particles from the prior distribution and the summed weight of all invalid particles is equally distributed over the newly sampled particles.

3) *Update step*: The update step is triggered when sensor fusion provides an updated estimate of $\mathbf{u}^{(\alpha)}(k)$ and $z^{(\alpha)}(k)$ which denotes the updated estimate of $\dot{v}^{(\alpha)}(k)$. We consider $z^{(\alpha)}(k)$ as a pseudo measurement of $\dot{v}^{(\alpha)}(k)$ with known uncertainty, while $y^{(\alpha)}(k)$ is the modeled $\dot{v}^{(\alpha)}(k)$ using the IDM. To cope with the uncertainty of \mathbf{u} , for all particles ρ , $\mathbf{u}^{(\alpha,\rho)}(k)$ is drawn from the distribution provided by sensor fusion, i.e. a Gaussian with mean $\mathbf{u}^{(\alpha)}(k)$ and variance. Subsequently, $y^{(\alpha,\rho)}(k)$ is calculated using (1) for each particle.

To yield an updated approximation of the a posteriori density, the particle weight update follows

$$\hat{w}^{(\alpha,\rho)}(k) = w^{(\alpha,\rho)}(k-1) \cdot p\left(z^{(\alpha)}(k) | \mathbf{x}^{(\alpha,\rho)}(k), \mathbf{u}^{(\alpha,\rho)}(k)\right) \quad (12)$$

normalized by

$$w^{(\alpha,\rho)}(k) = \frac{\hat{w}^{(\alpha,\rho)}(k)}{\sum_{\rho} (\hat{w}^{(\alpha,\rho)}(k))} \quad (13)$$

with

$$p\left(z^{(\alpha)} | \mathbf{x}^{(\alpha,\rho)}, \mathbf{u}^{(\alpha,\rho)}\right) = \mathcal{N}\left(z^{(\alpha,\rho)} | y^{(\alpha,\rho)}, \sigma_z^{(\alpha)}\right). \quad (14)$$

Here, $\mathcal{N}(\cdot | \cdot, \cdot)$ denotes the normal distribution at $z^{(\alpha,\rho)}$ using the normal distribution with mean $y^{(\alpha,\rho)}$ and standard deviation $\sigma_z^{(\alpha)}$, which is also provided by sensor fusion. The particle filter result represents the PDF

$$p\left(\mathbf{x}^{(\alpha)}\right) \approx \sum_{\rho} w^{(\alpha,\rho)} \cdot \delta\left(\mathbf{x}^{(\alpha)} - \mathbf{x}^{(\alpha,\rho)}\right). \quad (15)$$

To eliminate samples with small weight, resampling is performed at every cycle. For each object N particles are drawn from the current particle set with probability $P(\rho^{(\alpha)}) \sim w^{(\alpha,\rho)}$. The new weight is then assigned to $1/N$ for all particles.

B. Probabilistic vehicle prediction using the IDM

Prediction is triggered on demand from trajectory planning, which requests a probabilistic estimation of the evolved scene at target times $[t_1, t_2, \dots]$. The prediction algorithm is based on cyclic forward propagation using the intelligent driver model. The time $d\tau$ between the discrete prediction steps l is chosen to a maximum time $d\tau_{\max} = 0.1$ s. To meet the next upcoming target time steps t_i , $d\tau$ is reduced if $\tau(l) + d\tau_{\max} > t_i$ such that $\tau(l+1) = t_i$. Following the strategy of a holistic probabilistic processing, the algorithm is based on Monte-Carlo simulation, recycling the particles ρ from IDM parameter estimation.

1) *Prediction initialization*: The start conditions of the environment are acquired from the latest estimates of sensor fusion and IDM parameter estimation. For the applied Monte-Carlo simulation, the particles from IDM parameter estimation are augmented by $d^{(\alpha)}$, the longitudinal vehicle position on the lane in Frenet space. Thus, the particles represent the PDF $p(\mathbf{x}^{(\alpha)}, \mathbf{u}^{(\alpha)}, \dot{v}^{(\alpha)}, d^{(\alpha)})$. The components $\mathbf{x}^{(\alpha,\rho)}(t_0)$, $\mathbf{u}^{(\alpha,\rho)}(t_0)$ and $\dot{v}^{(\alpha)}(t_0) = y^{(\alpha)}(t_0)$ can be directly copied from parameter estimation at initialization, while $d^{(\alpha,\rho)}(t_0)$ is drawn according to the distribution provided by sensor fusion.

