intelligent Traffic Flow Assist:
Optimized Highway Driving Using Conditional Behavior Prediction

Thomas H. Weisswange, Sven Rebhan, Bram Bolder, Nico A. Steinhardt, Frank Joublin, Jens Schmuedderich and Christian Goerick

Abstract—Many conventional systems for driver assistance on highways determine vehicle motion based on static situation configurations only. Adaptive Cruise Control systems for example react to vehicles cutting-in from neighboring lanes only after these vehicles have entered the ego-lane. Some more recent systems include prediction of future behaviors, but often neglect mutual influences between these and the best behavior for the ego vehicle.

This paper introduces “intelligent Traffic Flow Assist”, an assistance system that analyzes the current traffic situation and predicts candidate future behaviors conditioned on different options for the ego behavior. It computes vehicle trajectories optimized for safety, comfort and utility for each of the possible future situations and selects the best option for vehicle control. We describe the system architecture and implementation on a prototype vehicle with close-to-production sensors and report on tests performed in different situations on proving ground and public highways.

Index Terms—Behavior Prediction, Situation Understanding, Trajectory Planning, Advanced Driver Assistance Systems

I. INTRODUCTION

In the last years automotive OEMs introduced a variety of advanced driver assistance systems and an increasing number of prototypes demonstrated partially automated driving functions in public traffic. For highway driving some of the most common production systems are adaptive cruise control (ACC) and lane keeping assistance (LKAS) systems.

ACC (see e.g. [1]) controls the longitudinal behavior of a vehicle to adjust its speed to the lower of a desired target speed and the speed of the predecessor vehicle. Conventional ACC systems only react to sensor measurements, e.g. reduce speed when they detect a slower vehicle on the same lane. In case of a cut-in from a neighboring lane, this can result in strong braking due to a late assignment of the vehicle to the own lane. In more advanced systems a reaction is already triggered when the sensors measure a significant lateral motion or displacement of a neighboring vehicle (see e.g. review in [2]). However, in many cases human drivers will adjust their speed much earlier because they can predict the behavior of other vehicles to a certain degree.

When a driver changes his behavior it is usually the effect of adapting to a current driving situation, i.e. he will change lanes to overtake a slower vehicle driving ahead of him if the next lane is free. By analyzing the driving situation of other vehicles, attentive drivers are well capable of foreseeing behavior before the actual motion starts. In 2015 Honda introduced intelligent Adaptive Cruise Control (i-ACC) [3], [4] (Fig. 1) as the first commercially available assistance system making use of such situation-based predictions. It behaves as a conventional ACC for free drive or follow situations, however in case of a cut-in it will react to the future predecessor vehicle even before it starts moving laterally. Our work will partially be based on the prediction models underlying this functionality, but extend it to cover the interactions between the prediction and the selected behavior of the ego vehicle and include the possibility for a (predictive) lane change.

A. Background

ACC systems limit their own reactions to longitudinal behaviors (acceleration, braking), whereas a human driver can obviously also change his vehicle’s lateral motion. LKASystems [5] were the first commercially available assistants to approach the latter behavior. Based on the detection of lane markings (e.g. [6]) and the current heading of the vehicle, the system adapts the yaw rate to reach a desired vehicle orientation and position along the lane center. Recently, first commercial systems that can perform lane-change motions automatically were introduced\(^1\), but for now the driver always

\(^1\)http://media.daimler.com/marsMediaSite/ko/en/9919909
\(^2\)https://www.tesla.com/file/autopilot-lane-change

All authors are with the Honda Research Institute Europe GmbH, Carl-Legien-Straße 30, 63073 Offenbach, Germany.
<first name>,<last name>@honda-ri.de
DOI: 10.1109/MITS.2019.2898969 ©2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
has to take the decision when to initiate it.

Some high-end cars are now equipped with a Traffic Jam Assistant (TJA) or similarly-named functions, which combine both longitudinal and lateral functionality within some strict contextual bounds (i.e., velocities, detected lanes, etc.).\(^1\) In all these cases future changes of the traffic situation due to other vehicles’ behaviors are not taken into account. Current commercial systems control the vehicle within a single behavior class and do not take the decision to initiate a change based on the situation (beyond ACCs switching between different controllers for car following versus free flow [7]).

Lane Change Decision Aid Systems (LCDAS) [8] can be considered as a first step to incorporate behavior decisions into applications, although they currently only warn of, or prevent erroneous human behavior decisions. In general, most systems use dedicated controllers for dedicated maneuvers and do not fully deal with the large behavioral complexity and interaction of vehicles in public traffic.

Beyond commercial assistance systems, research institutions and OEM development departments have presented a large variety of platforms in the past twenty years. They demonstrated more advanced abilities targeting fully automated driving both on test tracks and in public traffic. Early highway systems with functionality similar to today’s TJA were already demonstrated in the 1990’s [9]–[12]. In the DARPA Urban Challenge [13] or the Hyundai Autonomous Vehicle Competition [14] the tasks included driving through urban crossings and in traffic.

More recently, researchers of BMW reported extended tests on German public highways with an automated platform able to initiate lane-changes [15]. In the BERTHA Benz drive a vehicle managed to handle both highway and urban driving within one automated platform [16]. With the availability of methods and technologies for inter-vehicle communication there are also approaches that can directly take into account knowledge about the future behavior of other automated vehicles (Grand Cooperative Driving Challenge 2011 [17]).

C. Overview

In this paper we introduce a system that extends previous work on predictive assistance (i.e. i-ACC) beyond longitudinal control and behavior independent predictions. It also introduces an improved trajectory control algorithm using optimization. Decision mechanisms allow for an automated selection of the best future behavior for our own vehicle. While the system’s output could be used for both automated or assisted driving, our demonstrator provides assistance equivalent to SAE level 2 [26]. This means we compute all parameters for vehicle motion, but the driver always has the final control.

We control the longitudinal motion of the vehicle, but we use an human machine interface (HMI) to inform the driver of the details of the vehicle’s lane change plans. We rely on a manual execution to limit the safety implications of fully automated lane changes. Extensive tests on offline data, simulation and proving ground allowed us to demonstrate the system’s ability in public traffic on German highways (section VI). We named this system “intelligent Traffic Flow Assist” (iTFA), because it targets to keep the driving flow for both the ego vehicle and its surrounding traffic participants as high as possible (also compare [27] for ACC effects on traffic flow). In the next sections we will provide some more details on the different aspects of our systems, how they work together and how they perform in a real vehicle.

B. Related Work

A number of research projects explicitly consider behavior decision making for assisted and automated driving. The classical approach for behavior selection uses a state machine to switch between different controllers (e.g. [7]). Other systems use fuzzy rule-based switching [18] or hierarchical decision trees [15]. Nilssen et al. [19] compute utilities for different lanes based on average speed versus desired speed, average time-gap, and lane length. If the best fitting lane requires a lane-change, its timing is computed by first evaluating all candidate gaps that are predicted to exist long enough within a free reachable set and then choosing the largest one. Similar to most other systems (e.g. [20]), there is first an evaluation of the best behavior and then the best trajectory for this behavior will be computed. An alternative approach is not to model different behaviors but directly control (constrained) trajectories [21].

