This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Enhancing Pedestrian Safety Through Automated Emergency Steering: Design and Experimental Validation of a Risk-Based Architecture

Leon Tolksdorf, Manuel Muñoz Sánchez, Christian Loeffler, Timm F. Gloger, Yannick Fischer, Robin Smit, and Chris van der Ploeg

***Abstract*—Designing automated vehicles (AVs) to improve pedestrian safety has received increasing attention. Although many algorithms have been proposed for various AV functions that concern pedestrian safety, combining them in a functional architecture and real-world deployment of such an architecture is an open challenge. We review the most frequent car-to-pedestrian collisions from accidentology data and design our AV to avoid these accidents. Our AV employs a model-predictive control-based path planner that plans reference curves and a higher frequency trajectory generator that plans time-critical maneuvers. Furthermore, our proposed AV design tackles real-world challenges, such as noisy perception and late detection of pedestrians due to physical obstruction. These phenomena can possibly result in uncertain pedestrian detection and inaccurate tracking, for which we propose an uncertainty-aware pedestrian prediction model. Uncertainties along pedestrians' predictions are considered by a probabilistic risk measure, selecting minimum risk maneuvers by operating in closed-loop with the trajectory generator. We assess our system's performance by comparing some algorithms against literature counterparts and evaluate the overall system in real-world testing. We find that the computational performance of the proposed algorithms is sufficient for time-critical scenarios. In addition, our system exhibits safety benefits by considering uncertainty in decision-making processes, and our evasive steering system introduces additional avoidance potential to emergency braking.**

***Index Terms*—Pedestrian safety, pedestrian trajectory prediction, risk assessment, autonomous emergency steering.**

Received 18 July 2024; revised 8 May 2025 and 19 September 2025; accepted 13 March 2026. This work was supported by the EU Horizon 2020 Research and Development Program Project SAFE-UP (Proactive SAFEty Systems and Tools for a Constantly UPGrading Road Environment) under Grant 861570. The Associate Editor for this article was S. Hamdar. *(Corresponding author: Leon Tolksdorf.)*

Leon Tolksdorf is with the CARISSMA Institute of Safety in Future Mobility, Technische Hochschule Ingolstadt, 8504 Ingolstadt, Germany, and also with the Mechanical Engineering Department, Dynamics and Control Group, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands (e-mail: leon.tolksdorf@thi.de).

Manuel Muñoz Sánchez is with the Mechanical Engineering Department, Robotics Group, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands (e-mail: m.munoz.sanchez@tue.nl).

Christian Loeffler, Timm F. Gloger, and Yannick Fischer are with the Cross-Domain Computing Solutions, Advanced Engineering, Robert Bosch GmbH, 74232 Abstatt, Germany (e-mail: Christian.Loeffler@de.bosch.com; Timm.Gloger@de.bosch.com; Yannick.Fischer@de.bosch.com).

Robin Smit is with the Integrated Vehicle Safety Group, Netherlands Organisation for Applied Scientific Research, 5700 AT Helmond, The Netherlands (e-mail: robin.smit@tno.nl).

Chris van der Ploeg is with the Integrated Vehicle Safety Group, Netherlands Organisation for Applied Scientific Research, 5700 AT Helmond, The Netherlands, and also with the Mechanical Engineering Department, Dynamics and Control Group, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands (e-mail: c.j.v.d.ploeg@tue.nl).

Digital Object Identifier 10.1109/TITS.2026.3676623

I. Introduction

THE introduction of automated vehicles (AVs) has the potential to improve overall road traffic safety [1]. Without such an improvement, neither the authorities nor society are likely to accept this technology. Accidentologies show that vulnerable road users (VRUs), e.g., pedestrians and cyclists, are among the most endangered traffic participants in urban areas [2]. Therefore, increasing the safety not only of passengers in an AV but also of VRUs is of greater interest.

Designing an AV emphasizing the safety of VRUs requires identifying the scenarios in which VRUs are most frequently involved in critical accidents. Data from various European accident databases [3], [4] are combined in [2] to understand accidents involving VRUs. While critical scenarios are identified, technologies to specifically prevent these accidents are neither proposed nor deployed. Although extensive research has focused on individual algorithms within an automated vehicle architecture [5], most proposed techniques are tested stand-alone in simulation using deterministic input data without computational constraints. As a result, there is a lack of literature bridging the gap between individual algorithms and real-world deployment within a functional AV architecture with the objective of avoiding specific real-world accidents.

We focus on pedestrians as a VRU type, which requires an AV architecture perceiving the pedestrian with sensors, utilizing that information to estimate its state, and predicting the pedestrian's future positions (i.e., a trajectory). While many methods for pedestrian perception and prediction of their trajectories exist [6], [7], [8], most proposed methods assume a high level of accurate and readily available information about the pedestrian and the environment. When this information is unavailable or inaccurate, their performance can degrade drastically [9]. In real-world driving, the process of measuring and predicting introduces uncertainties from sensors and modeling [10]. In addition to that, the available information is often severely limited due to insufficiencies in modeling real-world phenomena, such as adverse weather conditions [11], [12] or sight obstructions. Especially the presence of sight obstructions, as often found in complex urban environments, limits observation time and, hence, the available information about the pedestrian. Other approaches [13] deal with obstructions assuming communication between road traffic actors, however, if no communication is available, backup functionality is still required. Limited observation time can lead to so-called late detections of pedestrians. That is, if a pedestrian and the AV are on a collision course, immediate intervention by the AV is required to avoid collisions. However, limited information, i.e., high uncertainty about the existence, current state, and previous states, can lead to inaccurate trajectory predictions of the pedestrian and, hence, wrong decisions on interventions. To address this challenge and account for the lack of information, some authors suggest capturing the uncertainty within the vehicle architecture [10], [14]. Capturing uncertainties in road users' motion predictions can be done using a risk measure, which is understood as a probabilistic quantity that describes potential harm due to limited knowledge of current and future driving and environment states. In the face of real-world deployment, safety standardization [15] requires the absence of unreasonable risk caused by performance insufficiency, e.g., uncertainty due to sensor and model limitations, specifying risk assessment as a pivotal procedure in the safety certification of AVs [16]. Recent literature suggests that embedding risk in the motion planning and decision-making process results in safer and, in various scenarios, human-like vehicle behavior [17], [18], [19]. Moreover, minimizing harm (represented within a risk measure) in the presence of unavoidable collisions is always preferable [20]. Therefore, we identify the need to utilize risk within the decision-making process to account for uncertainties in modeling, measurement, and prediction to generate safer and more desirable vehicle behavior.