2) *Forward propagation*: If a leading vehicle is not tracked, its acceleration is forward propagated using the constant acceleration model

$$\dot{v}^{(\alpha+1,\rho)}(l+1) = \dot{v}^{(\alpha+1,\rho)}(l) + n_{\ddot{v}}^{(\alpha+1,\rho)} \quad (16)$$

where $n_{\ddot{v}}^{(\alpha+1,\rho)}$ is drawn from $\mathcal{N}(0, \sigma_{\ddot{v}} \cdot d\tau)$, the normal PDF with mean 0 and $\sigma_{\ddot{v}} \cdot d\tau$, denoting the standard deviation of the jerk \ddot{v} with the gain factor $d\tau$. Invalid particles, e.g. with very high $|\dot{v}^{(\alpha+1,\rho)}(l+1)|$ are resampled from the prior to avoid implausible predictions.

The kinematics are forward propagated using

$$v^{(\alpha,\rho)}(l+1) = v^{(\alpha,\rho)}(l) + \dot{v}^{(\alpha,\rho)}(l) \cdot d\tau \quad (17)$$

and

$$d^{(\alpha,\rho)}(l+1) = d^{(\alpha,\rho)}(l) + v^{(\alpha,\rho)}(l) \cdot d\tau + 0.5 \cdot \dot{v}^{(\alpha,\rho)}(l) \cdot d\tau^2. \quad (18)$$

The behavior parameters are modeled with

$$\mathbf{x}^{(\alpha,\rho)}(l+1) = \mathbf{x}^{(\alpha,\rho)}(l) + \mathbf{n}_{\mathbf{x}}^{(\alpha,\rho)}(l) \quad (19)$$

where the process noise $\mathbf{n}_{\mathbf{x}}^{(\alpha,\rho)}(l)$ is drawn from a uniform distribution within predefined bounds. The use of the simple random process can yield to invalid parameters in a particle, like parameters < 0 or outside a plausible range. Particles containing invalid parameters are eliminated by resampling them from the prior.

Calculating s and dv in the input vector $\mathbf{u}^{(\alpha)}(l+1)$ is straightforward using (17) and (18) from the vehicles α and $\alpha+1$. However, if no leading vehicle $\alpha+1$ is included in the environment model, (8), (9) and (10) are used to update the input vector of vehicle α while the acceleration of the unknown object is modeled with (16). Also for the input vector, sampling invalid or implausible values must be encountered, e.g. $s < 0$.

At this stage, the new object accelerations can be calculated with

$$\dot{v}^{(\alpha,\rho)}(l+1) = \text{IDM}\left(\mathbf{x}^{(\alpha,\rho)}(l+1), \mathbf{u}^{(\alpha,\rho)}(l+1)\right). \quad (20)$$

The forward propagation is performed until the target time t_i is reached. The result is then the approximated PDF

$$p(d(t_i) | p(\mathbf{x}(t_0), \mathbf{u}(t_0), \ddot{v}(t_0), d(t_0)), \sigma_{\mathbf{x}}, \sigma_{\ddot{v}})) \approx \sum_{\rho} w^{(\alpha,\rho)} \cdot \delta\left(d^{(\alpha)} - d^{(\alpha,\rho)}(t_i)\right) \quad (21)$$

for each vehicle α .