This is often done with machine learning methods, e.g. through reinforcement learning [22] or supervised deep learning [23]. Reinforcement learning in combination with deep neural networks can also be used in a more controllable/transparent way to select the behaviors and then be followed by classical trajectory planning [24]. Bahram and colleagues [25] used a model predictive behavior selection approach with a Bayesian classifier for the future behavior of other traffic participants. Afterward, they performed a cost-based trajectory and behavior optimization using these predictions.

II. ARCHITECTURE & PERCEPTION

Figure 2 shows the main processing flow of the iTFA system. The input to the system is provided by state-of-the-art sensors on a prototype vehicle. These are further filtered and combined into a representation of the environment (section II). Hypotheses for the future development of the traffic situation are generated by using a combination of physical and context-based prediction methods that additionally take into account the influence of different planned ego behaviors on the surrounding traffic (section III). The system then uses a unified real-time trajectory optimization approach for longitudinal and lateral planning which selects control parameters in a way to balance driver comfort, traffic safety and consistency for each of the hypotheses (section IV). All this information is used in a risk-based decision module which determines if it is for example more appropriate to change lane to give way to a cutting-in vehicle, or to keep the speed in order to prevent a cut-in possibility if any other reaction led to strong safety or comfort compromises (section V).


\(^2\)https://www.continental-automotive.com/de-de/Landing-Pages/CAD/Automated-Driving/Driving-Functions/Traffic-Jam-Assist
In algorithm 1, we provide a schematic description of the processing steps. The details of the different modules will be provided in the following sections.

A. Platform

Our system is implemented on our vehicle platform named “HARP” (HRI-EU ADAS Research Platform), a 2012 Honda CR-V 2.2 i-DTEC modified with sensors and effectors. It is equipped with RADAR, LIDAR and camera sensors to provide redundant 360° perception (see Fig. 3). Most of these components are close-to-production to allow faster porting for possible future product applications.

For our application, we only use a subset of the sensors, namely two commercial long range (up to 250m) 76GHz 4-channel FMCW RADARs with embedded vehicle detection (front and rear bumper), as well as a setup of six Ibeo LUX\(^5\) LIDARs with a range up to 200m around the vehicle that comes with a detection and fusion ECU. Additionally, we use a commercial smart stereo vision camera on the windshield (50° horizontal field-of view), from which we get lane marking detections. All sensors and effectors provide or use data including timestamps in a common reference time to allow fusion and ego-motion compensation.

As a retro-fitted solution to control the longitudinal motion of HARP, we use a Paravan SpaceDrive\(^6\) system. It includes a servo motor that drives two levers pushing on either the accelerator or brake pedal, depending on the motor position. This motor is controlled by an ISO 26262 ASIL D [28] certified ECU which receives commands of a desired pedal position and desired brake cylinder pressure via a dedicated CAN bus. The driver is able to take-over control at any time by overriding brake or accelerator pedal. Pressing the opposing pedal immediately triggers the release of all

---

\(^5\)https://www.ibeo-as.com/
\(^6\)https://www.paravan.de/
actuation. As additional safety measure, we use a dead-man switch which needs to be constantly pressed during tests to enable any of our systems to control the main actuators.

In order to control the main actuators of the vehicle on a higher level rather than providing brake pedal pressure and accelerator pedal position, we have created an interface that allows to control the vehicle based on desired acceleration. The mapping is based on a vehicle model estimated from IMU, odometry, engine torque and gears and allows to compute the desired values independently of slope, air resistance, mass and wheel slip.

To run applications and to acquire and record sensor data, we use a mobile compute host which is able to cope with the high number and bandwidth of inputs from our sensors. For acquisition of the different sensor readings and their transformation into more generic data structures, we use the RTMAPS™ middleware. All further processing in the iTFA system is implemented within our own processing middleware RTBOS [29].

B. Perception

In order to be independent of specific sensor hardware and to be invariant to different sensor modalities, we have introduced a perception abstraction layer through a number of APIs describing data provided by the sensors or processing modules in the perception subsystem. For each category, we define a set of data elements that can be required by higher level processing modules and map data from all sensors that provide this information onto it. Our system uses three different data APIs: EgoVehicle (vehicle state containing e.g. vehicle speed and yaw rate), RoadModels (list of lanes and their boundaries as lists of segments of cubic polynomials), and TrafficObjects (list of detected objects including position and velocity).

All information is provided in a common, right-handed, car-centered coordinate system, with the longitudinal component \( x = 0 \) at the position of the ego vehicle rear axle and the lateral coordinate \( y \) pointing left.

The smart camera provides detected lane markings as a list of cubic polynomials with longitudinal (\( x \)) start and end distances and detected marker types (Fig. 4a). Unfortunately, the range is usually below 60m, while we require lane information for at least ±100m to assign all relevant vehicles to their lanes. Additionally, lane detection sometimes fails in situations with lane-changes and in dense traffic, and there is generally no consistent ID assigned. To deal with this, we use a method to infer lane structure, similar to the one proposed in [30] to extrapolate and track lanes through time and space. Our approach can be restricted to highway scenarios, i.e. parallel lanes with small curvature, so we can use a simpler representation and focus on the propagation of measurement errors. See Fig. 4 for an overview of the processing steps.

C. Scene Representation

For both the prediction of the behavior of other traffic participants as well as the decision about our own behavior, it is important to consider all traffic entities (structural and dynamic) and their relation to each other. We therefore construct a relational scene representation that allows to easily
find relevant connections. At first it specifies the order of
the detected lanes based on their lateral positions and marker
types, and with this the possible lane transitions of entities
that are driving on them. Each detected traffic participant
is then assigned to one of the existing lanes based on
the minimal lateral distance to the lane center. Additionally for
each traffic participant we determine all closest neighbors in
all directions (i.e. left/straight/right predecessors (addressed as
'A', 'B', 'C' respectively) and successors (addressed as 'D',
'E', 'F')). By doing so, all vehicles in one lane are connected
through consecutive predecessor/successor relations. This will
be particularly important when we simulate the effect of scene
changes (e.g. potential ego vehicle behavior) on the relations
of a vehicle (e.g. because it will get a new predecessor if the
original one changes lanes).

III. Prediction

The task of the prediction module is the estimation of the
future behavior of other traffic participants in the ego vehicle’s
vicinity. From overall system perspective, this prediction is
used to plan an appropriate ego vehicle behavior. The earlier
we have knowledge about the future behaviors of other vehicles,
the smoother we can react on it. Behavior prediction
has been a very active research field in the past years. For an
extensive survey we would like to refer the interested reader
to [2], [31], and some more recent examples in [32], [33].

