Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Online and Predictive Warning with Risk Maps for Forced Lane Changes

Tim Puphal, Benedict Flade, Malte Probst , Volker Willert, Jürgen Adamy, Julian Eggert

2021

Preprint:

This is an accepted article published in Transactions on Intelligent Vehicles. The final authenticated version is available online at: https://doi.org/10.1109/TIV.2021.3091188 Copyright 2021 IEEE

Online and Predictive Warning System for Forced Lane Changes using Risk Maps

Tim Puphal, Benedict Flade, Malte Probst, Volker Willert, Jürgen Adamy and Julian Eggert

***Abstract*—The survival analysis of driving trajectories allows for holistic evaluations of car-related risks caused by collisions or curvy roads. This analysis has advantages over common Time-To-X indicators, such as its predictive and probabilistic nature. However, so far, the theoretical risks have not been demonstrated in real-world environments. In this paper, we therefore present Risk Maps (RM) for online warning support in situations with forced lane changes, due to the end of roads.**

For this purpose, we first unify sensor data in a Relational Local Dynamic Map (R-LDM). RM is afterwards able to be run in real-time and efficiently probes a range of situations in order to determine risk-minimizing behaviors. Hereby, we focus on the improvement of uncertainty-awareness and transparency of the system. Risk, utility and comfort costs are included in a single formula and are intuitively visualized to the driver.

In the conducted experiments, a low-cost sensor setup with a GNSS receiver for localization and multiple cameras for object detection are leveraged. The final system is successfully applied on two-lane roads and recommends lane change advices, which are separated in gap and no-gap indications. These results are promising and present an important step towards interpretable safety.

***Index Terms*—Advanced driver support system, survival analysis, risk visualization, local dynamic map, motion planning, real vehicle demonstration, risk maps.**

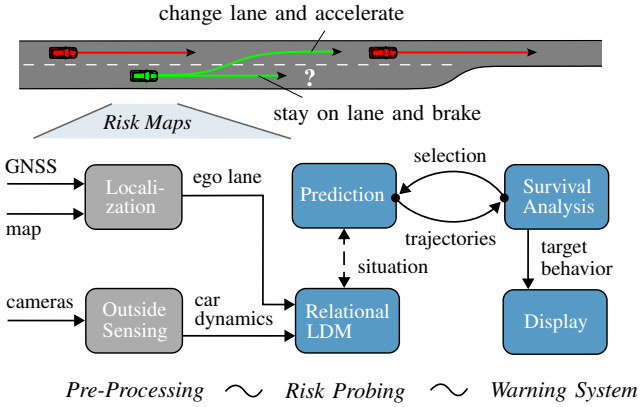


Fig. 1: Risk Maps (RM) is implemented in a test vehicle and supports the driver in its behavior finding for a situation with an upcoming forced lane change. We focus on the modules indicated in blue. RM allows to visualize future risks and uncertainties for the driver.

I. Introduction

ADVANCED Driver Assistance Systems (ADAS) aim for supporting the human driver in its driving task [1] and serve to avoid potential vehicle accidents. Prominent examples of ADAS are, e.g., the adaptive cruise control [2] and various lane keeping aids [3]. They successfully help drivers on highway scenarios. Still, most of the current ADAS are not incorporating explicit driving predictions and uncertainty considerations in their planning. Especially, in real traffic situations with high complexity, the technical requirements for ADAS robustness are high. Such situations include, e.g., dynamic lane changing or tailgaiting behaviors. Here, ADAS that allow to visualize the situation understanding modules in terms of driving risks are beneficial. A technical system must show reasons behind its decisions, which will eventually also increase the trust of the user.

In previous research, we proposed the survival analysis [4] for enhanced collision risk predictions under uncertainties. It is suitable for motion planning in situations with multiple interacting cars and has considerable advantages over common Time-To-X measures, which usually utilize time indicators as a risk approximation. In simulations [5], we proved that our method works reliably and does not produce collisions in a wide range of situations. For this paper, we target to extend and apply the risk framework on a test vehicle for online and predictive warning in forced lane changes (see Fig. 1). We thus show the benefitis of the approach: the easy visualization of the internal calculations.

Lane changes which are forced are complex and occur, e.g., due to road lanes that are ending, construction areas and parked cars. By evaluating behavior predictions with so-called Risk Maps (RM), we target to efficiently obtain lane change timings. A motion trajectory with minimal costs is selected (i.e., which balances risks with utility and comfort). In this process, a crucial component is the Relational Local Dynamic Map (R-LDM) that extracts driving paths, fuses the sensor information and prepares the driving situation.

Essentially, in this paper, we move from simulation into the real world, which requires a prototype. The prototype consists of an inexpensive GNSS and multiple camera sensors. Fig. 1, at the bottom, shows the architecture of the system. In real experiments, RM with R-LDM will run on the test car and create warning outputs for the driver. The system can recommend the target velocities as well as path choices (go left, straight, etc.), in gap and no-gap variations. For the experiments, recordings are hereby drawn from a demonstration of the ADAS in the ITS European Congress 2019, described in [6].

To summarize, the contribution of this paper is threefold. On the scientific side, we build upon our previously published works [4], [5] and [7]. However, we (i) extend the R-LDM

T. Puphal, B. Flade, M. Probst and J. Eggert are with the Honda Research Institute (HRI) Europe, Carl-Legien-Str. 30, 63073 Offenbach, Germany. Email: tim.puphal@honda-ri.de

V. Willert is with the Faculty of Electrical Engineering (FHWS), Konrad-Geiger-Str. 2, 97421 Schweinfurt, Germany.

J. Adamy is with the Control Methods and Robotics Lab (TU Darmstadt), Landgraf-Georg Str. 4, 64283 Darmstadt, Germany.

©2021 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.
DOI: 10.1109/TIV.2021.3091188

to incorporate dynamic cars and their predicted behaviors. We show novel possibilities of how measurements and calculated variables can be saved in a database. Furthermore, we (ii) extend RM and plan besides longitudinal velocities, also lateral path choices, such as lane changes. A major novelty is the visualization of uncertainties in a predictive risk graph. On the technical side, we also propose a first, novel and real proof of concept. Concretely, we (iii) integrate both technologies and apply them on a prototype with an HMI system, highlighting the transparancy of the risk-based planner.

The remainder of this paper is structured as follows: Section I-A summarizes related research for ADAS. We continue with an R-LDM introduction in Section II, and outline the planning method of RM with Section III. Then, in Section IV, the test car and real-world demonstrations are outlined. Section V finally presents conclusions plus future work.

A. Related Work

We will now compare the features of the risk-based ADAS with prior art from research. The focus of this section is put on the fields of trajectory prediction because the R-LDM uses map data to improve the predictions, on related risk methods, which are similar to the RM approach, and classical traffic models for motion planning in lane changes, since they are established baseline models.

In the automotive field, road or map geometries are usually used to enhance the results of driving predictions for vehicles. For example, prevailing ADAS are extrapolating trajectories along paths from sensors and self-driving cars similarly rely on pre-recorded data [8]. As further alternatives to these map-based predictions, recently, learning algorithms are investigated. The authors of [9], e.g., learn typical trajectories in dynamic lane changes with Graph Neural Networks (GNN). Due to the vast behavior options for vehicles, prediction with fail safe backups are also shifting into focus [10].

In contrast to related work, we combine online sensor data with stored crowd-sourced map data. The data is managed by the R-LDM that allows for efficient data queries.

Alongside trajectory prediction, the development of driving risks becomes omnipresent in research. Hereby, risks are often sorted into discrete time or acceleration indicators, probabilistic risks as well as learned risks. Methods based on time metrics (see [11] for Time-To-Brake) convince due to their intuitiveness. However, neglecting uncertainty does not reproduce realistic driving situations. In probability considerations [12], tradeoffs must be found between accurate and computationally inexpensive models. Lastly, learning hazards [13] are powerful for specific complex events but struggle the most in untrained scenarios.