Consequently, we derive the most frequent car-to-pedestrian accident scenarios and design an AV architecture capable of additional accident avoidance potential in these scenarios. Furthermore, we propose an AV design considering the measurement and modeling uncertainties in perception and prediction by a probabilistic risk measure and implement emergency maneuvering to avoid accidents with late detected pedestrians. Our presented AV architecture showcases its functionality by a real-world deployment. As a result, we make the following contributions:

(i) an AV architecture accounting for (possible) late detections of pedestrians combining path planning with higher frequency emergency trajectory generation, extending the collision avoidance potential of accident avoidance systems and switching collision constellations from frontal to side collisions given unavoidable car-to-pedestrian accidents,
(ii) a prediction model with increased robustness against perception noise and tracking uncertainty,
(iii) a probabilistic risk-based decision-making that assesses prediction uncertainties, exhibiting safety benefits under uncertainty when compared to a deterministic risk measure,
(iv) bridging the gap between simulation of individual algorithms and real-world application by deploying a computationally efficient architecture that operates in time-critical scenarios.

The remainder of this article is structured as follows. Sec. II gives preliminaries, derives relevant accident scenarios, and states the research problem along with our proposed approach. Sec. III details the hardware and software architecture of our AV. Sec. IV elaborates on all relevant algorithms. The evaluation methodology and the results of the individual modules and the overall system are given in Sec. V. Finally, Sec. VI and VII discuss the results, present opportunities for future research, and conclude the manuscript.

II. Problem Statement and Proposed Approach

In the following, we state notation and assumptions before reviewing accidentology and identifying frequently occurring scenarios for car-to-pedestrian collisions. Given the scenarios, we state our objective and proposed approach.

A. Preliminaries

The AV, considered from an ego-perspective, and the pedestrian are characterized by a state $\boldsymbol{x} := (\boldsymbol{q}, \theta, v, \omega) \in \mathcal{X}$. The state space $\mathcal{X}$ is composed of actor positions $\boldsymbol{q} := (c_1, c_2) \in \mathbb{R}^2$, heading angles $\theta \in [0, 2\pi)$, velocities $v \in \mathbb{R}$ directed along the heading axis, and turn rates $\omega \in \mathbb{R}$. A set of integers $\{a, a+1, \ldots, b\}$, with $a < b$, is denoted by $\mathbb{Z}_a^b$, the same notation is applied to the reals, i.e., $\mathbb{R}_c^d$, with $c < d$, and the set of positive reals including zero is denoted by $\mathbb{R}_{\geq 0}$. We denote a state at discrete time-step k as $\boldsymbol{x}_k$ and a predicted state at time-step $n > k$ given information available at time-step k as $\boldsymbol{x}_{n|k}$. The same notation is applied for inputs and constraints. Finally, all variables associated with the ego vehicle and the pedestrian will be identified by e and o subscripts respectively. In the sequel, the following definitions are needed.

Definition 2.1 (Path): A path $\mathcal{P}$ is the image of $f_P : \mathbb{Z}_{\lambda_0}^{\lambda_g} \to \mathcal{C}$ enforcing non-holonomic vehicle constraints on configurations $(\boldsymbol{q}, \theta)$ in the configuration space $\mathcal{C} = \mathbb{R}^2 \times [0, 2\pi)$, such that: $\mathcal{P} := \{(\boldsymbol{q}, \theta) \in \mathcal{C} \mid \lambda \in \mathbb{Z}_{\lambda_0}^{\lambda_g} \mapsto (\boldsymbol{q}, \theta) = f_P(\lambda)\}$.

Definition 2.2 (Vehicle Trajectory): A vehicle trajectory $\mathcal{T}$ is the image of $f_T : \mathbb{Z}_k^{k+N_t} \to \mathcal{X}$, parameterizing states by time and enforcing non-holonomic vehicle constraints on states such that: $\mathcal{T} := \{\boldsymbol{x}_{n|k} \in \mathcal{X} \mid n \in \mathbb{Z}_k^{k+N_t} \mapsto \boldsymbol{x}_{n|k} = f_T(n)\}$.

In general, the uncertainties associated with the future positions and headings of the ego vehicle are less significant than those of the pedestrian. Thus, we only consider uncertainty within the measurement and estimations of the pedestrian. As safety is of priority for real-world deployment, we impose the following conditions on our system, further specifying the functionality.

Condition 2.1 (In-lane AES): Autonomous emergency steering (AES) functionalities are aligned with regulation [21], constraining the system to perform only in-lane AES maneuvers.

Condition 2.2 (Human controllability): The yaw rate is constrained, as it is the most correlating quantity for humans to retain control of a vehicle [22].