In general, prediction in iTFA is based on our previous
approach (which is the basis for the Honda i-ACC system) as
described in [4]. This system is capable of predicting lane-
changes of other vehicles up to 5s in advance with high
reliability and maturity that led to its market introduction
within Honda’s i-ACC system in 2015. It combines two prediction
approaches, a maneuver-based or physical prediction that relies on perceived motion of a vehicle, and a context-
based prediction or interaction aware method that interprets
the current driving situation of the vehicle.

A. Physical Prediction

The physical prediction approach is visualized in Figure
5 (top). The measured history of recent vehicle positions is
compared to a set of trajectories that were defined based on
typical behavior trajectories from recorded real world data.
For each behavior we use multiple trajectories, all with the
same shape but with different starting-points. By computing a
point-wise distance between the perceived position history of
a vehicle and these trajectories we can select the best fitting
trajectory and from that compute a likelihood for each future
maneuver. To also apply this approach to curved roads, vehicle
locations are rectified using the measured lane geometry. For
details about this approach, please refer to [4]. This method
on its own can provide a prediction horizon for lane-changes
of up to 2s before the car is on our lane.

B. Context-based Prediction

The results of this first prediction are combined with proba-
bilities from a context-based method. The underlying rationale
is that drivers typically change their behavior for a reason,
that is they typically react to their current and future driving
situation. Examples might be a lane-change to overtake a
slower vehicle, or a merge from an entrance lane (see also
a similar earlier approach in [34]).

In highway driving, the future behavior is mainly influenced
by the relative dynamics between vehicles in addition to
traffic regulations. Therefore our prediction approach covers
multiple relations in so-called indicator functions (Υ). For
each behavior to be predicted, we can determine the vehi-
cles in the surrounding that have a relevant influence on its
probability (e.g. for a lane-change left (lcl) we consider the predecessor (B), and the left predecessor (A) and
successor (D)). Figure 5 (bottom) shows some of the measures
[Υ (A, B, D)] evaluated to predict the future behavior of the
red vehicle (the right predecessor of the ego vehicle). Based
on real data, we estimate the influence of these measures on
the behavior probabilities and then combine them into five
phenomenological Indicators.

The Approaching and Fitting Left Gap (AFLG) indicator for
example evaluates how a vehicle is approaching its predecessor
B by calculating the difference of velocity and the time to
contact (TTC(B)) with B (i.e. the time it takes to collide
with B if neither changes its behavior). This is combined
with the availability of a fitting gap on the passing lane, for
example by computing the time to reach a gap (TTG(D, A))
as the distance (Gap(D, A)) between the red vehicle and the
temporal gap (TTC(D), TTC(A)) to vehicles D and A. As
for all other left change indicators, a short time since the last
lane-change of the vehicle (lonlane) will reduce the probability
additionally.

\[
p_{AFLG}(b = lcl | A, B, D) = \prod_{v \in [Υ_{AFLG}(A, B, D)]} p(lcl|v) \quad (1)
\]

Three of the four remaining indicators are also set up to
cover lane-changes to the left, but model different reasons.
For more details please refer to our previous publication [4].
In a next step we assign the probability of the most dominant
indicator to the left behavior.
The fifth indicator estimates a probability for lane-changes to the right using the relative velocity of the predecessor, the gap on the right side and the TTC to the right predecessor. Since we do not have an explicit context-based estimator for the probability of going straight, we use:

\[ p(b = \text{str}) = (1 - p(lcl))(1 - p(lcr)) \] (2)

Finally all three probabilities are normalized by their sum, in order to produce a probability distribution over the three behaviors.

Using a decision threshold that balances the number of correct and false lane-change warnings, context-based prediction can often detect a lane-change maneuver up to 5s before a vehicle enters the ego lane (see detailed analysis in [4]).

C. Conditional Prediction

The context-based prediction algorithm is well suited for predicting the future behavior given a current traffic configuration. However, it does not model the effects of the ego vehicle’s behavior on the evolution of the future situation. Fortunately it is relatively easy to model the effect of hypothetical behaviors of any vehicle in the surrounding of the one we want to predict within our relation- and indicator-based representation. Given a certain behavior of a certain relevant vehicle, we simply recompute the relations as they would appear after that behavior was executed and compute the indicators exactly as before. Assuming for example a lane-change to the left of a left successor vehicle \( D \) (Fig. 5 bottom) would lead to a relation where its successor of \( D' \) is now assigned as new left successor. For our example this would lead to:

\[ p_{AFGL}(b = \text{lcl}|b_D = \text{lcl}) = p_{AFGL}(b = \text{lcl}|A, B, D') \] (3)

The combined context-based result is saved as a probability conditioned on the respective behavior of the respective surrounding car. For simplicity and to reduce computation load, we limit the number of conditionals of each prediction to the behavior of a single vehicle per conditional.

Theoretically we would need to combine the conditioned probability with the probability of the conditional itself, as was for example done in an earlier work [35]. Here we predicted lane-changes of vehicles on an entrance lane with context-based prediction and used the resulting probabilities to compute a lane-change prediction for vehicles on the neighboring give-way lane. We found that for regular highway driving, this will almost always result in very low overall probabilities. We therefore only use hypothetical behaviors of the ego vehicle as conditionals, since these are under our control and can be interpreted as having probability 1. In the rest of the paper we will write \( p(\cdot|\text{lcl}) \) for example refer to a probability conditioned on a lane-change left of the ego vehicle. In general, the framework would also be able to work with more complex interaction predictions with multiple participants and mutual influences as for example presented in [33], [36].

D. Combined Prediction

The physical prediction result of a vehicle can be combined uniformly with all conditioned context-based probabilities. The benefit of combining maneuver-based prediction with intention aware prediction is twofold: First, the use of two independent approaches increases robustness. In the i-ACC work [4] we found the best prediction results in terms of true positives versus false positives with such a combination. Secondly, the maneuver-based prediction complements the context-based prediction by adding a temporal component. The maneuver-based prediction localizes a vehicle on a prototypical trajectory and thus allows calculating when the maneuver will be completed. This benefits the accuracy of trajectory planning.

It does however not make sense to simply multiply the probabilities because this would effectively limit the prediction horizon to that of the physical prediction. Instead, we define three cases with different integration strategies – if both methods assign highest probability to the same behavior option, we let them reinforce each other; if they favor opposing behaviors (i.e. right versus left lane-change), we reset the predictions (i.e. probability for straight is 1.0); in all other cases we simply ignore the physical prediction. With a previous system on a data set with 100 hours of real world driving, we showed that this removed all false positive lane change predictions in the data set [4]. After further analysis of real driving data we had to introduce an exception to the last rule. If the context-based method has highest probability on a lane-change right, we always integrate it with the physical predictions. This effectively limits our prediction horizon for that behavior to 2s because it requires a lateral motion, but it drastically reduces false predictions in the many cases where drivers ignore the (continental European) right driving obligation (which was used in the design of the respective indicators).