The RM approach employs the survival analysis [4]. On the one hand, this approach is more uncertainty-aware than state-of-the-art risks. On the other hand, the computational costs are reduced due to the analytical probability equations which form its basis. We probe situations with risks in the longitudinal (i.e., ego velocity) and lateral direction (i.e., possible lane change) to retrieve one optimal motion. In this context, probing is the driving prediction of a single ego behavior with an evaluation of induced risks with other cars.

For further motion planning methods, we would eventually like to highlight traffic models in related work. They can easily be used for diverse applications. For instance, the Intelligent Driver Model (IDM) [14] allows to follow preceding cars over kinematic equations. Using the extension of MOBIL [15], lane changes are considered in this process. The lane changes are based on simple predictions and are therefore understandable. However, the IDM does not regard driving uncertainties.

In comparison, there are many advanced planners, such as lane change detectors [16] or potential field approaches [17], which are trying to include uncertainties in the car's positions for the planning. However, they often do not model uncertainty over the predicted time. Using RM, uncertainty-based risk is visualized over the predicted time and not only over the static positions. The proposed ADAS of this paper thus leverages the predictive nature in planning, while visualizing interpretable safety outputs to the user.

II. Representation of Driving Situations

In support systems, clear structures of sensor measurements, trajectory patterns and driving situations are a prerequisite for their efficient real-world application. In this regard, world models are helpful as central storages, whereby so-called Local Dynamic Maps (LDM) are a potential way for their realization. LDM serve for the fusion of ADAS-relevant data.

According to the spirit of a Local Dynamic Map, we construct a database that represents several layers depending on their dynamicity. Dynamic entities, such as traffic participants, are stored on the highest layer. The next layer is consisting of transient data, such as information on traffic signal phases, traffic congestion or road condition. On the second bottom layer, then, quasi-static objects, such as building outlines, are stored. The difference between the transient and quasi-static data is the dynamicity, which is changing on hours scale in the first and on the scale of days in the latter case. Finally, the bottom layer serves to manage static data of the environment, which includes, e.g., spatial geometry data of lower layers that can be enriched by dynamic entities, e.g., traffic participants, of higher layers.

The left part of Fig. 2 illustrates all of the four layers from the R-LDM concept. However, this paper focuses on the static and dynamic layer. Our LDM implementation, the so-called Relational Local Dynamic Map (R-LDM), is hereby realized as a graph. The R-LDM embodies an interconnected graph of nodes. Since relations present first order citizens of a graph, connections between entities (e.g., spatial or temporal) can be described in a straightforward way. Attributes of nodes present an additional possibility to store information that is directly linked to a certain node.

This section builds upon this R-LDM, initially introduced in [7], and describes extensions towards the consideration of driving risks. We start with the description of static map data in Section II-A, show how to manage dynamic entities with the graph in Section II-B and finish with the processing of sensor inputs in Section II-C.

Especially Section II-B and Section II-C show how to connect trajectories as well as driving risks in the R-LDM, which

©2021 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.
DOI: 10.1109/TIV.2021.3091188

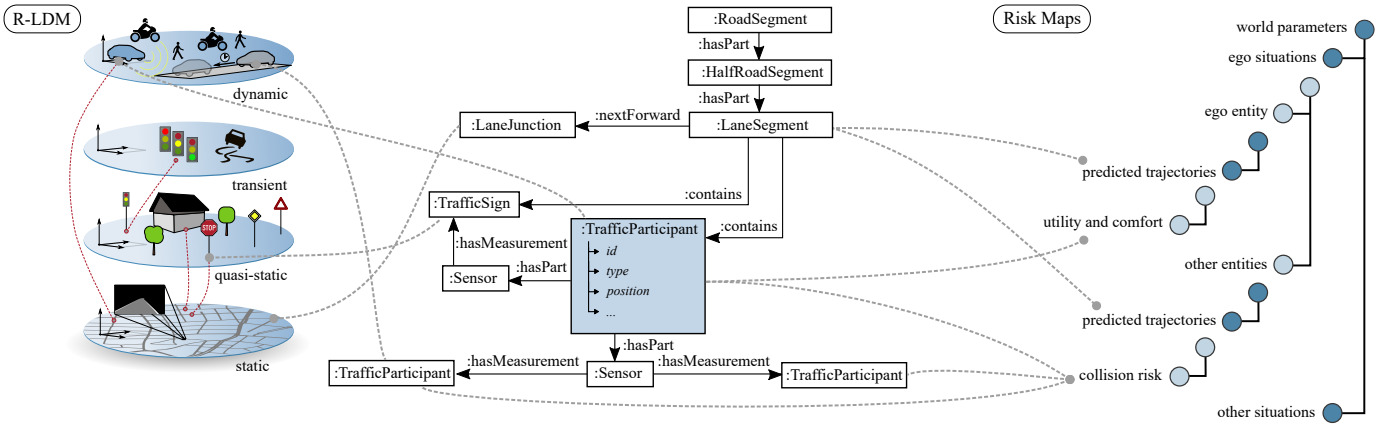


Fig. 2: Interaction of Relational Local Dynamic Map (R-LDM, left) and Risk Maps (right). Structured information, e.g., driving paths, from the local environment and traffic situation is leveraged for predicting trajectories, determining collision risk, utility and comfort. The data structures are connected in different levels of granularity, see dashed lines. Note that the parts of the figure can stand alone.

was not described in the original R-LDM implementation, published in [7].

A. Static Environment

Tasks such as the estimation of collision risk strongly benefit from knowledge on the current and future, i.e., predicted, traffic situation. Such situation evolution can be described by a list of potential paths a driver can take. Each path represents a sequence of positions that is derived from map data and builds the basis of the system's predictions.

The R-LDM enables the storage of road geometries on three levels of detail, i.e., road, half-road[1] and lane level. In terms of the graph, each level of detail is represented by a hierarchy of sub-graph patterns that consist of one central node (e.g., label ":LaneSegment") as well as related child nodes. For example, a road is connected to up to two half-road nodes, while each half-road node can be connected to an arbitrary amount of lane nodes.

Going into further detail, the focus now lies on the lane-level geometry since it builds the basis of the path retrieval. Yet, the information applies to road and half-road level accordingly. A :LaneSegment node stores the inherent information, such as a center polyline defining the direction of travel, as an attribute. Further node attributes include, amongst others,

- road type,
- surface material,
- lane marking type and
- road curvature.

Subordinate entities, e.g., lane markings or boundaries detected by sensors, are represented by individual nodes and are connected via ":hasPart" relations. Further connected nodes include traffic signs as well as further entities of the same or different R-LDM layers that are related in a direct or indirect sense.

The list of attributes and the level of detail can be chosen in accordance with the individual use case. However, also the omission of information can be beneficial, since the computing time correlates with the amount of data stored in the graph.

Regarding the acquisition of raw map geometry data, common approaches are the extension of publicly available sources (e.g., OpenStreetMaps [18]), the parsing of digital orthophotos (satellite imagery) or the derivation of road geometry from pre-recorded GNSS trajectories. Assuming that a vehicle drives in the center of the road, the latter approach natively presents a suitable means for validating the hypothesis.

B. Dynamic Environment

As already mentioned, a crucial group of objects are the traffic participants, such as, vehicles, cyclists, and pedestrians. This is especially the case for a predictive warning system for dynamic lane changes. In this context, the R-LDM can be interpreted as an ego-centered knowledge hub that represents the surrounding of an agent zero. More specifically, the agent zero is moving within a (potentially) pre-mapped environment that is continuously enriched by measurements and sensed data, which is shown in the middle of Fig. 2.

The ego car, or more general the ego vehicle, is represented by a node. For any type of sensor, a node with the label :Sensor is created that is connected to the entity node via a ":hasPart" link. The type of sensor is stored as an attribute with the key type. Depending on type and amount of information, the actual data can be stored explicitly as attribute or implicitly as a reference to another place as, e.g., a hard drive or a time series database. In the case of explicit storage, it can be inefficient and computationally expensive to push data from an Inertial Measurement Unit (IMU) with 100 Hz. Instead, the sensor data updates are pushed at a pre-defined, suitable frequency to the R-LDM.

Any kind of measurement, e.g., object detection from cameras or GNSS position estimates can be considered, including other traffic participants. Each sensed entity, such as another vehicle, can afterwards be represented by a graph node that is connected via a "hasMeasurement" link.