Condition 2.2 is imposed such that, in case of a system malfunction, a human driver should be capable of taking over the driving task.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

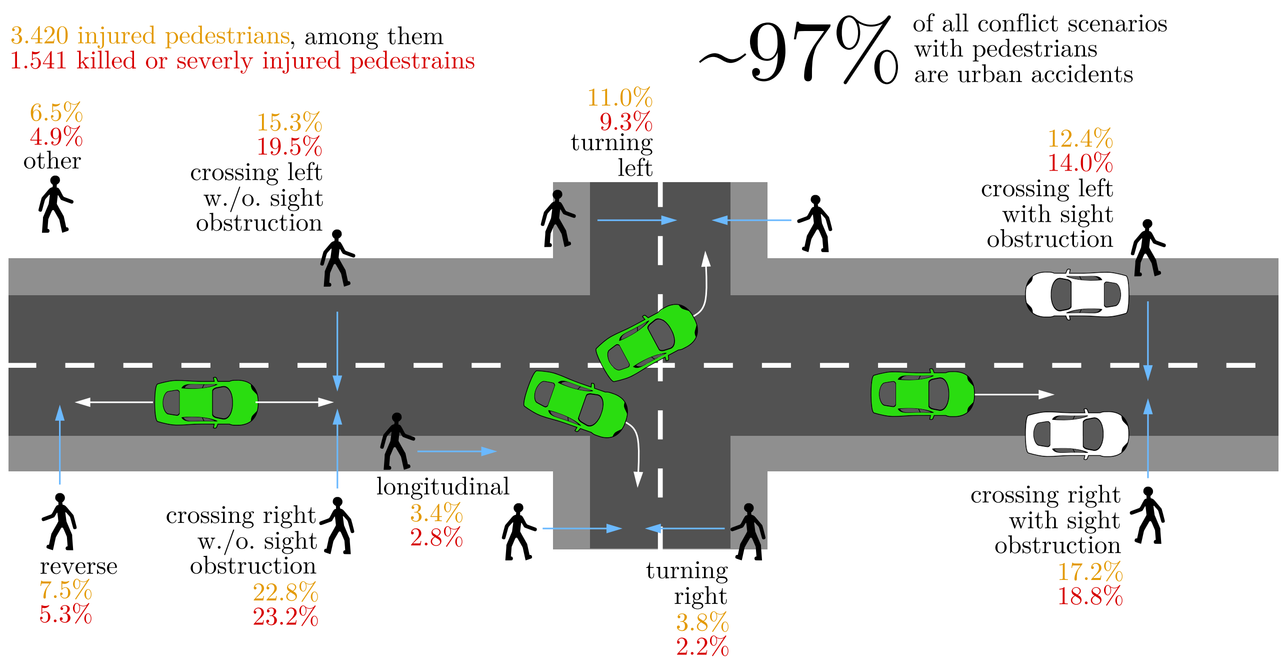


Fig. 1. Fact sheet accidentology, car-to-pedestrian collisions. The green cars collide with the pedestrians and the white car depicts a physical sight obstruction. Adapted on the basis of [2].

TABLE I
MOST FREQUENT ACCIDENT SCENARIOS

Number	Pedestrian direction	Sight obstruction	Velocity range	KSI
(I)	crossing right	without	[26, 48] $\mathrm{km\,h^{-1}}$	23.2%
(II)	crossing left	without	[30, 50] $\mathrm{km\,h^{-1}}$	19.5%
(III)	crossing right	with	[26, 45] $\mathrm{km\,h^{-1}}$	18.7%
(IV)	crossing left	with	[28, 45] $\mathrm{km\,h^{-1}}$	14.0%

B. Scenarios

The accidentology study in [2] identified frequently occurring car-to-pedestrian collision scenarios. Here, accident data from the GIDAS (acronym for German in-depth accident study) database [3] is analyzed. An in-depth analysis is carried out by [2], and the results are depicted in Fig. 1. The top four (by highest occurring killed or severely injured (KSI) pedestrians first) scenarios[1] are further elaborated in Tab. I. In total, the scenarios in Tab. I account for 75.4 % out of the 1541 KSI pedestrians recorded by [2].

Objective: Given scenarios (I) - (IV) from Tab. I, design an AV that maneuvers through the scenarios tracking a trajectory given a reference path and velocity and, if possible, avoids collisions or reduces severity, while satisfying Conditions 2.1 and 2.2.

C. Proposed Approach

The proposed AV is designed to navigate through the scenarios in Tab. I in two modes; first, a nominal mode which applies to all times where no collision is detected, and second, a collision mode in case a collision is detected. The characteristics of each mode are described in the sequel.

1) *Nominal driving mode:* In the nominal driving mode, the vehicle is performing lateral control, i.e., steering, of a reference trajectory following the reference path. Longitudinal control, i.e., gas and brake, is performed by a human driver. The reference path leads towards a desired goal location while following the road network.
2) *Collision mode:* If a collision in the nominal driving mode is detected, the AV plans AES and autonomous emergency braking (AEB) maneuvers. Of all possible maneuvers (including continuing in the nominal mode), the AV selects the maneuver that minimizes the estimated risk along the trajectory. If an AEB maneuver is selected, the AV performs longitudinal control. In contrast, if an AES maneuver is selected, the AV performs lateral control.

III. ARCHITECTURE

To navigate through the scenarios in the respective driving modes, we propose the following vehicle hardware and software architecture.

A. Hardware Architecture

The hardware architecture of the AV is documented in [23]. Summarizing, the AV is equipped with radar and camera sensors for environment perception and a global navigation satellite system with real-time kinematics (GNSS-RTK) module to collecting positional data for the validation process.

B. Software Architecture

The software architecture is displayed in Fig. 2. The block *Sensor input* represents the processing of sensed data, i.e., HD

[1]The pedestrian velocities are not further specified in the data.