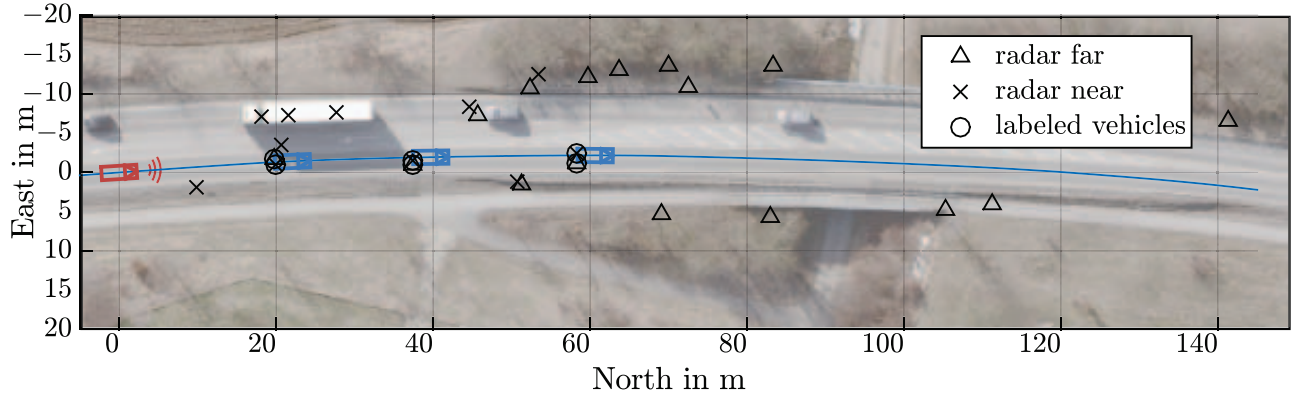


Fig. 3: Data excerpt from Berliner Ring in Ulm illustrated with a satellite image. The ego vehicle on the left measures three vehicles ahead using far and near radar sensors. Clutter and vehicle measurements were labeled manually.

IV. EXPERIMENTS

Experiments were carried out on real traffic data. The data was recorded using the autonomous vehicle of Ulm University [16]. The car is equipped with a differential global positioning system (DGPS) [17] used for high precision ego motion estimation and localization. The vehicle carries a variety of sensors for 360° environment perception. However, for evaluation of the presented algorithms the use of two radar systems in the front of the vehicle is sufficient. The LRR3 radar [18] provides measurements at 12.5 Hz with a detection range up to 250 m and an opening angle of 30°. The used ARS300 [19] includes a radar unit for near distant (up to 60 m) and far distant (up to 200 m) objects. The field of view covers an opening angle of about 56° (near) and 17° (far) with an update rate of 15 Hz. We make use of the radar sensors capabilities to detect occluded cars. The sensor setup easily detects the next two leading vehicles and commonly more as illustrated in Fig. 3.

We extracted object trajectories and removed clutter from the recorded sensor data, to gain a labeled dataset including car following traffic with at least two leading vehicles where at least one is moving. The generated dataset covers different start, stop and following situations on country roads and in urban areas. Due to the recording in real traffic, the dataset provides a broad spectrum of different drivers with different behavior.

The labeled data serves as ground truth information for position, velocity and acceleration of the upcoming two leading objects. The evaluation is limited to two detected leading vehicles and considered the third leading vehicle as undetected, although the sensor setup is often capable to detect these objects. As a consequence, $\mathbf{u}^{(\alpha=2)}$ wasn't provided by sensor fusion but included in the particle based filter process. Since our experiments aim at evaluation of the prediction algorithm, we simulated the sensor fusion module and provided the labeled object data with additive noise to the prediction algorithm. Further, we constantly triggered the prediction every 0.1 s with prediction target times $t_i \in [1\text{s}, \dots, 10\text{s}]$. This leads to 5516 episodes in total for evaluation.

A. Parameter Estimation

Experiments for parameter estimation validate the applicability of the IDM in real scenarios in terms of convergence with a slowly changing process. Fig. 4 illustrates the particle results during a 30 s excerpt with a stopping and starting situation. It can be seen, that the particle filter converges quickly if the parameter is relevant in the situation. At the beginning of the sequence, the desired velocity v_0 condenses approximately at the speed limit. When decelerating the influence of v_0 decreases, leading to a increased variance, while b_0 converges due to the gained influence. When the vehicle stops due to the decelerating leading vehicle, the precision of s_0 increases significantly. The process of all parameters changes slowly, while the uncertainty of the IDM based estimate of \dot{v} is approximately constant around the ground truth vehicle acceleration. Based on this observation, the IDM is considered suitable for prediction in real traffic scenarios.