Additionally, traffic participants are also connected to static elements, in other words, the road infrastructure. One common

[1]Here, a half-road is defined to be the union of all lanes that point into the same direction.

©2021 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.
DOI: 10.1109/TIV.2021.3091188

example is a lane segment that is utilized by vehicles, indicated in the R-LDM by the ":contains" links between the lane and vehicle nodes. In this context, the center of a lane serves as an approximation for the driving path, i.e., the path that is most likely be driven by the vehicle. In multi-lane scenarios, several paths can be determined with discrete sets of behavior, e.g., stay on the lane or perform a lane change.

While a path represents consecutive points in a 2D-space, a trajectory also conveys timestamps for each data point.

For Risk Maps (RM), we require trajectories that are based on the assumption of fixed dynamics (e.g., constant velocity) for predicted times and are described as a sequence of position (x, y), velocity v, acceleration a and jerk j. Furthermore, RM considers benefits of the ego vehicle (e.g., desired speed) and risks between the ego and other vehicles. Fig. 2, on the right, depicts how RM benefits from the R-LDM.

C. Data Processing

The preceding Sections II-A and II-B explained how map data and dynamic entities can be stored in a graph. Now, we illustrate how stored data interacts with acquired sensor data. In this context, Fig. 3 shows the ego (green) and a sensed (red) vehicle that are both projected onto the closest map paths. For doing so, two techniques are employed to process such sensor signals: 1) the alignment of the ego car on a driving path and 2) the filtering of unlikely obstacle detections. In this way, we can afterwards predict the trajectories along map paths.

1) Ego Alignment: The positions $(\mathrm{lat}, \mathrm{lon})$ of the ego car from a GNSS sensors are given. In a first step, we transform the GNSS signals to Cartesian coordinates. The equirectangular projection with a radius of earth r_e and latitude lat_0, which is close to the center of the map, allows for the conversion of geodetic GNSS coordinates $(\mathrm{lat}, \mathrm{lon})$ to Cartesian coordinates (x, y) via the equations

$$\mathrm{x} = r_e \cos \mathrm{lat}_0 \cdot \mathrm{lon}, \tag{1}$$

$$\mathrm{y} = r_e \cdot \mathrm{lat}. \tag{2}$$

Here, the x-axis of the global reference frame is facing east, while y is facing north. We project the ego car on the closest path. This description assumes the earth to be an evenly round globe. However, the reader can choose any earth model that fits the individual accuracy requirements, such as assuming an earth ellipsoid. For deeper insight into map-based lane-level localization, we refer to [19].

2) Obstacle Filter: The visual detection of the surrounding vehicles requires a coordinate transformation of their positions from an ego-relative body frame $(x_{\mathrm{rel}}, y_{\mathrm{rel}})$ into the shared world reference frame (x, y). For this purpose, we use a rotation transformation with the angle Θ between both reference frame. Hereby, we use the standard, right-handed orientation of the world frame. If we additionally assume flat surfaces, we get

$$\mathrm{x} = \cos\Theta \cdot x_{\mathrm{rel}} - \sin\Theta \cdot y_{\mathrm{rel}}, \tag{3}$$

$$\mathrm{y} = \sin\Theta \cdot x_{\mathrm{rel}} + \cos\Theta \cdot y_{\mathrm{rel}}. \tag{4}$$

The equations can be extended to cover 3D-space in a straightforward way. This allows for the consideration of, e.g., non-flat environments.

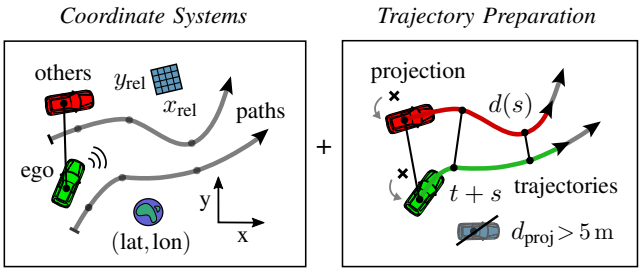


Fig. 3: Pre-processing steps for map-based planning in R-LDM. Left: Car signals are transformed into the same global reference frame. Right: For motion predictions, we filter other cars that are far away.

Similar to the ego alignment on its path, the other cars are projected to their paths as well. However, the main difference is that we only consider vehicles that are close to any path and filter cars with a distance $d_{\mathrm{proj}} > 5\,\mathrm{m}$ away. This geofencing limits position errors from sensors. In the end, we compute the distance d in the global frame for the collision risk calculation. In summary, the R-LDM allows us to prepare situations for risk probing within RM. This method will be described in the following.

III. Probing inside Risk Maps

Using the R-LDM, we are able to fuse measured data into driving situations. However, assessing a multi-lane situation based on its risks thoroughly and, at the same time, in a fast manner remains challenging. The reason lies mostly in the variety of possible predictions for such situations. Additionally, there are underlying driving uncertainties, which arise from car sensors and likewise, e.g., from the driver behavior.

In this context, planning methods must determine an optimal maneuver for the ego driver. With map data, the driving space was constrained and we differentiated between static paths on the one hand, and dynamic trajectories on the other hand. Still, a single path and trajectory that is safe and beneficial for the ego driver needs to be found. Therefore, so-called "probing" inside RiskMaps (RM) will be leveraged. We consider probing as the prediction of an ego driving motion and its evaluation of induced risks. By intelligently probing adequate numbers of fixed trajectories along paths in RM, motions will be efficiently planned.

This section describes the concepts behind the planner RM. For this purpose, we will first explain the trajectory probing in Section III-A with RM and we will continue with the cost evaluation of risk and benefits in Section III-B. In the last Section III-C and Section III-D, we will finalize the description of RM with the extension of path probing, which ultimately allows to warn the driver.

A. Trajectory Probing

Hereinafter, we analyze a car-to-car encounter with collision risks, depicted in Fig. 4. For every ego trajectory variation h, constant velocity is assumed for the other car (i.e., $v = \mathrm{const.}$). For the ego car, we predict and create variations for a piecewise function composed of an acceleration/braking phase and

©2021 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.
DOI: 10.1109/TIV.2021.3091188

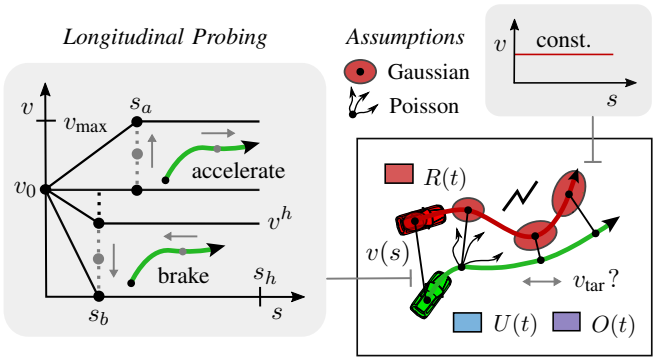


Fig. 4: Trajectory probing within RM. Left: Ego trajectories are planned with different acceleration and braking strengths. Right: Evaluation of the risks, utilities and comforts for constant velocity of another car. The figure shows the risk calculation in more detail.

a zero-acceleration, resp., constant speed segment. Technically, we first sample N_t velocity profiles $v(s)$ on the path that start at the current velocity v_0 and end at the planned velocities v^h in the future time s, equidistantly sampled inside $v \in [0, v_{\max}]$. In total, we thus get

$$v^h = \frac{h}{N_t - 1} \cdot v_{\max} \text{ with } h \in 0, ..., N_t - 1. \tag{5}$$

To reach those end velocities v^h, the accelerations a^h are afterwards calculated that are maximal with $a_{\max}$ in case of an ego trajectory that corresponds to $v_{\max}$. For the cases ending in a full stop $v^h = 0\,\mathrm{m/sec}$ of the ego vehicle, the acceleration a^h becomes minimal with $a_{\min}$. Note that $a_{\min}$ represents hereby a negative acceleration (i.e., braking motion). This leads to

$$\text{if } v^h > v_0\text{: } a^h = a_{\max} \cdot \frac{v^h - v_0}{v_{\max} - v_0}, \tag{6}$$

$$\text{else if } v^h < v_0\text{: } a^h = a_{\min} \cdot \frac{v_0 - v^h}{v_0}. \tag{7}$$

The intervals with either an acceleration or braking phase are planned for the predicted durations of $s \in [0, s_a]$ or $s \in [0, s_b]$, respectively. We formulate the included times as

$$s_a = \frac{v_0}{a_{\max}} \text{ and } s_b = |\frac{v_0}{a_{\min}}|. \tag{8}$$

The included time in the equations is denoted as s since it is not the real time t but the predicted time. We assume a constant velocity for the ego vehicle in the subsequent segment $[s_a, s_h]$ and $[s_b, s_h]$. The parameter s_h defines the time horizon of the velocity profile and is set according to the task. Fig. 4 (on the left) depicts this longitudinal probing.