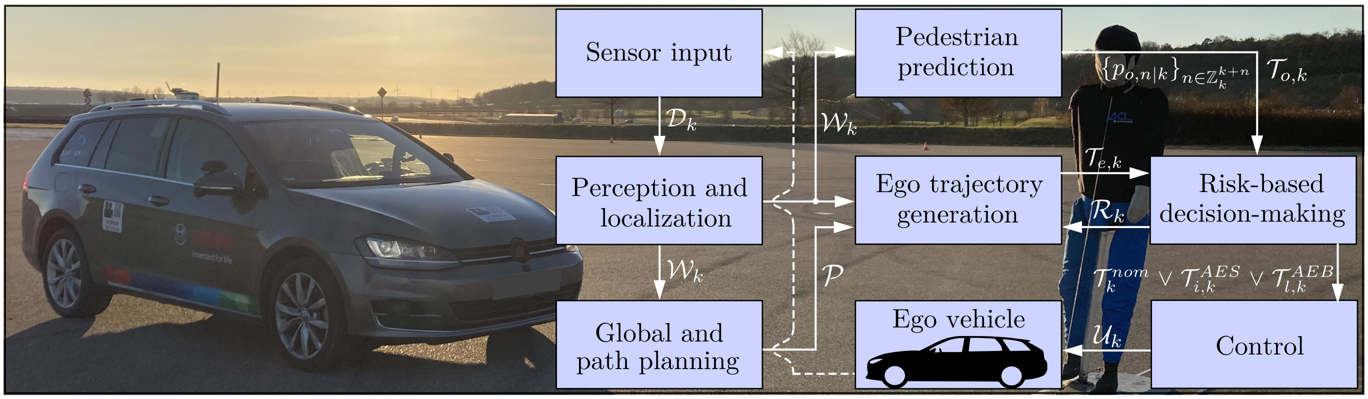


Fig. 2. Test vehicle and test setup (left and background) and software architecture (right). The dashed arrow (right depiction) represents the feedback through interaction with the environment.

images, 3D radar points, IMU data, and provides the data in $\mathcal{D}_k$ at time k for the *Perception and localization* block. This block fuses the sensor data, thereby determining the state of the AV $\boldsymbol{x}_{e,k}$. Also, it detects, classifies, and tracks objects over time. If a pedestrian is detected, a state $\boldsymbol{x}_{o,k}$ is appended to the corresponding state list. Additionally, the road is perceived (i.e., the position of the lane markings) and all information about the road and the states of the detected objects are compiled in $\mathcal{W}_k$ at time k. The block *Global and path planning* derives a reference path $\mathcal{P}$ that follows the lane center based on a set of waypoints leading to a goal destination. Based on the path information $\mathcal{P}$ and environment information $\mathcal{W}_k$, the *Ego trajectory generation* block derives (in the nominal mode) a comfortable ego vehicle nominal trajectory $\mathcal{T}_k^{nom}$, for the AV to track. If a pedestrian is detected (i.e., a pedestrian state $\boldsymbol{x}_{o,k}$ is available), the *Pedestrian prediction* block predicts the pedestrian's future trajectory[2] $\mathcal{T}_{o,k}$ based on the pedestrian's current and past information accompanied by a probability density function of the pedestrian's configuration $(\boldsymbol{q}_o, \theta_o)$ at each time-step along the trajectory, i.e., $\{p_{o,n|k}\}_{n\in\mathbb{Z}_k^{k+n}}$. In case a pedestrian trajectory is available, the *Risk-based decision-making* block checks whether the nominal AV trajectory $\mathcal{T}_k^{nom}$ and the pedestrian's trajectory $\mathcal{T}_{o,k}$ would lead to a collision. If a collision is detected, the *Ego trajectory generation* block is tasked with sampling a set of N_I AES and N_L AEB ego vehicle trajectories contained in $\mathcal{T}_k^{AES} = \{\mathcal{T}_{i,k}^{AES}\}_{i\in N_I}, \mathcal{T}_k^{AEB} = \{\mathcal{T}_{l,k}^{AEB}\}_{l\in N_L}$ respectively. The collection of all ego vehicle trajectories at time k given by $\mathcal{T}_{e,k} = \{\mathcal{T}_k^{nom}, \mathcal{T}_k^{AES}, \mathcal{T}_k^{AEB}\}$ is returned to the *Risk-based decision-making* block. For every ego vehicle trajectory, the block *Risk-based decision-making* is providing information within $\mathcal{R}_k$ containing a risk value for the trajectory, whether a collision is detected and if so, the position and time of the collision on the trajectory. The trajectory with the least risk is selected, such that $\mathcal{T}_k^{nom} \vee \mathcal{T}_{i,k}^{AES} \vee \mathcal{T}_{l,k}^{AEB}$ is forwarded to the *Control* block. This block is tasked with controlling the actuation of the vehicle to follow the selected trajectory. Thereby, a sequence of feed-forward control actions $\mathcal{U}_k$ is derived and actuated on the vehicle.

[2]That is a trajectory in the sense of Definition 2.2, however, it does not satisfy non-holonomic vehicle constraints.

IV. Technical Details

This section details the relevant algorithms in our architecture (Fig. 2) for our contributions.

A. Global and Path Planning

A global planning module is required for the local planner to know roughly which direction to take on the public road. The objectives of this paper are scoped to show the efficacy of our architecture in late-detection scenarios for pedestrians. Therefore, we keep the global planning strategy elementary while we emphasize deriving feasible paths of that global plan.

1) Global Planning: We assume that the vehicle is supposed to stay within its lane center and that the planning reference given to the local planner contains a configuration, i.e., $\boldsymbol{r}_N = (\boldsymbol{q}_e, \theta_e)$, representing the lane center and where N denotes the prediction horizon of the local planner. Note, that miscellaneous works in literature have provided efficient strategies for global navigation of a vehicle [24].

2) Path Planning: The main functionality of the path planner within the scope of our architecture consists of generating a dynamically feasible path that is planned in the lane of interest. The planner used within this architecture is derived from the work in [25], which uses a model-predictive planner and derives risk-averse trajectories using artificial potential fields. As opposed to a local planner, the goal is not to follow a certain trajectory, defined at each time-step, but to follow a discrete set of points generated by the global planning strategy (Sec. IV-A.1). These points mainly assist in route planning for making general decisions (e.g., turning left or right at an intersection). The feasibility of the path is governed through the use of a kinematic bicycle model presented in [26, Eq. (1)], for which we collect the states in $\boldsymbol{x}_k \in \mathcal{X} \subset \mathbb{R}^4$ and the inputs in $\boldsymbol{u}_{P,k} \in \mathcal{U}_P \subset \mathbb{R}^2$, where $\mathcal{X}, \mathcal{U}_P$ are bounded sets representing the state and input constraints, respectively. Such allows us to ensure the boundedness of the model-predictive program, as well as imposing, e.g., comfort-driven constraints. On the public road, a driver and his vehicle are always expected to follow the lanes and comply with the regulations. For this purpose, many researchers have augmented the problem with a potential field [17], [25], [26]. This field analytically captures the level of safety of driving at certain positions by exhibiting a low potential in safe areas (e.g., lane centers) and a high potential in unsafe areas (e.g., in-between lanes). The construction of the potential field is further elaborated in [25, Eq. 5], and results in the mapping $\mathcal{A}(\boldsymbol{x}_k)$. The model-predictive formulation of the path planner is formulated as follows.