B. Long-Term Prediction

1) *Evaluation Method:* We compare the IDM based long-term prediction to an approach using the classical constant acceleration (CA) model. This model is commonly used for tracking tasks. The main difference lies in the relational dependencies between objects which are not present in the CA model. The CA model covers kinematics that are physical reasonable, while the IDM model takes relations to leading vehicles together with human driving behavior into account. It can be seen, that the CA model is marginalizing human and relational factors and is therefore more general. However, it will also overestimate the variance of the future scene evolution. Due to its solely kinematic constraints it suits as a minimum performance baseline, where the prediction of a competing method should provide a smaller variance in the predicted future scene. However, a underestimation of the evolution uncertainty might result in future manifestations of the evolved scene that were predicted with a very low probability. Therefore, considering the future scene evolution in terms of a PDF, the manifestation of the future scene evolution should in mean yield a higher value of the predicted

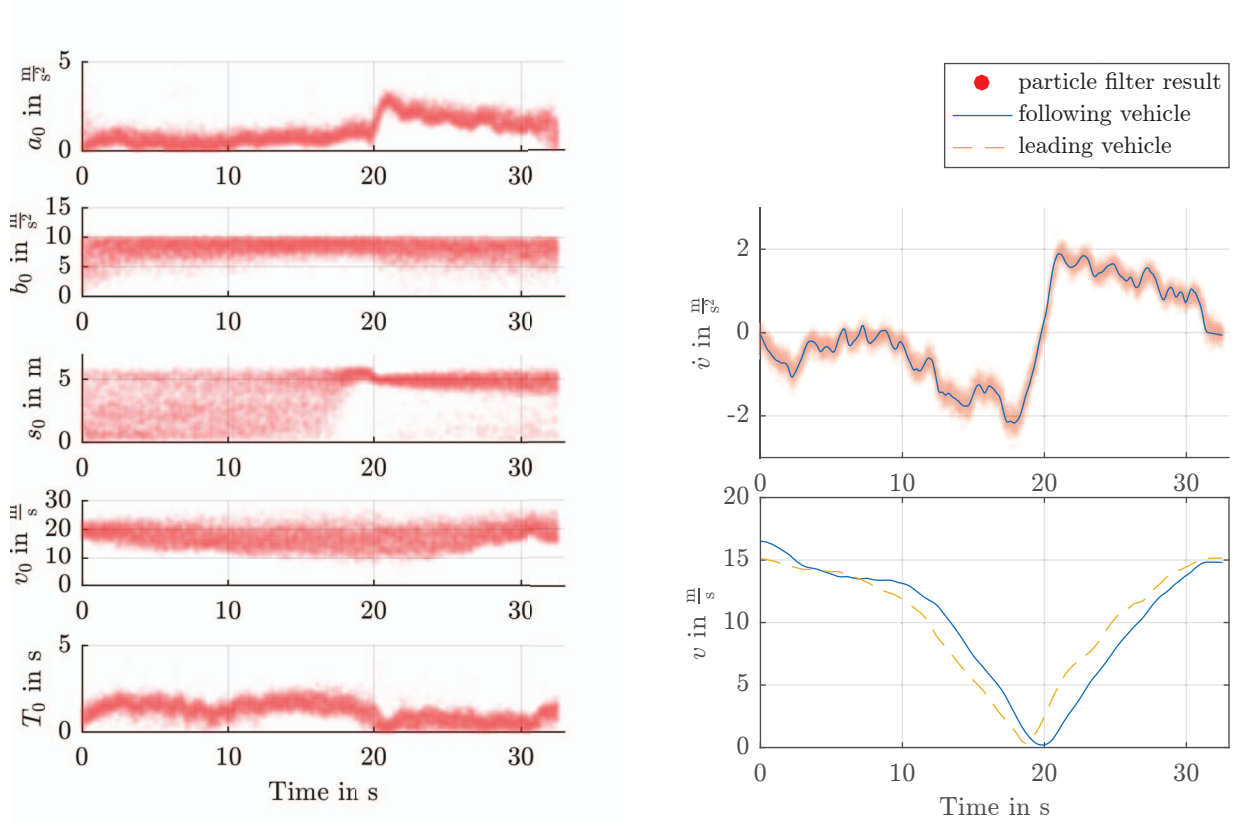


Fig. 4: The particle filter based estimate of behavior parameters converges when the influence is high in the 30 s excerpt (left). The IDM output of the filter has approximately constant deviation to the ground truth (top right). The excerpt includes a car-following stop and start situation (bottom right). The parameters change slowly compared to the measurement.

probability density when using the IDM method compared to the general CA model based method. Otherwise, the used model does not fit to the predicted situation, which can be used as an indicator for self monitoring.