In the presented planner, a target velocity $v^h = v_{\text{tar}}$ from the RM set is eventually selected based on the explicit tradeoffs between risks $R(t)$ and benefits, which are further divided into the ego utility $U(t)$ and ego comfort $O(t)$, see Fig. 4 on the right. In this process, at least $N_t \geq 3$ ego trajectories have to be sampled so that RM can choose between an acceleration and a braking option. Altogether, RM could potentially represent a fast planner applicable to avoid accidents.

The next section will now describe the mentioned costs to select an optimal motion. Using the survival analysis approach, we include probabilistic assumptions of Gaussian uncertainty (visualized as 2D, red ellipses) and of Poisson uncertainty (escape arrows for the ego cars). However, this section represents a summary of the approach. For more details, please refer to the original publication [4]. We extend and show novel ways how to plan lane changes afterwards.

B. Survival Analysis

We cannot presume that predictions of other cars, e.g., constant velocity, will be followed in reality. The survival analysis thus models uncertainties for their predicted positions with 2D Gaussians that are growing over time. These uncertainties are influenced by sensor uncertainty (dominant in first steps) and behavior uncertainty (dominant in later steps). Finally, a collision probability is given by the overlap of the Gaussians from the ego vehicle and all other vehicles.[2]

Since risk is defined as the probability of this critical event multiplied with a damage outcome, we include a severity term in the final risk output. Besides collision risks, RM allow for the analysis of further critical events, such as upcoming sharp curves. More specifically, RM model the probability to lose control and skid off during lane changes. In these cases, we assume 1D Gaussians around the ego car and look at the lateral acceleration which is influenced by road curvatures.

1) Risk Equations: With the help of a Poisson process, we are able to accumulate probabilities over the future time s (i.e., from collisions and a curve) with the damages into a single, scalar risk value $R(t)$ for the current time t. In detail, we calculate the time difference between two critical events $\tau_{\text{crit}}^{-1}(s)$. Those critical events are then divided into collision rates $\tau_{\text{coll},j}^{-1}$ (with the index defining the considered other car j) and a curve rate τ_{curv}^{-1}.

Due to the included survival function S in a Poisson process, risks that occur further away in the prediction are considered less in the risk $R(t)$. This so-called "escape" effect is modeled for the ego car. We thus describe $R(t)$ by

$$R(t) = \int_0^\infty (\sum_j \tau_{\text{coll},j}^{-1} D_{\text{coll},j} + \tau_{\text{curv}}^{-1} D_{\text{curv}}) S \, ds. \tag{9}$$

For Eq. (9), the severity is hereby formulated in a desired accuracy based on a collision model D_{coll} and D_{curv}.

In the end, we can visualize a risk graph showing risk hot spots for the taken velocity profile that correspond to the other cars. Such a risk graph is plotted in Fig. 5 on the top left. As an example, it shows risk spots for two separate cars, which become visible for trajectories when the ego car come closest to the other car, while the shape of the hot spots depends on the Gaussian uncertainties of the vehicles. At this point, it should be highlighted that the risk graph shows risks over the future time, which is different to common potential field visualizations, such as [17].

2) Utility and Comfort Prediction: The target of RM is to mimize risks as well as to maximize benefits $B(t)$. The total

[2]Note that this collision probability is, in the end, dependent on the relative velocity and distance of the vehicles.

©2021 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.
DOI: 10.1109/TIV.2021.3091188

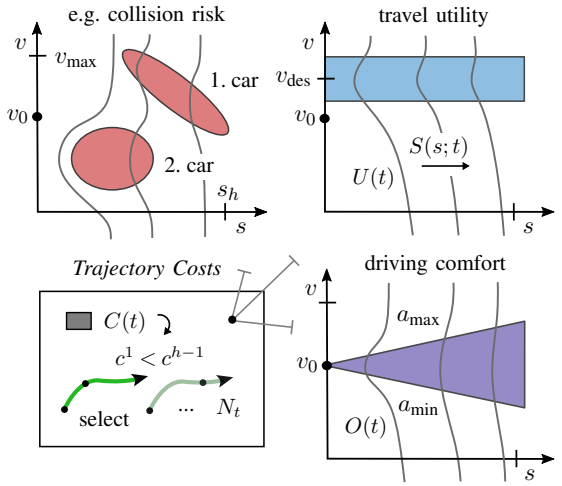


Fig. 5: Visualization of the driving risk (i.e., from collision and curve), utility and comfort costs. The trajectory costs are computed over the future time and then integrated with the survival analysis to obtain a single cost scalar. Note: The graphs should only give a qualitative notion of how the costs may look like.

costs are calculated with $C(t) = R(t) - B(t) = R(t) - U(t) - O(t)$. In this sense, we express benefits with high utility and comfort, both also depicted in Fig. 5.

Utility $U(t)$ depends on the driven ego velocity v_1, which attempts to reduce the general required time to arrive at a goal. Additionally, we consider a desired velocity v_d for individual preferences. For decreasing effects of $U(t)$ at higher times s, we multiply the components with the survival function S and write

$$U(t) = \int_0^\infty (b^t|v_1| + b^d|v_1 - v_d|)S\,ds. \tag{10}$$

Comfort $O(t)$ takes ego acceleration a_1 and jerk j_1 into account and ensures less abrupt switches between selected behaviors. The major target is still to avoid risks. Consequently, comfort is also reduced over the survival function S. For $O(t)$, we gain

$$O(t) = \int_0^\infty -(b^c|a_1| + b^j|j_1|)S\,ds. \tag{11}$$

With the weight parameters b^t, b^d, b^c and b^j, in both $U(t)$ and $O(t)$, the importance of risk versus benefit can be tuned. This enables us to select a single trajectory with minimal costs c^h among all sampled trajectories N_t. Particularly, in Fig. 5, the first trajectory v^1, i.e., with $h = 1$, was chosen because c^1 represents the smallest cost of all trajectories. With the cost graphs of $U(t)$ and $O(t)$, visualized also in the figure alongside risks $R(t)$, we can therefore intuitively indicate the underlying reason for any selection of the planner.

C. Path Probing

For the last module of the resulting warning system, this section outlines path planning. Building upon the ego trajectory probing and cost evaluation from RM, a novel path probing technique is proposed for RM. In this way, the forced lane change of the introduction is solvable. The driver has the support of the HMI, recommending a target motion.

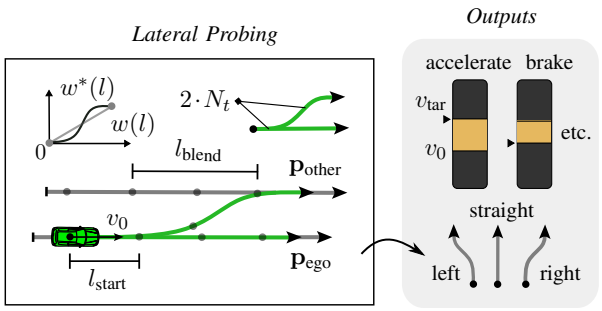


Fig. 6: Signal outputs of RM. By also probing the costs for possible ego paths, we can plan tactical lane changes. Left: Blending of path options and their selection. Right: HMI with target velocity and lane change advice.