$$V^{MPC}(\boldsymbol{x}_k, \boldsymbol{r}_k) := \min_{\boldsymbol{u}_P, \boldsymbol{x}} \sum_{n=k+1}^{k+N_p-1} \ell_{\boldsymbol{x}}\left(\boldsymbol{x}_{n|k} - \boldsymbol{r}_{n|k}\right)$$
$$+ \sum_{k=n+1}^{k+N_p-1} \left[\ell_{\boldsymbol{u}}\left(\boldsymbol{u}_{n|k}\right) + \mathcal{A}(\boldsymbol{x}_{n|k})\right] + m(\boldsymbol{x}_{k+N_p} - \boldsymbol{r}_{k+N_p}), \tag{1a}$$
$$\text{s.t.} \quad \boldsymbol{x}_{k|k} = \boldsymbol{x}_k \tag{1b}$$
$$\left.\begin{aligned} \boldsymbol{x}_{n+1|k} &= f(\boldsymbol{x}_{n|k}, \boldsymbol{u}_{P,n|k}), \\ \boldsymbol{x}_{n|k} &\in \mathcal{X}, \\ \boldsymbol{u}_{P,n|k} &\in \mathcal{U}_P, \end{aligned}\right\} \forall n \in \mathbb{Z}_k^{k+N_p-1}, \tag{1c}$$

where, in (1a), the stage cost $\ell_{\boldsymbol{x}}(\cdot)$, and the input cost $\ell_{\boldsymbol{u}}(\cdot)$ quadratically penalize the control error $\boldsymbol{x}_k - \boldsymbol{r}_k$, and the input $\boldsymbol{u}_{P,k}$, respectively. Furthermore, the term $m(\cdot)$ quadratically penalizes the terminal state at time $k + N_p$. The state $\boldsymbol{x}_k$ in (1b) represents the state of the vehicle at this time. The function f in (1c) represents the non-linear kinematic bicycle model dynamics of [26, Eq. (1)]. Finally, the constraints in (1c) enforce the boundedness of the state $\boldsymbol{x}$ and the input $\boldsymbol{u}_P$. The assumption is that the planner targets a singular goal point set at a constant longitudinal distance ahead of the vehicle. This approach is similar to the concept of pure pursuit control [27], except that potential fields are utilized to shape the remaining path of the vehicle towards this target point. This allows the planner greater flexibility by keeping specific decisions, such as choosing which lane to drive in, open-ended. As a result, the stage cost $\ell_{\boldsymbol{x}}$ is used solely to track the desired velocity. However, the terminal cost is used and is compared with the goal of the global planner, in this case $\boldsymbol{r}_{k+N_p}$. Note that even though the planner can accommodate a trajectory (position, heading, and velocity, mapped in time), only the sequence of configurations $\{(\boldsymbol{q}_n, \theta_n)\}_{n \in \mathbb{Z}_{k+1}^{k+N_p}}$ (i.e., the path) is extracted and used by the trajectory generator in Sec IV-C. Furthermore, it is worth noting that this planner can be extended to the works in [25] and [26] not only to provide a path that tracks the middle of the lane, but can also incorporate static and dynamic objects. This allows us to maneuver around objects before ending up in a safety-critical scenario, a functionality that would be vital for levels of automation beyond the scope of the is work.

B. Pedestrian Trajectory Prediction

State-of-the-art trajectory predictors are typically trained on large, high-quality driving datasets [28]. Such models assume readily available, accurate sensor data, an assumption violated by common low-fidelity in-vehicle setups, leading to performance drops in real deployments [9]. To bridge this gap, we propose a hybrid architecture: one branch optimized for high-fidelity inputs, the other for robustness under low-fidelity conditions.

1) Hybrid Model Architecture: Fig. 3 summarizes our pipeline. If fewer than three past observations of a pedestrian are available, we fall back to a constant-velocity (CV) model,

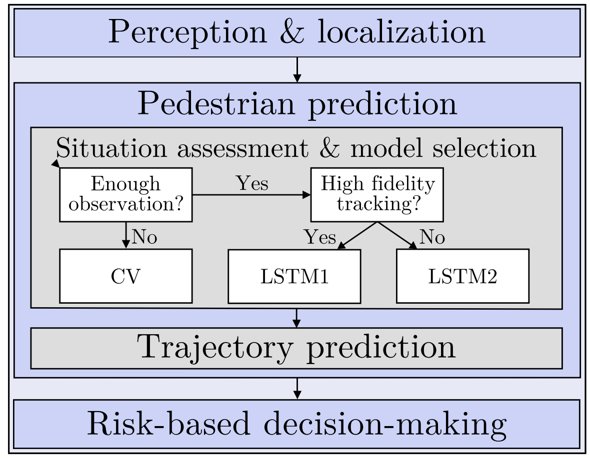


Fig. 3. Overview of the pedestrian prediction module.

which despite its simplicity has been shown to provide competitive results since it describes well average pedestrian walking behavior [29]. Otherwise, we select between two LSTM encoder–decoder variants based on tracking fidelity. LSTM1 follows standard design [30]; LSTM2 incorporates modifications such as noise-aware training and reduced reliance on rotation to improve robustness to unreliable inputs.

2) Architecture Optimization: We perform grid search over LSTM layer counts and neuron numbers, evaluating each configuration by Huber loss on validation data [31]. The chosen model balances low loss with minimal complexity. Further details are in [30].

3) Increasing Robustness: To emulate real in-vehicle sensor shortcomings, we inject artificial noise during training, including sporadic heavy noise on position and heading and late detections by randomly truncating observation sequences [9]. Additionally, we disable coordinate-frame rotation to avoid amplifying heading errors. Tracking fidelity is monitored online via simple heuristics, e.g., abrupt heading flips or implausible acceleration spikes, triggering use of the noise-trained LSTM2 when thresholds are exceeded.