2) *Baseline Method:* The constant acceleration model based method takes probabilistic input data $d^{(\alpha)}(t_0)$, $v^{(\alpha)}(t_0)$ and $\dot{v}^{(\alpha)}(t_0)$ from environment perception and uses the standard deviation of the jerk $\sigma_{\ddot{v}}$ as a model parameter. Like the method under test, the CA-based prediction uses Monte-Carlo simulation with particles ρ . Since no relation between vehicles α and particles ρ is modeled, the notation is omitted in the following. Forward propagation is straight forward following

$$\dot{v}(l+1) = \dot{v}(l) + n_{\dot{v}}(l) \quad (22)$$

with $n_{\dot{v}}(l)$ drawn from $\mathcal{N}(0, \sigma_{\ddot{v}} \cdot d\tau)$,

$$v(l+1) = v(l) + \dot{v}(l) \cdot d\tau \quad (23)$$

and

$$d(l+1) = d(l) + v(l) \cdot d\tau + 0.5 \cdot \dot{v}(l) \cdot d\tau^2. \quad (24)$$

In this approach, in each iteration, when the acceleration or velocity reaches implausible values due to the random process, the particle will be resampled from valid particles. Examples for limits in the considered scenario are $[0 \text{ m/s}, 28 \text{ m/s}]$, or $[-10 \text{ m/s}^2, 10 \text{ m/s}^2]$

The forward propagation is performed until the target time t_i is reached. The result is then the approximated PDF

$$p(d(t_i) | p(d(t_0), v(t_0), \dot{v}(t_0)), \sigma_{\ddot{v}})) \approx \sum_{\rho} w^{(\rho)} \cdot \delta(d - d^{(\rho)}(t_i)) \quad (25)$$

for each vehicle α .

3) *Prediction results:* The output of the prediction algorithm is a PDF of the evolved scene. Unfortunately, the true PDF of the future cannot be observed, rather than a single manifestation of the future. The aim of the evaluation is to show that the IDM based algorithm yields a better PDF describing the future, compared to the CA model based algorithm, considered as base line method. However, the PDF highly depends on initial conditions ζ , that are

$$\zeta_{\text{IDM}} = \{p(\mathbf{x}(t_0), \mathbf{u}(t_0), \ddot{v}(t_0), d(t_0)), \sigma_{\mathbf{x}}, \sigma_{\ddot{v}}\} \quad (26)$$

and

$$\zeta_{\text{CA}} = \{p(d(t_0), v(t_0), \dot{v}(t_0)), \sigma_{\ddot{v}}\} \quad (27)$$

respectively for IDM and CA based prediction. It is nearly impossible to repeat a prediction with exactly the same start conditions, since simulation yields no meaningful results. Alternatively, the experiment $\omega \in \Omega$ can be repeated with varying starting conditions $\zeta^{(\omega)}$. Evaluating the predicted

PDF at the ground truth object position $d_{GT}(t_i)$ for a large number of experiments, yields the expected value of

$$\mathbf{E}\{p(d_{GT}(t_i))\} = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} p(d_{GT}^{(\omega)}(t_i) | \zeta^{(\omega)}) , \quad (28)$$

where $d_{GT}^{(\omega)}(t_i)$ denotes the ground truth object position in the experiment ω at target time t_i , and $\zeta^{(\omega)}$ denotes the according starting conditions. The expectancy of the IDM PDF compared to the base line PDF, both evaluated at the ground truth object position, serves as a metric for the algorithm performance. Fig. 5 shows the results of the 5516 evaluated experiments. In mean, the IDM PDF yields a higher probability density compared to the base line method. This indicates that the PDF of the IDM approach is closer to the actual future compared to the baseline PDF. The evaluation of the presented approach shows a significant improvement over the baseline method.

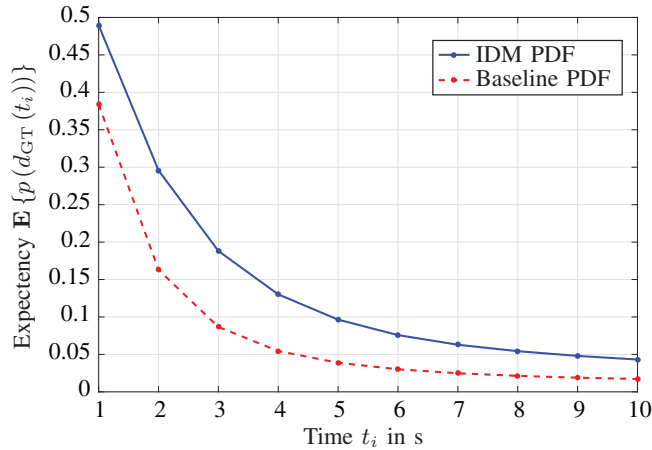


Fig. 5: Expectancy of the IDM PDF compared to the base line PDF, both evaluated at the ground truth object position. The prediction horizon covers discrete target times t_i .