As already mentioned, in large multi-lane roads (i.e., highways), a driver has distinct spatial options with the possibility of performing a lane change. A forced lane change arises either for, e.g., following the navigation route, avoiding neighboring cars or both at the same time. In these instances, the lane change path and its start time and duration must be determined. The RM make tactical decisions, which are, e.g., lane changes for a future time (depicted in Fig. 6).

In detail, we compute a path change that serves to blend between centerlines of the ego lane and of a neighbouring lane. The blending begins at a longitudinal distance l_{start}, which is defined as

$$l_{\text{start}} = v_0 \cdot s_{\text{start}}. \tag{12}$$

Accordingly, this distance l_{start} depends on the current velocity v_0 and the future time s_{start}. The length of the blending interval l_{blend} is, in contrast, influenced by the current, lateral distance d_{path} between the two path options. Since the segment end l_{end} is composed of l_{start} and l_{blend}, we retrieve

$$l_{\text{end}} = l_{\text{start}} + l_{\text{blend}} = l_{\text{start}} + v_0\sqrt{l_c \cdot d_{\text{path}}}. \tag{13}$$

In Eq. (13), the parameter l_c defines the scale factor for the increase. Simply put, for higher l_c, the lane change would also take more time.

In a final, next step, we blend the ego path $\mathbf{p}_{\text{ego}}(l)$ into the neighboring path $\mathbf{p}_{\text{other}}(l)$ within the segment l_{blend}, using a sigmoidal weighting term $w^*(l)$. Paths from Section II-A are initially resampled with evenly spaced points. The computed lane change path $\mathbf{p}_{\text{blend}}(l)$ then follows, with

$$\mathbf{p}_{\text{blend}}(l) = (1 - w^*(l)) \cdot \mathbf{p}_{\text{ego}}(l) + w^*(l) \cdot \mathbf{p}_{\text{other}}(l). \tag{14}$$

Generally, linear blending is achieved via the term

$$w(l) = \frac{l - l_{\text{start}}}{l_{\text{blend}}}, \text{with } l \in [l_{\text{start}}, l_{\text{end}}]. \tag{15}$$

and we thus utilize the weight $w(l)$ in a sigmoidal weight function $w^*(l)$ to gain

$$w^*(l) = \frac{1}{1 + e^{-k(w(l) + 0.5)}}. \tag{16}$$

The constant k allows to hereby tune the blending steepness and should be set to achieve a smooth blending.

©2021 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.
DOI: 10.1109/TIV.2021.3091188

Fig. 6 (left box) depicts two prototypical paths. They are calculated with the aforesaid variables. Note that points in the path have to be sampled densely enough for a reasonable blending. With the path blending, we can initiate an immediate lane change by setting $s_{\text{start}} = 0\,\text{sec}$, while a tactical change is obtained with, e.g., $s_{\text{start}} = 2\,\text{sec}$. A utility offset is necessary for the costs in order to incentivize the planner making a lane change. After all, RM choose the path based on its costs.

D. Selection and Warning

For multiple paths, the path blending is done iteratively. With the total path number M_p, ultimately, $M_\text{p} \cdot N_\text{t}$ samples are generated in RM, since ego trajectories are varied on each path. We select a single trajectory with the lowest costs for obtaining the target velocity v_{tar} as well as the target path $\mathbf{p}_{\text{tar}}$. In other words, RM probe in longitudinal and lateral directions. The runtime of RM is constant because we have a fixed sample size. This suits very well for real-time purposes. In the case of Fig. 6, we get $M_p = 2$ and sample $2 \cdot N_\text{t}$ trajectories.

The planned, safe motion is now transferred into a driver suggestion or warning in an HMI so that it is explained to the driver. We compare the planned with the actual one to infer a warning.

As shown in Fig. 6 (right-hand side), the developed HMI contains a velocity scale with the current velocity v_0 and the safe velocity v_{tar}. Depending on the difference $|v_0 - v_{\text{tar}}|$, the driver needs to change its behavior (accelerate, brake, etc.). Furthermore, a directional arrow depicts the path choice from the planner, which can be either “left”, “straight” or “right”. The driver should make a lane change, i.e., left, right, or stay on the lane, i.e., straight motion. The figure shows some output examples of the HMI, which are derived from the solution of the planner.

IV. Real-World Demonstration

We described the concepts behind R-LDM (Relational Local Dynamic Map) as a supporting module for a risk-based motion planning, and the Risk Maps (RM) technology representing an uncertainty-aware planner determining optimal driving trajectories and paths.

This section presents quantitative experiments for the system combination of RM with R-LDM. A demonstration took place within the frame of the ITS European Congress 2019 at Helmond, the Netherlands. The system was run live in front of scientific and non-scientific audiences. The main focus lied hereby on the resulting HMI that has been shown and served to warn the driver. The test car and HMI is summarized in Fig. 7. Concretely, by developing an integrated HMI, the suggestions of the systems were rendered transparent.

A. Test Setup

The used car prototype is called Carlota and was provided by the Spanish research institute Vicomtech. Its sensor setup consists of a mid-range GNSS[3] and cameras. Here, the cameras are four Sekonix [4] cameras, which are combined with two

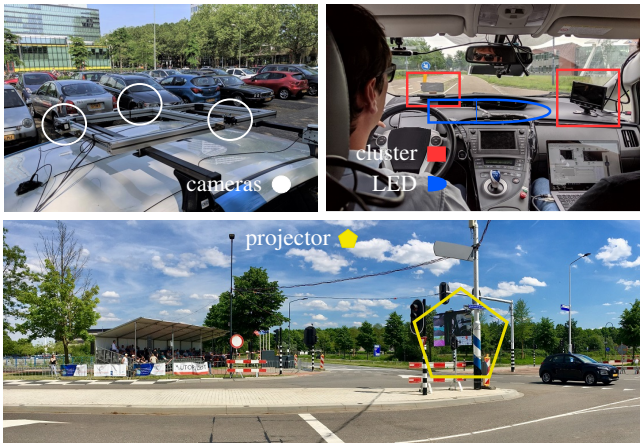


Fig. 7: Top left: Outside cameras mounted on car prototype (three of four cameras shown). Top right: The HMI consists of clusters showing the recommended velocity and LED stripes signaling the path choices (left, right or straight). Bottom: Test area with projector visualizing the risk graph.

Nvidia GPUs[5] and one Intel CPU. For this reason, Carlota can be seen as a low-cost test solution.

In what follows, we analyze and evaluate the given HMI concept, which is built on three components: a risk graph (top left of Fig. 5), a velocity scale (Fig. 6, top right) and lane change recommendations (Fig. 6, bottom right). While demonstrating the system online, the velocity scale was displayed on an instrument cluster, inside the vehicle. For the lane change recommendation, LED stripes were used that were attached to the windshield and are visually conveying suggested driving directions. Additionally, both elements, together with the risk graph, have been presented on a projector to the audience.

In total, the R-LDM serves to provide environment data, i.e., the paths for the risk calculation. We performed situations with three test drivers, maneuvering Carlota and two further vehicles. In each of those runs, RM supports to execute a safe lane change. The end of the road, indicated by cones, forces a lane change of the ego car. In this context, the speed ranges for the tests lied in between $20\,\text{km/h}$ and $40\,\text{km/h}$ with a closed public highway stretch of $1.5\,\text{km}$ length. Our test area is a two-lane road, with both lanes in the same direction and without oncoming traffic. Data recordings from the demonstration were the basis for the evaluation.

We now characterize in Section IV-B benefits that arise from utilizing the R-LDM as a knowledge hub for support systems. Afterwards, within Section IV-C, the results of the RM are described and discussed in regard of analyzing the lane change situations.

B. Sensor Data Fusion

Required inputs for RM are the ego states and its map relation. In a first step, we obtain a (visually) lane-matched ego position from a localization module. The ego vehicle is thus projected onto the lane center, determined by the localization module (i.e., the current ego lane). The R-LDM can now

[3]The product model is given on the website https://www.u-blox.com/de/product/evk-8evk-m8.

[4]See the cameras on http://sekolab.com/products/camera.

[5]See https://www.nvidia.com/en-us/geforce/graphics-cards/rtx-2080-ti.