C. Ego Trajectory Generation

The ego vehicle trajectory generation utilizes the information provided by the path planning (Sec. IV-A.2) and pedestrian prediction (Sec. IV-B) modules. It comprises three subsystems: Nominal driving trajectory generation, AES trajectory generation, and AEB trajectory generation. AES and Nominal driving trajectory generation use the same planning method described for the AES case in Sec. IV-C.1. For the nominal driving case (Sec. IV-C.2), we report solely the differences to the AES case, i.e., parameterization, sampling rate, and cyclic replanning. The AEB case is detailed in Sec. IV-C.3.

1) AES Trajectory Generation: The AES trajectory generation is based on the trajectory planning approach presented in [32]. It constitutes a sampling-based trajectory generation algorithm with its basic structure shown in Fig. 4.

The algorithm calculates sets of evasive emergency trajectories using a constrained flatness-based trajectory

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

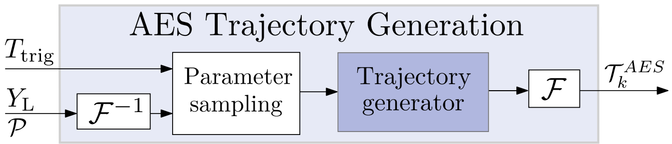


Fig. 4. Schematic of the AES trajectory generation algorithm.

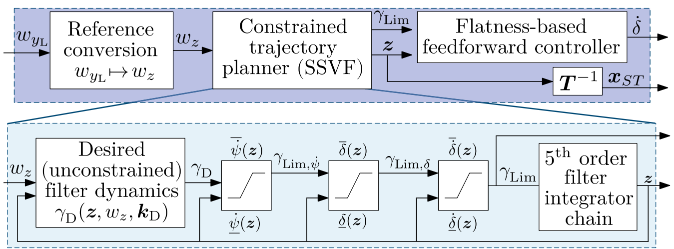


Fig. 5. Schematic of the trajectory generator: A constrained flatness-based trajectory generator using a switched state variable filter (SSVF).

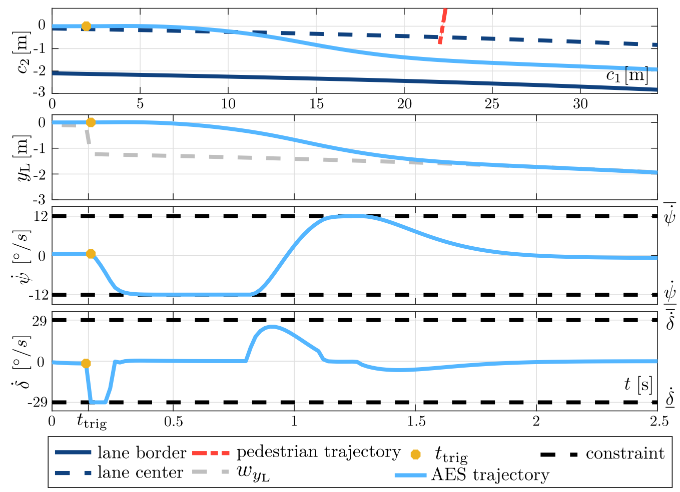


Fig. 6. Exemplary planning of a single AES trajectory including constrained states using the constrained flatness-based trajectory generator.

generator [33], depicted as the embedded box *Trajectory generator* in Fig. 4. Therefore, the target lateral displacements w_{y_L} from the path $\mathcal{P}$, are sampled from the interval of possible lateral displacements Y_L. Similarly, maneuver trigger times t_trig are sampled from the interval of possible trigger times T_trig. The interval of possible lateral displacements Y_L spans from the lane center to the left or right lane border (adjusted by the vehicle width), depending on the direction of the evasive maneuver. The sampling interval T_trig for the maneuver trigger times spans from the time of trajectory generation ($t_\mathrm{trig} = 0$ s) to the predicted time-to-collision when following the vehicle's current trajectory. During runtime, trajectories for each tuple of sampled displacement and trigger time are calculated by the *Trajectory generator*. By using a vehicle model, the calculation of each trajectory yields the time evolution of all vehicle states $\boldsymbol{x}_{n|k}^{ST}$ as well as the corresponding control inputs, i.e., the steering angle rate $\dot{\delta}_{n|k}$, for the planning horizon $\mathbb{Z}_k^{k+N_t}$.

The motion dynamics of the vehicle are modeled by a continuous-time, time-invariant linear single-track model [32] where the vehicle state $\boldsymbol{x}^{ST} = [\beta, \dot{\psi}, \psi, y_\mathrm{L}, \delta]^\intercal$ is composed of the slip angle β, yaw rate $\dot{\psi}$, yaw angle ψ, the lateral displacement y_L, and the steering angle δ. The model input is given by the steering angle rate $\dot{\delta}$. The trajectory generator utilizes the characteristics of differential flatness [34], to generate the constrained AES trajectories, by mapping all given state and input constraints to input constraints of a chain of integrators in a flat state space.

A schematic of the *Trajectory generator* is depicted in Fig. 5. First, the sampled target lateral displacement w_{y_L} is transformed into its representation in the flat state space w_z. Then, w_z is used as a reference for a switched state variable filter (SSVF) calculating smooth trajectories for the system states z as introduced in [33].