To finally provide the trajectory planning with all relevant information, we combine for each possible ego behavior the respective conditional predictions of all relevant vehicles surrounding the ego vehicle in a set of future scene configurations (for examples, see Fig. 6). The probability of the top left configuration \( c_1 \) in Fig. 6 can for example be computed as

\[ p(c_1|\text{str}) = p(b_C = \text{lcl}|\text{str})p(b_B = \text{str}). \]
These configurations specify the behavior dynamics that will take place within the next few seconds. Although this produces a huge number of variants, we can skip a significant part of them when any one of the involved probabilities is close to 0. Additionally, not all vehicle relations are relevant to compute trajectories of all ego behaviors, e.g. vehicles that stay on the right lane do not influence a lane-change to the left. In the case of the direct predecessor, we can also combine two predictions into a single configuration, since for straight driving it does not matter if it changes to the left ($c_{6}$) or right ($c_{7}$) (Fig. 6). All configurations that are kept and have a combined probability above a certain threshold are then passed to the trajectory optimization.

IV. TRAJECTORY OPTIMIZATION

The goal of the optimization module is to create the best ego vehicle trajectory for each possible future configuration. Best in this context is defined by a cost-function that takes into account safety, comfort and utility of the trajectory. To allow for an efficient optimization we decided to use a parametrized representation of the trajectory. For the ego vehicle these parameters are then adapted by the optimization algorithm such that the cost-function becomes minimal while respecting certain constraints. The resulting ego trajectory includes control commands in both lateral (e.g. lane-relative position sequence) and longitudinal direction (e.g. acceleration profile) and an associated cost.

There is a large body of previous work in the area of control trajectory planning, as the topic is at the heart of any system influencing vehicle motion. A general overview of existing approaches can for example be found in [37]. Most approaches are based on a classical planning scheme that first selects the desired maneuver (e.g. a lane-change) and subsequently generates a trajectory for this maneuver while fulfilling some constraints and minimizing cost. This faces the problem that only one solution is generated and alternative solutions cannot be evaluated and the costs are not available anymore at the behavior selection stage.

In [38] different maneuver alternatives are generated by sampling from all possible trajectories within a simplified parameter space (e.g. considering only few surrounding traffic participants without dynamics, only few comfort parameters). Ziegler et al. [39] propose optimization based trajectory planning similar to our approach. Their cost function includes the most relevant comfort parameters and considers the surrounding traffic through constraints. Planning multiple alternative trajectories was deemed computationally infeasible due to a very large and complex search space of ego trajectory samples and the modeling of traffic through constraints. In contrast, our method is able to optimize multiple (>$10$) alternative situations with a larger number (>$100$) of optimization steps in real time.

A. Trajectory Representation

Choosing a parametric representation allows to efficiently optimize the ego vehicle’s trajectory later by modifying only a small set of parameters. To ensure smoothness we use C2-splines [40] for both the lateral and longitudinal part of the trajectory. The optimization is applied to the parameters of the control-points connecting the splines with each other. While modeling the longitudinal motion as an acceleration profile, we use a spatial representation for the lateral motion by mapping longitudinal positions to lateral ones to decouple both motion paths. As a consequence, changes in longitudinal motion do not affect the lateral representation, which is beneficial for the subsequent optimization. It is important to note that both trajectory parts are always coupled through the car dynamics. This is taken into account via the cost-terms during the subsequent optimization by twice integrating the acceleration profile over time to determine the longitudinal position and use this position when calculating the lateral position at a certain time $t$.

For the longitudinal part we decided to use an acceleration profile over time to ensure smoothness and to stay close to the control regime we use in our vehicle (see section II-A). Furthermore, we circumvent the problem of estimating reachable properties (as would be necessary for e.g. position or velocity) of the last control-point. We use a set of three C2-splines (four control-points) to be able to model a pattern of two different acceleration/deceleration profiles and a connection between them. The representation of a possible longitudinal acceleration profile is shown in Figure 7 (top). A control-point for longitudinal motion is defined by a time $t$, an acceleration $a$ at $t$, and a jerk $\dot{a} = da/dt$ at $t$. We fix the number of parameters to ensure an efficient optimization. The first control-point is defined by the current state of the vehicle (e.g. current acceleration, jerk, etc.) measured by sensors. The last control-point at the end of the planning horizon $T$ is kept fixed such that there is a constant motion of the vehicle by setting $a(T) = 0$ and $\dot{a}(T) = 0$. To avoid redundancy in the two remaining control-points we fix the slope $\ddot{a}(t)$ to zero for $t_2$ and $t_3$. This way the spline is exactly defined by the four free parameters $t_2$, $a_2$ and $t_3$, $a_3$. 

![Diagrams](image-url)
For the lateral trajectory we define five control points through position \((x, y)\) and orientation \((\dot{y} = dy/dx)\) (Fig. 7 (bottom)) relative to the origin of the coordinate system centered at the longitudinal position of the ego vehicle and the lateral center of the ego lane. The first control point describes the current state of the vehicle and is given by the measured state of the vehicle containing current position and orientation with respect to the lane. Point five marks the end of the lateral movement and is fixed longitudinally to a position outside the prediction horizon \(x_{\text{end}} = v_{\text{max}}T\) and laterally to the center of the target lane with an orientation along that lane \((\dot{y} = 0)\). We define the start (second control point) and end (fourth point) of the lane-change (in \(x\)) relative to the longitudinal position where the vehicle crosses the lane-boundary \((x_x)\), defined as position of point three. As the lateral motion of a typical lane-change maneuver is symmetrical, we define a span \(\kappa\) between start and end, so \(x_2\) and \(x_3\) are \(\pm \kappa/2\) away from the lane-boundary crossing point. The lateral positions of points two and four are given by the lane-centers with an orientation along that lane \((\dot{y} = 0)\). To determine the slope of the lane-change \((\dot{y} \text{ at } x_x)\), we analyzed a large set of recorded lane-changes (which were also used for the physical prediction models in section III-A) deriving a constant slope of \(\kappa = 100\) m from the mean slope for the targeted velocity range of about 120kph. Even though this empirically determined constant worked well in our tests, the parameter might be dependent on the ego vehicle’s velocity. This leaves only the longitudinal position of the lane-boundary crossing \(x_x\) as a free parameter.

To summarize, using our proposed representation we are left with five free parameters \((x_x, t_2, a_2, t_3, a_3)\) defining the lateral and longitudinal motion of a vehicle.