©2021 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.
DOI: 10.1109/TIV.2021.3091188

provide driving paths. As a second input of the demonstrated system, we need measured positions and velocities of other cars, see Section II-C. In this pre-processing step, the four cameras (i.e., left, right, front and back) are used to detect surrounding obstacles in a 360° view. Specifically, detected bounding boxes of a YOLOv3 neural network [20] are projected from each camera image into a joint 3D world.

In the end, we fuse the pre-processed sensor signals with map data from the R-LDM, the graph-based environment representation. This can give us a driving situation. At every timestep, namely, $N_t = 21$ ego trajectories are sampled and predicted with RM (10 acceleration and 10 braking ramps with different acceleration/braking strength, and 1 constant velocity) to retrieve a situation. Here, sampling is done for the current and parallel lane paths (i.e., $M_p = 2$).

With the middleware RTMaps, we may integrate the software components of R-LDM and RM in a single system. RTMaps reads the sensor inputs and writes final HMI outputs. The system could run with a frequency of around $f = 10\,\mathrm{Hz}$. Our RM can then output a driving path and a velocity that promises safe behavior. As mentioned, the outputs are visualized on the final HMI for driver warning.

C. Results

In the tests, the ego car drives on an ending lane and must take the gap between two vehicles. As we know, the driver may switch to its neighboring lane before or after the passing car (refer to Fig. 1). We will analyze examples of gap and no-gap situations below. These lane changes simultaneously include longitudinal and lateral spatial risks, which make them complex maneuvers.

1) Gap Scenario: For the scenario, recordings have been replayed with three vehicles and, subsequently, we predict their trajectories. Fig. 8 pictures the different situation snapshots with the ego velocity v_0 as well as distances d_1 and d_2 to the two other cars on the neighboring lane. Herein, the planned ego trajectory is colored green, with the predicted trajectories of surrounding vehicles colored red. On average, the other cars drive in the complete stream with constant velocity.[6]

A planned lane change is realized by blending the current path with the parallel path at a start time $t_{\text{start}} = 1\,\text{sec}$ with a duration of 3 sec (for details, see Section III-C). This aligns with usual lane change durations, e.g., according to [21]. For the visualizations, we choose a prediction horizon s_h of 6 sec but the actual risk is evaluated for $s_h = 12\,\text{sec}$. The car signals and possible paths are updated in the R-LDM on demand.

Fig. 8 also illustrates the risk graph. The sampled ego trajectories are plotted as curves of velocity over the future time and the sum of of collision and curve rates with τ_{crit}^{-1} are further visualized. The blue areas in this graph represent low probabilities between $[0\,\%/\text{sec}, 0.5\,\%/\text{sec}]$ and red areas are high values with $[0.5\,\%/\text{sec}$ and also values $> 1\,\%/\text{sec}]$. Finally, the single chosen trajectory is highlighted with green points. RM always tries to find a trajectory that bypasses the red hot spots.

[6]Note that the other trajectories' length relates to a constant velocity, while the ego trajectories incorporate acceleration or deceleration.

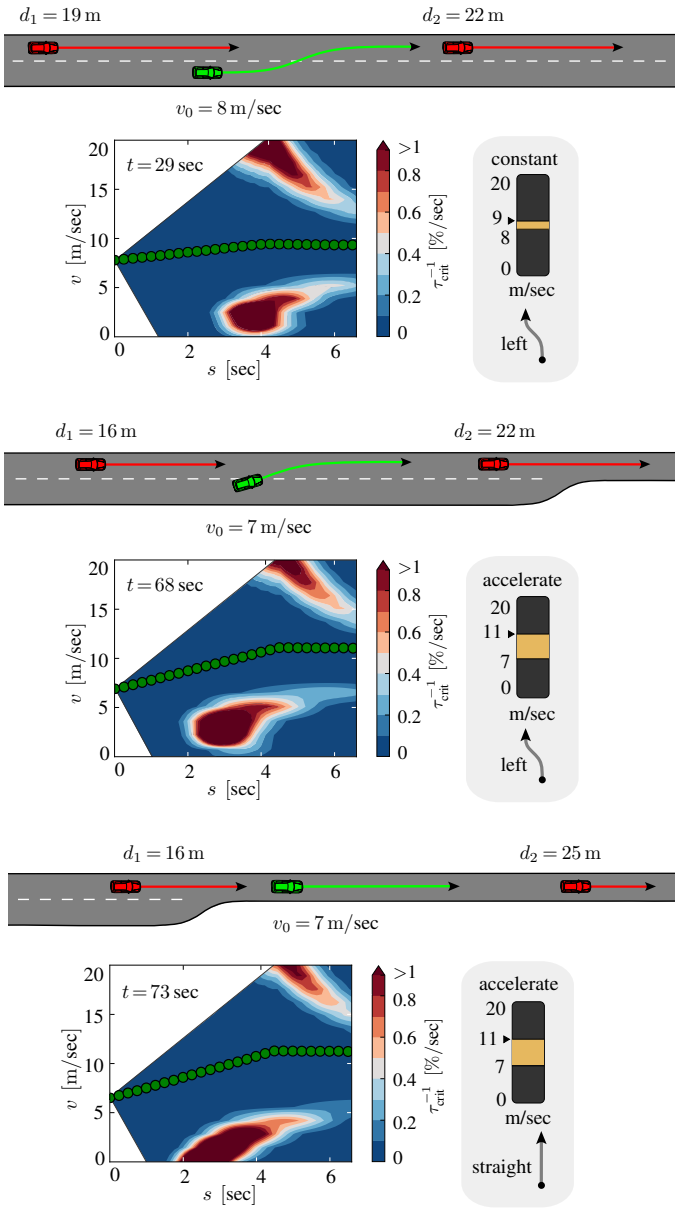


Fig. 8: Behavior of RM and R-LDM in a gap scenario. Given are three snapshots from real-world recordings. The system signals to directly perform a lane change due to sufficient space between the neighboring cars. Top: Road layout and predictive situation, Bottom: HMI with risk visualizations.

In this scenario, RM successfully judges the gap as sufficiently large. Changing the lane with moderate acceleration from $7\,\mathrm{m/sec}$ to $v_{\text{tar}} = 11\,\mathrm{m/sec}$ presents the optimal maneuver. After the lane change, the message on the HMI changes from "go left" to "go straight" (i.e., the target path $\mathbf{p}_{\text{tar}}$). This example emphasizes the possible proactive support that can be provided by RM.

2) No-Gap Scenario: In the previous gap scenario, the system recommended to either drive with constant velocity or to accelerate for performing the lane change. The front and back car were determining the gap. In this second scenario, the gap is too small for safely conducting a lane change. Only after the two other vehicles have passed and drive with appropriate

©2021 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.
DOI: 10.1109/TIV.2021.3091188