The SSVF calculates a desired unconstrained system input

$$\gamma_\mathrm{D} = -\boldsymbol{k}_\mathrm{D}^\intercal \, \boldsymbol{z} + k_{\mathrm{D},1}\, w_z, \tag{2}$$

based on a given state feedback vector $\boldsymbol{k}_\mathrm{D}$. Afterwards this system input is constrained by a cascade of dynamic saturation functions,[3] that restrict the state variables to bounded intervals in order to fulfill Condition 2.2 as well as to account for actuator limitations. Here, Condition 2.2 is represented by a yaw rate limitation $\underline{\dot{\psi}}(z)$ and $\overline{\dot{\psi}}(z)$, for the lower and upper bound, respectively denoted by the lower and upper bar. The actuator limitations are represented by constraints on the steering angle $\underline{\delta}(z)$, $\overline{\delta}(z)$ and the steering angle rate $\underline{\dot{\delta}}(z)$, $\overline{\dot{\delta}}(z)$. The constrained system input γ_Lim is then used to calculate the time evolution of the constraints-satisfying system states z, which in the flat state space is equivalent to propagating the input through a fifth order filter integrator chain. Lastly, the resultant trajectories for the system input and states are transformed back to the original state space using the linear transformation $\boldsymbol{T}^{-1}$ for the states and a flatness-based feedforward controller for the input, respectively. The output of the *Trajectory generator* is a trajectory and input sequence that represents a single AES trajectory for a specific lateral displacement w_{y_L} and maneuver trigger time t_trig.

As the *Trajectory generator* operates in a Frenet frame [35] of $\mathcal{P}$, the trajectories are transformed into a global Cartesian frame by the transformation $\mathcal{F}$, see Fig. 4, prior to further processing by other subsystems. The output of the overall AES trajectory generation is a collection $\mathcal{T}_k^{AES} = \{\mathcal{T}_{i,k}^{AES}\}_{i \in N_I}$ of N_I of sampled AES trajectories.

Fig. 6 shows an illustration of a single exemplary AES trajectory in the global cartesian frame. In the considered example, the vehicle is assumed to follow the lane center (dashed dark blue line) while a pedestrian is crossing the lane (dash-dotted dark red line), leading to a potential collision. The lane is 4 m wide and its border is marked as a solid dark blue line. The lane center serves as the reference curve of the planner's Frenet frame. The vehicle velocity is assumed to be constant at a typical urban driving speed of 13.8 m/s. Only the most relevant signals are depicted in the graph: the sampled trajectory (upper graph) as well as the time evolutions

[3] The saturation functions are dynamic in the sense that their actual bounds depend not only on the state variable bounds itself, but also on the on the current system state z.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

of the planned lateral offset y_{L}, yaw rate $\dot{\psi}$, and the steering angle gradient $\dot{\delta}$ (lower graph). The example demonstrates the planners capability of planning AES trajectories that satisfy the given controllability constraints on the yaw rate $\underline{\dot{\psi}}(z)$, $\overline{\dot{\psi}}(z)$ as well as actuator constraints in terms of the steering angle rate $\underline{\dot{\delta}}(z)$, $\overline{\dot{\delta}}(z)$. We highlight that the simulation result shows effective constraint exploitation as the dynamics are maximized for the evasive maneuver, underpinning the effectiveness of the chosen approach.

The overall algorithm is capable of performing cyclic trajectory replanning during an AES maneuver to adapt the actually driven vehicle trajectory to environmental changes, such as changes in the pedestrian's motion. During the execution of an AES maneuver, the trigger time is fixed to the current time, and only the sampling of target lateral displacements is continued.

2) Nominal Trajectory Generation: When there is no need for emergency maneuvering, nominal trajectories are generated to follow the path planned by the path planning module (see Sec. IV-A.2). The nominal trajectory planner utilizes the constrained flatness-based trajectory generator from Sec. IV-C.1 with a less dynamic parameterization of the SSVF, i.e., different $\boldsymbol{k}_D$, and with a target lateral displacement of $w_{y_{\mathrm{L}}} = 0$ m. Additionally, only a single trajectory per planning cycle is calculated instead of sampling a set of trajectories. A cyclic replanning of nominal trajectories is performed throughout the driving operation, which generates the behavior of a low-frequency closed-loop path-tracking controller.

3) AEB Trajectory Generation: The current nominal trajectory is used as the basis for the AEB trajectory. The point in time at which the AEB shall bring the vehicle into standstill is determined via the predicted time-to-collision on the nominal trajectory. The AEB trajectory is calculated including a given safety distance between the planned vehicles location of standstill and the predicted location of the collision.

D. Risk-Based Decision-Making

A quantitative assessment of the set of possible trajectories is necessary to decide which trajectory to pursue. By design, however, a set of trajectories is only sampled if a collision on a nominal trajectory is detected. Thus, collision checking is required before assessing trajectories with a risk measure.

1) Shape Approximation and Collision Checking: Since the ego vehicle's and pedestrian's predicted trajectories are calculated assuming point masses, we model their footprints with circles, of which we can check for collisions. We utilize a circular shape approximation proposed by [36] for each actor, where the ego vehicle is approximated by three circles and the pedestrian by a single circle. Clearly, checking if two circles collide is computationally efficient as it only requires verifying that the distance between two circle centers is less or equal than the sum of both radii. Provided with the geometric center of the ego vehicle $\boldsymbol{q}_{e,c}$, the circles shall be placed such that the rectangular footprint of the ego vehicle is entirely covered by the circles. Having three circles of equal radii, we assign one to be frontal circle, one middle circle, and one rear circle. The inter-circle distance d_c and the ego vehicle circle radii r_e for a three circle covering can be found in [36]. As a shorthand we denote $\beta \in \{f, m, r\}$ for the front, middle, and rear ego vehicle circle, respectively. This yields the respective inter-circle distances $d_{\beta,k} = \|\boldsymbol{q}_{e,\beta,k} - \boldsymbol{q}_{o,k}\|$ from the centers of the ego vehicle's circles $\boldsymbol{q}_{e,\beta,k}$ to the pedestrian's circle center $\boldsymbol{q}_{o,k}$, at time k. We can check for collisions for each ego circle with the pedestrian's circle, with radius r_o, with the indicator function

$$I(\boldsymbol{q}_{e,\beta,k}, \boldsymbol{q}_{o,k}) = \begin{cases} 1 & \text{if } \|\boldsymbol{q}_{e,\beta,k} - \boldsymbol{q}_{o,k}\| \leq r_e + r_o, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Given an ego vehicle trajectory $\mathcal{T}_k^{type}$, where $type \in \{nom, AES, AEB\}$ and an object trajectory, we can check each time-step for a collision with (3). For clarity, we denote at least one collision among a pedestrian and an ego vehicle trajectory as $C^{type} = 1$ and, contrarily, $C^{type} = 0$ if no collision occurs.