### B. Traffic Representation

Using this trajectory representation we are able to model a large variety of maneuvers while allowing for an efficient optimization. To actually plan a realistic trajectory in a complex scene, we also need to take into account other traffic participants and their future behavior as represented by the set of configurations passed on by the prediction module. Based on the set of relevant vehicles with their measured properties (position, velocity, etc.) within each configuration, we construct stereotypical trajectories for each relevant vehicle according to their predicted behavior. To determine the free parameters \(t_2, a_2, t_3, a_3\), we assume a constant velocity of the other vehicles, as deviations are mitigated by frequently re-planning the trajectories with updated sensor readings. For the lane-boundary crossing point \(x_x\), we use the estimation of the physical prediction as described in III-A (see also [4]). If no lateral motion is observable yet, we assume that the lane-change of the other vehicle will start immediately by setting \(x_x = x(0) + \kappa/2\). As a result, we get a set of trajectories defining a possible situation dynamics \(S_i\) for each possible configuration \(c_i\) that can be used to compute spatio-temporal cost functions.

### C. Cost Function

Based on the set of possible situation dynamics \(S\) we can define a cost-function used to optimize the parameters of the ego vehicle trajectory independent of the actual probability of the situation. Each \(S_i \in S\) contains lateral and longitudinal trajectories \((y_{i,j}(x), a_{i,j}(t))\) of all relevant vehicles \(j\) taken into consideration when optimizing the lateral and longitudinal trajectory of the ego vehicle \(m_i = (y_{i,\text{ego}}(x), a_{i,\text{ego}}(t))\) being also part of \(S_i\). For a given dynamics \(S_i\) we can calculate the cost \(\text{cost}(m_i|c_i)\) as a weighted sum of \(\Gamma\) cost-terms

\[
\text{cost}(m_i|c_i) = \sum_{k=0}^{\Gamma} w_k \gamma_{i,k}
\]

with situation independent weights \(w_k\). We tuned the weights to \(\vec{w} = (0.1, 0.1, 10.0, 1.0, 10.0, 1.0, 0.01, 0.1)\) during our experiments.

The cost-terms can be grouped as either targeting driving comfort, safety with respect to the surrounding traffic, and utility or deviation from a desired behavior. For the sake of readability we will drop the time dependence of our variables and write \(a = a(t)\) (and respectively for \(s, v\)) and define \(\hat{a} = da/dt\) and \(\ddot{a} = d^2a/dt^2\). Furthermore, as the impact of a deviation from a target value is usually not symmetrical (e.g. a too large headway is fine but a too low headway is critical), we define an asymmetric weighting function \(\theta_{\xi}()\) with tuning parameter \(\xi\) as

\[
\theta_{\xi}() = \begin{cases} e^{\xi} - 1, & \text{if } \cdot > 0 \\ 0, & \text{otherwise}. \end{cases}
\]

We first define cost-terms considering the ego vehicle dynamics that can be related with driving comfort. Two functions force the optimization to minimize the jerk of the trajectory and the transition between the acceleration maneuvers:

\[
\gamma_{i,0} = \theta \left( \max_t (\hat{a}_{i,\text{ego}})^2 - \ddot{a}^2 \right) 
\]

\[
\gamma_{i,1} = \max_t (\hat{a}_{i,\text{ego}})^2
\]

\(\hat{a}\) specifies a jerk level that is still considered as comfortable by the driver. The acceleration based functions \(\gamma_{i,2}\) and \(\gamma_{i,3}\) punish strong decelerations and accelerations according to the minimum/maximum value within the trajectory.

Two safety cost-terms consider the trajectories of the surrounding traffic. First we compute the cost implied by the vehicles in front

\[
\gamma_{i,A} = \sum_{j:v_j > 0} \int_{0}^{T} f(\theta(\Delta h_{i,j}(t), |\Delta y_{i,j}(t)|, \Delta v_{i,j}(t)))dt
\]

with \(\Delta h_{i,j}(t)\) denoting the headway deviation from the target value \(\hat{h}\):

\[
\Delta h_{i,j}(t) = \hat{h} - x_{i,j}(t) - x_{i,\text{ego}}(t)
\]

\(\Delta y_{i,j}(t)\) the lateral offset to the ego vehicle, and \(\Delta v_{i,j}(t)\) the difference in velocity between vehicle \(j\) and the ego vehicle. The function \(f(\cdot)\) weights the headway deviation with the lateral offset such that the cost is zero, when the vehicles do not overlap laterally. It linearly approaches one at the time when vehicle \(j\) overlaps with the ego vehicle by at least half a vehicle width. Furthermore, we accept a larger headway deviation, if vehicle \(j\) is faster. We solve the integral of the complex function \(f\) using numeric integration.
The next cost-terms utilize the time of the lane-crossing of the ego vehicle \( t_x \) to decide if another vehicle will be in front of us at this point in time. We approximate \( t_x \) by solving \( s(t_x) - x = 0 \), with \( s(t) \) and \( x \) given. Similarly, we calculate the start of the lane-change \( t_{start} \) solving the same formula for \( x_{start} = x - \frac{a}{2} \). Based on this, we compute for all vehicles that are behind us on a neighboring lane

\[
\gamma_{i,5} = \sum_{j: x_j(t_{start}) < x} \theta \left( \max_t (a_{i,j}(t)) - \hat{a} \right). \tag{9}
\]

This function implies a cost if the required deceleration \( a_{i,j} \) exceeds a certain limit \( \hat{a} \). The required deceleration is calculated as

\[
a_{i,j}(t) = \begin{cases} 
2 \frac{\Delta x - (x_{i,ego}(t) - x_{i,j}(t))}{(t - t_{start})^2}, & \text{if } t_{start} < t \leq T \\
0, & \text{otherwise}
\end{cases} \tag{10}
\]

where we assume that the other vehicle only starts braking after the ego vehicle has started its lane-change motion (\( t > t_{start} \)) and up to this point has been driving with constant speed. The minimum distance \( \Delta x \) that the other vehicle wants to keep is dynamically defined taking into account the speed of the vehicles by using \( \Delta x = \hat{h}v_{i,ego}(t_x) \) with \( \hat{h} \) being the minimum headway that we also use as a constraint for the ego vehicle trajectory.

The last group of cost-terms model the desired behavior. Two velocity-based terms

\[
\gamma_{i,6} = \theta \left( \max_t (v_{i,ego} - \hat{v}) \right) \tag{11}
\]

\[
\gamma_{i,7} = \int_0^T (v_{i,ego} - \hat{v})^2 dt \tag{12}
\]
target at limiting the velocity by the target speed \( \hat{v} \) and on the other hand force the optimization to stay as close as possible to the target speed at all times. Term \( \gamma_{i,8} \) considers the desired timing of a lane-change. Based on common European driving practice, the lane-change should be as late as possible for a lane-change to the left (cost scaling with \( t_x \)) and as soon as possible for a lane-change to the right (cost scaling with \( T - t_x \)) in order to keep to the right lane as long as possible. Additionally, we restrict the earliest point of starting a lane-change by

\[
\gamma_{i,9} = \theta (\hat{t}_{start} - t_{start}) \tag{13}
\]
to allow for a preparation of the driver at least \( \hat{t}_{start} \) in advance. The last cost-term \( \gamma_{i,10} \) prevents overtaking vehicles on their right lanes. To realize this, we apply (7) to compute a headway for a set of virtual vehicles (but with smaller \( \hat{h} \)). This set consists of vehicles driving slower than the target speed \( \hat{v} \) on any of the left lanes.