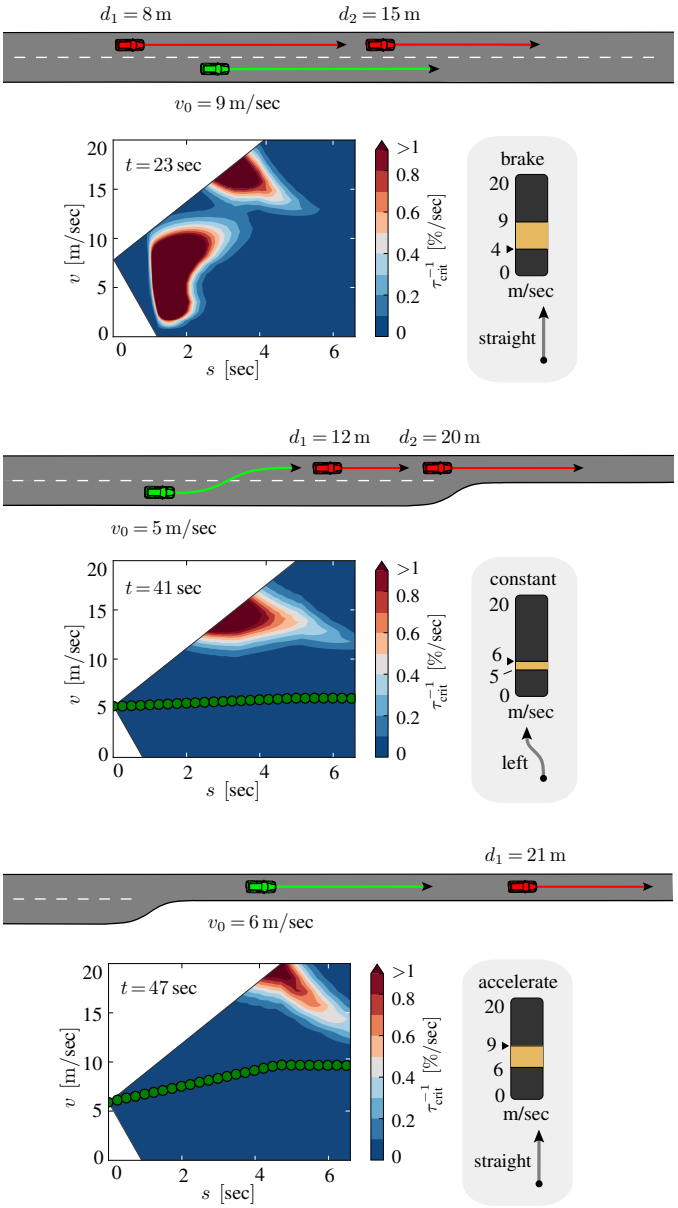


Fig. 9: Behavior of RM and R-LDM in a no-gap scenario. A forced lane change with an acceleration is advised only after the preceding car passes. This represents a safe motion and the driver follows the warning.

distance to the ego vehicle, RM signalizes that the driver could make a lane change.

Within Fig. 9, the preceding car hinders the ego car to go on the target lane. Hereby, we want to analyze more closely the risk graph for the first screenshot of the scenario. If the ego vehicle brakes or drives constantly and makes the lane change, high risks are caused by the preceding vehicle. Due to the close lateral distance, a large risk spot is depicted in the bottom left of the graph. Secondly, if the ego car accelerates and changes lane, there is a high risk of colliding with the front vehicle, indicated by the other red area. The risk graph shows the reasoning behind the system's recommendation output: "brake" and "stay on the lane".

The actual driven speed v_0 and target speed v_{tar} is depicted in the velocity scale. As a reminder, the comfort velocity parameter v_d determines a utility gain and can be chosen by the user. In Fig. 9, the system penalizes velocities that deviate from $v_d = 10\,\mathrm{m/sec}$. To find an ego trajectory in terms of risk versus comfort, we sanction high accelerations as well. RM, therefore, initially recommends to brake until $4\,\mathrm{m/sec}$ because of the ending lane. When the preceding vehicle passed the ego vehicle, RM then correctly recommends to drive with constant velocity $6\,\mathrm{m/sec}$ and v_{tar} is lastly accelerating until $9\,\mathrm{m/sec}$. The ego driver's speed stays around $v_0 = 5\,\mathrm{m/sec}$ and the lane change suggestion is followed.

D. Discussion

As we could see, the combination of RM and R-LDM allowed for a reasonable support of the driver in lane change scenarios. In this Section IV-D, a short discussion concerning the results of the system is given. After showing insights into noise robustness with real-world sensors, the generalization for other driving situations is discussed.

Noise in the vehicles' position and velocity due to the real cameras and GNSS sensors can potentially hinder a faultless operation. In the system, we parametrized the weights in the planning (i.e., for risk, utility and comfort) to achieve robustness against such noise errors. As explained in Section II-C, positions of all vehicles are projected to the closest path, derived from the R-LDM. With regard to risk computations, the noise did not affect the lane change advices of our experiments. Position noise that is constantly present is merely carried into RM in the future time, where the survival analysis compensates its influence.

However, the safe speed output from the planner can slightly fluctuate. This is due to the velocity noise in the ego and other vehicles, which grows over the prediction time. Furthermore, a projection on wrong lanes induces the biggest errors. To tackle this noise propagation and the impact of discrete errors, we introduce a hysteresis in RM. The hysteresis changes the outputs solely when the new selection shows lower risk values for at least 2 sec. The more RM is robust against this sensor noise, the less it will proactively react on the changes in the environment. In the end, a tradeoff was chosen between stable warnings versus better sensitivity.

The given experiments show that a map-based behavior support works online and has benefits in terms of its predictability and transparent risk model. The last Section IV-C served as a first proof of concept that RM can be applied in real-time on a car platform. The tests were conducted with velocities of $30\,\mathrm{km/h}$. Nevertheless, other applications on Spanish city streets (i.e., $50\,\mathrm{km/h}$ limit) and German highway entrances (on average, $100\,\mathrm{km/h}$) show the same functionalities and beneficial properties. For these test cases, we applied RM on other car platforms using higher-cost GNSS and lidar.

In addition, in previous scientific work we showed that the same risk-based planner can be successfully applied in simulation to turning at curves (see [5]), merging at intersections (see also [5]) or crossing at intersections (see [4]). For this reason, we consider the planner to be generalizable to various traffic cases in urban or highway driving due to the generic

©2021 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.
DOI: 10.1109/TIV.2021.3091188

characteristics of the models. The transparent model-based risk approach allows to visualize and understand the reasoning behind its decisions.

V. Conclusion and Outlook

In summary, this paper served to evaluate an application of the Risk Maps (RM) technology on a prototype vehicle in order to show its real-world applicability. In an online demonstration, we presented warning functions for the example of forced lane changes. For this purpose, we extended RM to handle lateral path options.

The system utilizes the Relational Local Dynamic Map (R-LDM) for the alignment, filtering and dynamic layering of car signals with map data. Afterwards, RM are used to predict the current traffic situation into the future and probe trajectories and paths based on driving risks for planning. An improved GNSS position hereby allows to position the ego vehicle on the road. RM can handle noise of the camera sensors for object detection by considering uncertainty by design. In addition, the visualization of a risk graph supports the human driver to understand the driving situation.

Our resulting ADAS was demonstrated on the ITS European Congress 2019 in Eindhoven, the Netherlands. Videos of the results can be found on Youtube: https://www.youtube.com/watch?v=8o3hT3H_gDU. The real-world evaluation shows that the system is capable of giving reasonable advice on target speeds and lane changes, separated in gap and no-gap situations. Effectively, we may improve the prediction capability for the user with a risk-based planner. RM represents a white-box model, which handles sensor noise and complex predictions. This is promising to increase trust on ADAS.

In the current realization of the system, we assume path blending with fixed parameters of duration time. In a revised solution, the parameters should be optimized [22]. Helpful are heuristics to make sure that the lane change is conducted with sufficient space to the preceding and leading vehicle. By using general risks, we believe that RM can implicitly consider dependencies between velocities and paths. This is especially helpful for autonomous driving.

For future research, we likewise envision the use of situation classifiers to improve risk predictions. Potential behavior of other cars need to be explicitly reflected (i.e., lane changes, accelerations or decelerations). Currently, interaction between vehicles is only modeled by a general increase of uncertainty in the trajectories. However, for interaction-intensive situations (e.g., lane changing in dense traffic, abruptly stopping vehicles, suddenly appearing occluded vehicles or similar dynamic conditions), the intention of other cars need to be predicted. Hereby, machine learning methods offer themselves to predict trajectories based on previously measured positions.

Finally, a face detection and gaze estimation from eye pupils [23] were showcased in the demonstration as well. Our planner can benefit from adding such detection technologies. Knowing the gaze allows for the inference of a targeted lane change and helps to determine if a driver is aware of critical objects. We are thus able to limit warnings to non-look cases. Since this reduces the alarm rate and workload of the driver, the social acceptance of the ADAS may be further improved.

Acknowledgment

This work has been supported by the European Union's Horizon 2020 project *VI-DAS*, under the grant agreement number 690772. The authors would like to thank all project partners of the *VI-DAS* demonstrations.

References

[1] K. Bengler, K. Dietmayer, Färber, M. Maurer, C. Stiller, and H. Winner, "Three Decades of Driver Assistance Systems - Review and Future Perspectives," *Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 6–22, 2014.