V. CONCLUSION

In this paper we treated the parameters of the IDM as continuous random variables. Thereby, we modeled the driving style as a distribution of behavior parameters. The probabilistic use of the model allows the estimation of the driving style of a tracked vehicle from the moment of the first detection. Experiments on recorded data showed the convergence of behavior parameter estimation and the slow changing process demonstrates a realistic representation of traffic. Forward propagation using Monte-Carlo simulation provides an approximate PDF of the future scene. The presented evaluation method allows performance assessment of the PDF result, although the ground truth is not observable as a PDF. The direct comparison to the presented method with the widely used CA-Model shows a significant performance gain over a prediction horizon of several seconds. A double expected probability density of the object position is reached at prediction times between 2 s and 5 s. Future work includes an advanced process model for changing behavior parameters. Also, the prediction method will be extended to incorporate lateral position prediction and multiple lanes.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union under the H2020 EU.2.1.1.7. ECSEL Programme, as part of the RobustSENSE project, contract number 661933. Responsibility for the information and views set out in this publication lies entirely with the authors. The authors would like to thank all partners within RobustSENSE for their cooperation and valuable contribution.

REFERENCES

- [1] M. Werling, *et al.*, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 987–993.
- [2] J. Ziegler, *et al.*, "Trajectory planning for bertha - a local, continuous method," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, June 2014, pp. 450–457.
- [3] O. Tas, *et al.*, "Functional system architectures towards fully automated driving," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016, pp. 304–309.
- [4] J. Van Den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1263–1278, 2012.
- [5] Y. Bar-Shalom and X.-R. Li, *Estimation and Tracking: Principles, Techniques, and Software*. Artech House Inc., 1993.
- [6] R. Graf, *et al.*, "A learning concept for behavior prediction at intersections," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, June 2014, pp. 939–945.
- [7] F. Kuhnt, *et al.*, "Understanding interactions between traffic participants based on learned behaviors," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016, pp. 1271–1278.
- [8] T. Kumagai, *et al.*, "Prediction of driving behavior through probabilistic inference," in *Proc. 8th Intl. Conf. Engineering Applications of Neural Networks*, 2003, pp. 117–123.
- [9] J. Maye, *et al.*, "Bayesian on-line learning of driving behaviors," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4341–4346.
- [10] J. Wiest, *et al.*, "Probabilistic trajectory prediction with gaussian mixture models," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June 2012, pp. 141–146.
- [11] A. Kesting and M. Treiber, "Calibrating car-following models by using trajectory data: Methodological study," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2088, pp. 148–156, 2008.
- [12] M. Liebner, *et al.*, "Driver intent inference at urban intersections using the intelligent driver model," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June 2012, pp. 1162–1167.
- [13] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.
- [14] C. Hermes, *et al.*, "Long-term vehicle motion prediction," in *Intelligent Vehicles Symposium, 2009 IEEE*, June 2009, pp. 652–657.
- [15] D. Petrich, *et al.*, "Map-based long term motion prediction for vehicles in traffic environments," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, Oct 2013, pp. 2166–2172.
- [16] F. Kunz, *et al.*, "Autonomous driving at ulm university: A modular, robust, and sensor-independent fusion approach," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, June 2015, pp. 666–673.
- [17] ADMA Automotive Dynamic Motion Analyzer, GeneSys Elektronik GmbH, Oct 2016. [Online]. Available: <http://www.genesys-offenburg.de/en/products/adma-series/>
- [18] Chassis Systems Control - LRR3: 3rd generation Long-Range Radar Sensor, Robert Bosch GmbH, 2009. [Online]. Available: http://products.bosch-mobility-solutions.com/media/db-application/pdf_2/en/lrr3_datenblatt_de.2009.pdf
- [19] ARS 30X-2/-2C/-2T/-21 Long Range Radar, A.D.C. GmbH, Apr 2016. [Online]. Available: http://www.conti-online.com/www/download/industrial_sensors_de_de/themes/download/ars_300_datenblatt_de.pdf