While the cost-function represents desired properties of a trajectory, we also formulate constraints that are hard requirements. These arise for example from regulations such as the ISO 15622 [41], limiting the maximum acceleration and deceleration, maximum jerk and minimum headway. Additionally, we constrain the maximum lateral slope.

D. Optimization

Based on the combined cost-function and constraints the parameters of the ego vehicle trajectory are adapted using the derivative-free gradient descent method of [42]. The method allows for direct application of the (inequality) constraints above. To provide a good and valid starting point a heuristic is used to determine the starting parameters \( t_2, a_2, t_3, a_3 \) and \( x \) when we first encounter a new configuration. Parameters \( t_2 \) and \( t_3 \) are set to fixed values (we use 2.0s and 8.0s), \( a_2 = a_3 \) is chosen as the minimal acceleration or maximum deceleration to keep a safe headway to all vehicles. By empirically determining the typical duration of a lane-change as 3 seconds and by allowing for an acceleration and stabilization phase in the beginning and end of the maneuver, we fix the planning horizon \( T \) to 10 seconds. With the acceleration profile of the previous step we can calculate \( x \) such that we cross the lane boundary at a position where we have the desired headway to our predecessor. For configurations seen in consecutive frames we use the result of the previous optimization run shifted by the passed time \( \Delta t \) and driven distance \( \Delta s \). If the second lateral control point overlaps with the first one (i.e. \( x_2 < x_1 \)), we keep the previous trajectory fixed without further optimization to guarantee a smooth lane-change motion. For each \( S_i \) an optimization is run until convergence or for a maximum number of iterations. The resulting parametrized trajectory is then passed on together with its cost and a flag denoting if it adheres to the given constraints.

V. Decision & Control

A. Behavior Selection

To finally control the ego vehicle (or request the driver to do it), we have to select one of the trajectory-pairs (longitudinal plus lateral) computed in the optimization step.

We can make use of both the computed cost of trajectories and the probability for the respective configuration. It is not sufficient to simply choose the trajectory with the lowest cost, since these are only valid if the situation evolves as described. Similarly, choosing the configuration with the highest probability neglects the influence of our behavior decision on the future situation, and might result in a missed opportunity to further reduce effective cost. Furthermore, we also need to factor in certain traffic rules that might limit or bias the selection of a behavior.

We combine cost and probability by computing their product, which is often named “risk”, as is e.g. done in [43]. To meaningfully apply it in our case, we need to evaluate for each trajectory \( m_i \) the risk of each combination with all possible configurations \( \{c_b\} \) (including the one it is optimized on) within the same behavior \( b \) (see Fig. 8):

\[
\text{risk}(m_i) = \sum_{c \in \{c_b\}} \text{cost}(m_i|c)p(c|b) \tag{14}
\]

For each of the up to three available ego vehicle behaviors (straight (\( str \)), lane-change right (\( lcr \)), lane-change left (\( lcl \)), we then select the trajectory with the lowest risk (\( m_l \) in Fig. 8). To stabilize behavior over multiple computations and
to limit the amount of cost-function fine-tuning, we consider all behaviors whose risk lies within a certain small distance to the minimum at a given time. If there are multiple possible selections, we use traffic rules for the final choice, i.e. prefer lcr over straight over lcl unless we detect a vehicle in front that is significantly slower than our target speed (in that case ranking will be str > lcl > lcr to prevent overtaking on the right). The winning behavior and corresponding trajectory will finally be checked to have a risk below a safety threshold. If no acceptable trajectory was found, we will issue a take-over request to the driver and will hand over vehicle control.

Although all generated trajectories cover the multi-second planning horizon, we run our algorithm at a much higher rate (∼10 Hz) to be able to quickly react to changes or prediction outliers. The optimizer will use the current state for the fixed first control point of all trajectories, which, by using the spline representation, will prevent any non-smooth transitions between updates. However, it is still possible that a new lane change trajectory will have a difference in the timing of the lane change motion due to subtle changes in the environment. It means that the “real” trajectory might differ from the smoothly planned optimal trajectory. Such a deviation could result in maneuvers with much higher or lower effective lateral acceleration than planned for the optimal trajectory. Additionally, we want to guarantee a certain consistency of the behavior decisions, that is, not interrupting an ongoing lane-change maneuver. For this reason, we decided to fix the selected trajectory while the vehicle is performing its lateral motion within a lane-change maneuver (i.e. if the ego position is behind the second control point), unless the computed risk of this trajectory exceeds the safety threshold (i.e. we will still run the optimization to evaluate the cost).

The winning trajectory splines are sampled at a rate of 25 Hz and the requested acceleration is fed into the pedal predictor component. The requested lateral offset could optionally be fed into a separate lateral controller (as is done for the simulator experiments).

### B. HMI

To communicate the system state and any lane-change recommendations to the driver, we designed a simple human-machine-interface (HMI). The target was to find an intuitive way to communicate the key elements of the optimized lateral trajectory. Similar to the design presented in [44], we want to transmit both temporal and directional information. Since we plan a very detailed trajectory and the longitudinal control is optimized to the exact timing of the lane-change initiation, we chose a countdown-like HMI. This also provides the driver with sufficient time to check the safety of the situation and turn on the indicator.

Two LED stripes were installed on both sides of the windscreen to inherently communicate the direction. Illumination will be white while the system is running and controlling acceleration and turn red to signal a takeover request or any other condition that cuts the connection to the vehicle controls. When a lane-change behavior is selected and the time to initiation reaches our preparation time $\delta_0$ (set to 3 s in our experiments), we set the top-most of three virtual segments of the LED stripe on the respective side to yellow (Fig. 12 center). Each of the segments represents one second of a countdown to initiation (i.e. middle segment is activated below 2 s, lowest below 1 s). When the countdown reaches zero the full stripe is set to green for three more seconds (Fig. 12 bottom) during which the driver should perform the lane-change motion. We tested a number of variations, e.g. countdown with green lights (see Fig. 11), higher number of sections or later activation, but the selected setup was appreciated most by our test drivers.

### VI. EXPERIMENTS

The iTFA system was tested and evaluated in three setups with increasing complexity and performance requirements.

#### A. Tests on Recorded Data

In a first desktop setting we used recorded data from 5 hours of driving on German highways for a qualitative validation of...
our conditional prediction methods. This set contained 24 situations with vehicles cutting in in front of our vehicle (within sensor range) and 6 cases where the ego-driver performed a predictive lane change to give way to another vehicle (i.e. that vehicle finished a lane change after the initiation of the maneuver of the ego-driver and before the ego vehicle had passed its position). Unfortunately, these numbers do not allow for a proper quantitative evaluation, we will therefore focus on an indicative analysis. Generally, however, inconsistencies between the system recommendation and the actual behavior do influence the prediction performance quite heavily, as was previously observed in evaluations of the i-ACC system. There we found significant variations in prediction performance on recordings with a human driver versus ones with an activated system.