[2] H. Winner, S. Witte, W. Uhler, and B. Lichtenberg, "Adaptive Cruise Control System Aspects and Development Trends," *SAE Technical Paper 961010*, 1996.

[3] S. Ishida and J. E. Gayko, "Development, Evaluation and Introduction of a Lane Keeping Assistance System," in *Intelligent Vehicles Symposium*, Parma, 2004, pp. 943–944.

[4] T. Puphal, M. Probst, and J. Eggert, "Probabilistic Uncertainty-Aware Risk Spot Detector for Naturalistic Driving," *Transactions on Intelligent Vehicles*, vol. 4, no. 3, pp. 406–415, 2019.

[5] T. Puphal, M. Probst, Y. Li, Y. Sakamoto, and J. Eggert, "Optimization of Velocity Ramps with Survival Analysis for Intersection Merge-Ins," in *Intelligent Vehicles Symposium*, Changshu, 2018, pp. 1704–1710.

[6] ITS European Congress, "Demonstrations: VI-DAS (Vision Inspired Driver Assistance System)," accessed 17.02.2021. [Online]. Available: https://2019.itsineurope.com/demonstrations/

[7] J. Eggert, D. Aguirre Salazar, T. Puphal, and B. Flade, "Driving Situation Analysis with Relational Local Dynamic Maps," in *International Symposium FAST-zero*, Nara, 2017.

[8] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion Planning in Urban Environments: Part I," in *Conference on Intelligent Robots and Systems*, Nice, 2008, pp. 1063–1069.

[9] F. Diehl, T. Brunner, M. Le, and A. Knoll, "Graph Neural Networks for Modelling Traffic Participant Interaction," in *Intelligent Vehicles Symposium*, Paris, 2019, pp. 695–701.

[10] C. Pek and M. Althoff, "Computationally Efficient Fail-Safe Trajectory Planning for Self-driving Vehicles Using Optimization," in *Intelligent Transportation Systems Conference*, Maui, 2018, pp. 1447–1454.

[11] B. Gassmann, F. Oboril, and et al., "Towards Standardization of AV Safety: C++ Library for Responsibility Sensitive Safety," in *Intelligent Vehicles Symposium*, Paris, 2019, pp. 2265–2271.

[12] M. Ruf, J. R. Ziehn, and et al., "Comparison of Local vs. Global Optimization for Trajectory Planning in Automated Driving," in *Workshop Driver Assistance Systems*, Walting, 2015, pp. 71–83.

[13] H. Watanabe, L. Tobisch, and et al., "A Method for the Estimation of Coexisting Risk-Inducing Factors in Traffic Scenarios," in *Intelligent Vehicles Symposium*, Paris, 2019, pp. 2243–2250.

[14] M. Treiber, A. Hennecke, and D. Helbing, "Congested Traffic States in Empirical Observations and Microscopic Simulations," *Physical Review E*, vol. 62, pp. 1805–1824, 2000.

[15] A. Kesting, M. Treiber, and D. Helbing, "General Lane-Changing Model MOBIL for Car-Following Models," *Transportation Research Record Journal*, vol. 1999, pp. 86–94, 2007.

[16] C. Zhang, J. Zhu, and et al., "Spatiotemporal Learning of Multivehicle Interaction Patterns in Lane-Change Scenarios," in *arXiv*, 2020.

[17] H. Guo and et al., "Safe Path Planning with Gaussian Process Regulated Risk Map," in *Conference on Intelligent Robots and Systems*, Macau, 2019, pp. 6268 – 6275.

[18] OpenStreetMap Contributors, accessed: 2020-08-27. [Online]. Available: www.openstreetmap.org

[19] B. Flade, A. Koppert, G. Velez, A. Das, D. Betaille, G. Dubbelman, O. Otaegui, and J. Eggert, "Vision-enhanced low-cost localization in crowdsourced maps," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 3, pp. 70–80, 2020.

[20] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," in *arXiv*, 2018.

[21] T. Toledo and D. Zohar, "Modeling Duration of Lane Changes," *Transportation Research Record*, vol. 1999, pp. 71–78, 2007.

[22] T. H. Weisswange, S. Rebhan, and et al., "Intelligent Traffic Flow Assist: Optimized Highway Driving Using Conditional Behavior Prediction," *Intelligent Transportation Systems Magazine (Early Access)*, 2019.

[23] J. Goenetxea and et al., "Efficient Monocular Point-of-Gaze Estimation on Multiple Screens for Driver Behaviour Analysis," in *Driver Distraction and Inattention Conference*, Gothenburg, 2018, pp. 118–125.

©2021 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.
DOI: 10.1109/TIV.2021.3091188



Tim Puphal studied mechatronic systems and received his bachelor's and master's degree from TU Darmstadt, Germany. From 2016 to 2021, he was an external PhD student at the TU Darmstadt and worked as a scientist at the Honda Research Institute Europe in Offenbach, Germany. During this time, he was involved in several international projects for automated driving technologies, amongst others, the EU project Vision Inspired Driver Assistance System (VI-DAS). Since 2021, he is a senior scientist at the Honda Research Institute. His main research interests include trajectory prediction, stochastic risk models, behavior planning, motion planning as well as human-machine interaction.



Benedict Flade studied simulation and control of mechatronic systems and received his master's degree from TU Darmstadt, Germany. Since 2016, he works as a research scientist at the Honda Research Institute Europe in Offenbach, Germany. He has been involved in several international projects for automated driving technologies, including the project Vision Inspired Driver Assistance System (VI-DAS) financed by the European Commission. His research interests cover the fields of digital cartography, vehicle localization, computer vision, sensor fusion and risk estimation.



Malte Probst studied Computer Science and Business Adminstration at the University of Mannheim, Germany. He worked as a technical consultant for Hewlett-Packard. He received his Ph.D. from the University of Mainz, Germany, where he was working on using Machine Learning methods within Evolutionary Computation for solving combinatorial optimization problems. At the Honda Research Institute, Malte's main research interests are representations for learning systems as well as generalizable behavior planning in cooperative traffic situations.



Volker Willert received the Dipl.-Ing. degree in electrical engineering and information technology and the Dr.-Ing. degree in control theory and robotics, with a focus on dynamical computer vision, from Technical University of Darmstadt, Germany, in 2002 and 2006, respectively. From 2005 to 2009, he was a Senior Scientist at Honda Research Institute Europe GmbH, and from 2009 to 2020, he has been with the Chair of the Control Methods and Robotics Laboratory, TU Darmstadt. As a lecturer, he lead the research group Machine Vision and Mobile Robotics. Since 2020, he has been a professor at the University of Applied Sciences Würzburg-Schweinfurt, Germany. His main research interests are in the fields of machine intelligence, computer vision, distributed controls, and machine learning for mobile robotics, multiagent systems, and driver assistance systems.



Jürgen Adamy received the Diploma and Dr.-Ing. degrees in electrical engineering from Technical University of Dortmund, Dortmund, Germany, in 1987 and 1991, respectively. Afterwards, he was an Engineer and a Manager in the area of control applications at Siemens, Erlangen, Germany. In 1998, he became a Full Professor at Technical University of Darmstadt, Darmstadt, Germany, where he is the Head of the Control Methods and Robotics Laboratory and was holding various further roles, such as the Director of the Institute of Automatic Control and Mechatronics and Dean of the Department of Electrical Engineering and Information Technology. His research interests are in the areas of nonlinear control, fuzzy systems, and mobile robots.



Julian Eggert received his Ph.D. degree in physics from the Technical University of Munich, Germany, where he was working in the Theoretical Biophysics Department of Prof. J. L. van Hemmen on models of attentional processing in the visual system. In 1999 he joined the Future Technology Research division of Honda R&D (Germany) and in 2003 the Honda Research Institute, where he is currently in the position of a Chief Scientist and leader of projects in the area of artificial cognitive systems with applications in car and robotics domains. His fields of interest are generative models for perception, large-scale models for visual processing and scene analysis, as well as semantic environment models for context-embedded reasoning, situation classification, risk prediction and behavior planning under uncertainty.

©2021 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.
DOI: 10.1109/TIV.2021.3091188