To get a qualitative assessment of the performance, we selected situations in our data, where the original i-ACC algorithm predicted a cut-in. As we are using the same indicators for the prediction, we get the same performance under the straight conditional for cases where the driver does stay on its lane An example for such a situation is shown in Fig. 9 left. The system predicts with high probability that the van on the right lane will cut-in, both given our car would stay on its current lane and even stronger given it would change lane. About 5 seconds later the van does move into the ego lane while the ego-driver has kept his lane. The performance on such scenes can therefore be expected to be similar to statistical results shown in our previous work on i-ACC [4], i.e. around 85% of all cut-in cases should be correctly predicted. For most of the cases, we find that the behavior selection chose the same behavior as the ego-driver, i.e. to stay on lane, because the risk of a lane change was considered higher in comparison. For 3 of the 6 predictive lane change cases, we saw similar predictions and the selection of a lane change, twice the ego lane change started before we could get a prediction. The final situation, which is shown in Fig. 9 right, shows a prediction with significant probability only for the case that the ego vehicle would change to the next left lane. Decision making selected the straight behavior since there was no significant risk involved, but the ego-driver initiated a lane change maneuver a few seconds after the initial prediction. At 5.5 seconds after the first prediction we can see that the white vehicle actually changes lane into this new gap. Our prediction was therefore correct, however we are not able to evaluate, what would have happened if the ego vehicle would have kept its lane.

B. Tests in Simulation

Using recordings it is neither possible to assess the computed trajectory nor the selected behavior, since we cannot influence the vehicle control. We can, however, easily perform those tests in a simulation environment. The perception and control modules of the iTFA system were replaced by functionality provided by the simulator. The system was the same as for real world application except for using lateral control.

The main focus of the simulation tests was on numerical stability of all the parts of the system and the number of take-over requests, since the trajectory optimization module is guaranteed to return results only if they adhere to the safety thresholds, given its internal physical prediction (and the underlying configuration) is true (see chapter IV).

We constructed a large variety of scenarios using the IPG CarMaker simulation software. These were short instances that varied on the free variables of the ego vehicle (i.e. target velocity and target headway, as well as changes of those), the static environment (i.e. number of lanes, starting lane of the ego vehicle), the traffic environment (number and lanes of other vehicles and their velocities), as well as the traffic dynamics (lane change behaviors and smooth changes of their velocities and the timing of these). We focused on regular driving and did not include emergency maneuvers or e.g. cut-in maneuvers that violate our safety settings, as the system was designed as a comfort function. For each group of setups, we also defined minimum/maximum target measures that had to be fulfilled to pass a scenario. Beyond those measures we were not able to find good measures to quantify the quality of the selected trajectories and behaviors, as the evaluations are often highly dependent on a driver’s subjective preferences. For example the timing of an overtaking/give-way lane-change is coupled to the explicit user setting of the headway size, and user preferences for example towards deceleration cost vary significantly as we could see from our i-ACC experience.

In the first set of cases, we tested the basic ACC functionality, i.e. setting and changing of a variety of target speeds in a single lane, with and without front vehicles with different velocities. The target measures for these scenarios included the maximum difference to the target speed after the initial acceleration or deceleration, the time to reach target speed, the minimum headway at any time and the headway variation after the initial approach. We also varied these scenarios with additional longitudinal dynamics of the preceding vehicle, i.e.
accelerations or decelerations that were implemented through a change of a cruise control target velocity to guarantee smoothness. We tested with a particular focus on the initial undershoot of the headway and the duration that it was below the target setting. In the last group of longitudinal test we defined traffic situations that involved lateral behaviors of other vehicles (and therefore an additional lane on the right of the ego vehicle), i.e. cut-in and cut-out maneuvers with an impact on the ego vehicle path. Since the simulator’s driver modeling of traffic is relatively simple and uses triggers that are basically covered by one of our prediction indicators, behavior prediction was always accurate and we could only evaluate the appropriateness of the trajectories.

The second set of scenarios was targeting lateral trajectories and selection between lane change and straight behaviors and its timing. We also used the same headway targets but extended their measurement period to include the time until a lane change is finished. The timing of a lane change depends heavily on the variables in the trajectory cost optimization, and could therefore be adjusted to a driver’s preferences. First, all scenarios from the first set were used with additional lanes to the left and right, to enable different behavior selections. We then added more vehicles with different gaps between them both to the right and to the left, to evaluate the headway of the ego vehicle and the new successor after and during an ego lane change maneuver. Furthermore, we set up scenarios with slow vehicles to the left, to test if the system adheres to the right overtaking regulations. Finally, we constructed a number of scenarios with complex interactions, e.g. a successor vehicle changing lane to overtake while the ego vehicle was approaching a slower predecessor, or a preceding vehicle doing a predictive lane change itself.

In general, we tested the system for successful completion of all scenarios in the simulator after every change of the software before we deployed it on the real vehicle.

C. Tests on Proving Ground

We further tested our system under controlled conditions on a test track with a 2km multi-lane highway section. Using multiple additional ‘traffic’ vehicles we could repeat all test scenarios defined in the simulation, albeit with a smaller range of variations in the underlying parameters. Besides additional noise and delay factors from using real sensors and actuators, the main difference was the use of the lane-change HMI to trigger a manual lateral maneuver instead of an automated one. Figure 10 (a) shows velocity and headway values during a test case where our vehicle approached a slower car on the same lane with solid lane markings to the left. After a short acceleration to close the gap, the velocity quickly settled around the speed of the predecessor (who was on cruise control) with small initial oscillations at a maximum amplitude of ±1m/s. Similarly the headway reached the desired value of 2.5s and, besides some oscillations, basically never surpassed it. The oscillation effect is mainly due to un-modeled non-linearities in the pedal-to-engine-torque mappings.

In Fig. 10 (b) we can see results for a predictive lane-change scenario (also see Fig. 11). The vehicle was smoothly accelerating to its target speed. At 12.5s (circle marker), a slower vehicle on the right lane was predicted to cut-in given an ego lane-change, and a 3s countdown for such a lane-change was started on the HMI2 (Fig. 11 top). The human driver initiated the lane-change and the vehicle center crossed the left lane marking at 15.5s (star marker). Just a few hundred milliseconds before, the other vehicle has been detected as entering our lane (i.e. its center has crossed the lane marking) as can be seen from the short peek in the headway plot.

The plots of Fig. 10 (c) document the dynamics for a case where a lane-change on a slower predecessor had to be withheld until the faster car on the left lane had passed the ego vehicle. A few seconds before the lane border was crossed (star marker, note that headway and \(v_{pre}\) refer to a different vehicle afterward), there was a slight deceleration, visible in the upper plot. This resulted in a headway that reached the desired value at the time of the lane-change and the ego vehicle cut in behind the faster left vehicle at the time when it also approximately had reached the desired headway to this new predecessor.

D. Tests on Public Highways

After the system successfully completed all situations on the proving ground, we tested iTFA in public traffic on a German highway. We drove with target velocities between 80 and 130 km/h and kept the target headway at 2.5s to guarantee sufficient reaction times in case of a takeover. We noticed that in dense traffic many vehicles (particularly from left lanes) cut into our lane with a time gap smaller than the hard minimum constraint (which is also outside of our prediction corridor). This resulted in frequent take-over requests by the system. For this reason we decided to only enable the system at moderate traffic densities to not compromise on safety.

Note that the HMI version in these tests, used green countdown LEDs. This was later changed to yellow as a reaction to test driver comments.
In general we encountered many situation and behavior variations including predictive lane-changes (see as an example Fig. 12), predictive velocity reduction (due to blocked left lanes), delayed overtaking after waiting for a gap and overtake and change right maneuvers. The measured velocity and headway dynamics were found to be similar to those seen on the proving ground, as is shown for two examples in Fig. 13. The plots in 13(b) actually show the measured parameters for the situation from Fig. 12 where the first image is taken at the time marked by the dashed circle and the second one at the time marked by the diamond.

We could show that the system generally performs well in all these situations, although a quantitative evaluation is difficult due to the sparsity of relevant events and the reasons stated in subsection VI-A. Infrequently we also encountered smaller issues due to perception limitations (e.g. wrong lane curvatures led to erroneous scene reconstructions) and due to the delay of our vehicle control hardware setup (i.e. inertia of vehicle engine torque resulted in delayed acceleration to lane-change). However, due to the generic interface design, we believe that this could easily be improved through employment of improved commercial components.

VII. Conclusion

Latest developments on driver assistance systems are integrating longitudinal and lateral control, at least for highway driving. After previously introducing a functional improvement of longitudinal assistance through the application of advanced prediction methods in the Honda i-ACC system, we propose an extension to a holistic two dimensional trajectory generation and vehicle control system that also considers the influence of our own behavior on the surrounding traffic. By generating hypothetical future scene configurations for each vehicle, we could use a context-based prediction algorithm to evaluate the change of behavior predictions in response to an assumed maneuver of another vehicle. We used this to include the effect on other traffic participants in the selection of the best ego behavior. For all potential and predicted future scenes, we then optimized vehicle control trajectories, taking into account costs from a variety of parameters for comfort, safety and consistency. The final decision for the trajectory of the vehicle is based on a combination of prediction probability, cost and traffic rules, which is used to control the longitudinal velocity of the vehicle and eventually provide a temporally precise recommendation for a lane-change to the driver. We demonstrated this system in a large variety of situations in simulation, on the proving ground and also during test drives on public highway. This shows that there is a high potential for advanced behavior prediction methods to also improve combined longitudinal-lateral assistance functions.

Besides further tuning of the many parameters, in particular
in the balance between the cost-terms and some dedicated means to cope with the reality of human drivers that do not adhere to certain safety measures, the system is designed to be able to perform well on any platform that can provide the necessary perception and actuation means (and can actually be vastly improved through those means as well). Potential research areas for further enhancement could be personalization of the optimization parameters to driver style (as e.g. discussed in [45]), or an incorporation of the effects of our own behavior on global traffic flow or with respect to ethical considerations [46].

ACKNOWLEDGMENT

The authors would like to thank Achim Bendig for hardware and testing support and Georg Kotrotsios for adaptations of the PARAVAN controller.

REFERENCES


[33] S. Klingelschmitt, F. Damerow, V. Willert, and J. Eggert, “Probabilistic situation assessment framework for multiple, interacting traffic parti-


Thomas H. Weisswange received his Diploma in bioinformatics in 2008 and the Ph.D. degree in computer science in 2012, both from Goethe University and the Frankfurt Institute for Advanced Studies in Frankfurt, Germany. Since 2011 he has been working at the Honda Research Institute Europe GmbH in Offenbach, Germany on topics including methods for spatial representations, situation prediction, and cognitive system concepts for intelligent automotive systems.

Sven Rebhan received the Diploma degree in computational engineering from the Technical University of Ilmenau, Germany, in 2005. Since then he has been working at the Honda Research Institute Europe GmbH, Offenbach, Germany. In 2010 he received his Ph.D. degree from the Technical University of Ilmenau. His current research interests include methods for situation recognition, behavior prediction and trajectory planning for intelligent autonomous systems.

Bram Bolder studied physics at the University of Wuppertal resulting in a Diploma degree in 2000. In 2005 he received his Ph.D. at the Institute for Neural Computation at the Ruhr University Bochum. Since then he has been working at the Honda Research Institute Europe GmbH, on topics including system integration, entity tracking, synchronization, sensory processing, and real-time programming.

Nico A. Steinhardt received the Diploma degree in electrical engineering and information technology in 2007 from the Technical University of Darmstadt, Germany. From 2007 to 2009, he was working as hardware developer at Peiker acustic, Friedrichsdorf, Germany. In 2014, he received his Ph.D. degree from the Institute of Automotive Engineering at the Technical University of Darmstadt. Currently, he is working at the Honda Research Institute Europe GmbH with main focus on ADAS, sensor data fusion and prototype vehicles.

Frank Joublin received his Ph.D. degree in neuroscience from the University of Rouen (France), in 1993. From 1994 to 1998 he was postdoctoral research fellow at the Institute for Neuroinformatics, University of Bochum, Germany. From 1998 to 2001 he was customer project manager at Philips Speech Processing, Aachen. Since 2001, he is Principal Scientist at the Honda Research Institute Europe GmbH, where he is involved in the development of multisensor test vehicles. His research interests include developmental robotics, and semantic acquisition.

Jens Schmuedderich received the Diploma and Ph.D. degree in computer science from Bielefeld University, Germany, in 2006 and 2010, respectively. Since 2009, he has been with the Honda Research Institute Europe GmbH in Offenbach, Germany, where he is currently working as Chief Scientist and Manager in the area of Cooperative Intelligence. His research interests are behavior prediction in the automotive domain and privacy preserving computation.
Christian Goerick received the Diploma degree in electrical engineering and the Ph.D. degree in electrical engineering and information processing from Ruhr-University Bochum, Germany. During his time in Bochum, he was research assistant, doctoral worker, project leader, and lecturer in the Institute for Neural Computation and Chair for Theoretical Biology. He is currently a Chief Scientist and Group Leader with the Honda Research Institute Europe GmbH in Offenbach, Germany. His current scientific interests are behavior-based vision, behavior generation, cognitive robotics, advanced driver assistance systems, system architecture, and hard- and software environments.