2) Risk Estimation: The specific system design for late detections of pedestrians faces the challenge of inaccurate pedestrian trajectory predictions. Thus, we utilize a probabilistic risk measure, considering the level of uncertainty in the predictions. We use the risk definition of [37]. Herewith, risk is understood as the expected severity of an event, i.e., the event of a collision. The severity of a collision with the pedestrian's circle and an individual ego circle measured by

$$s_\beta(v_{e,k}, v_{o,k}) = W_\beta \frac{1}{2} \left| \left(m_e v_{e,k}^2 - m_o v_{o,k}^2 \right) \right|, \tag{4}$$

where $m_e, m_o \in \mathbb{R}_+$ are the masses of the ego vehicle and pedestrian and $W_\beta \in \{W_f, W_m, W_r\}$ is a non-zero positive weighting factor. The weighting factor allows to assign different severities to the individual ego vehicle circles. In case of an unavoidable collision, that allows to tune the ego vehicle to prefer collisions with any of the circles, since the accident severity also depends on the point of impact [38]. Denote $p_{\boldsymbol{q}_{o,k}}$ the probability density function associated with the pedestrian's position $\boldsymbol{q}_{o,k}$. For any PDF the risk can be estimated by applying Monte Carlo sampling (MCS). Therefore, the indicator function (3) is utilized, with which we approximate the risk (see [37, Eq. 1]) by the law of large numbers, where $\boldsymbol{q}_{o,j,k}$ is one of N_J samples drawn from the density $p_{\boldsymbol{q}_o,k}$. Furthermore, as multiple ego vehicle circles can collide simultaneously with the pedestrian's circle, we select the ego vehicle circle that maximizes the severity. This leads the risk at time k to be estimated as

$$\max_{\beta \in \{f,m,r\}} s_\beta(v_{e,k}, v_{o,k}) \frac{1}{N_J} \sum_{j=1}^{N_J} I(\boldsymbol{q}_{e,\beta,k}, \boldsymbol{q}_{o,j,k}) \approx R_k. \tag{5}$$

For a trajectory $\mathcal{T}_k^{type}$, we obtain the total risk $R_{k,tot}$, by summing the predicted risk $R_{n|k}$ over the prediction horizon $\mathbb{Z}_{k+1}^{k+N_p}$ of the trajectory.

3) Decision-Making: The decision-making process is illustrated in Fig. 7. Here, we distinguish between four maneuver states (depicted as solid line boxes): *Nominal driving*, *Nominal driving (collision predicted)*, *AEB maneuver*, and *AES maneuver*. Within the state of *Nominal driving*, no collision on the nominal trajectory has been predicted, i.e., $C^{nom} = 0$, and thus no emergency trajectory planning is needed. In the state of *nominal driving (collision predicted)*, a collision on the nominal trajectory has been predicted, i.e., $C^{nom} = 1$, (and

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

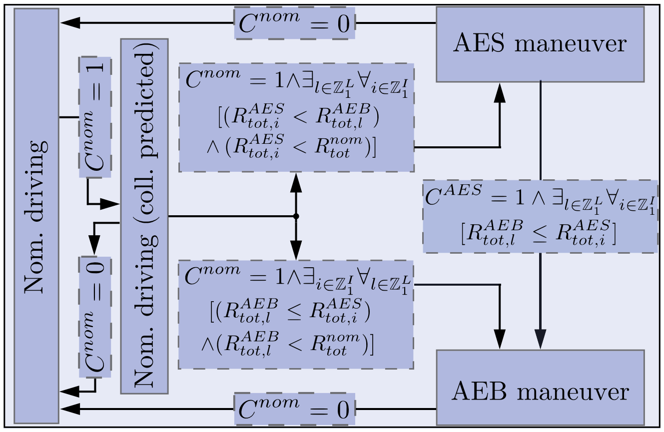


Fig. 7. Decision-making process. The solid line boxes represent driving states and the dashed line boxes depict the conditions for state transition, adapted from [30].

the trajectory planners for the AEB and AES trajectories have been activated) but not all information needed to decide on the scheduling of the following maneuver is available yet. The required information are sets of $\mathcal{T}_k^{AES}$ and $\mathcal{T}_k^{AEB}$ trajectories and their respective risk values $\mathcal{R}_k$. If for each of the N_I AES trajectories, N_L AEB trajectories, and the nominal trajectory a total risk estimate R_{tot}^{type} is present, we select an *AES maneuver* if

$$C^{nom} = 1 \wedge \exists_{i \in \mathbb{Z}_1^{N_I}} \forall_{l \in \mathbb{Z}_1^{N_L}} [(R^{AES}_{tot,i} < R^{AEB}_{tot,l}) \wedge (R^{AES}_{tot,i} < R^{nom}_{tot})].$$

Contrarily, we select an *AEB maneuver* if

$$C^{nom} = 1 \wedge \exists_{l \in \mathbb{Z}_1^{N_L}} \forall_{i \in \mathbb{Z}_1^{N_I}} [(R^{AEB}_{tot,l} \leq R^{AES}_{tot,i}) \wedge (R^{AEB}_{tot,l} < R^{nom}_{tot})].$$

Once in the *AES maneuver* state, the system can transition into the *Nominal driving* state, if no collision is present on the nominal trajectory. Alternatively, it can also transition into the *AEB maneuver* state if, given an updated set of all trajectories, i.,e., $\mathcal{T}_{e,k+1}$, the AES trajectories collide and there is an AEB trajectory with less or equal risk, i.e.,

$$C^{AES} = 1 \wedge \exists_{l \in \mathbb{Z}_1^{N_L}} \forall_{i \in \mathbb{Z}_1^{N_I}} [R^{AEB}_{tot,l} \leq R^{AES}_{tot,i}].$$

Once within the *AEB maneuver* state, the ego vehicle can only transition into the *Nominal driving* state, provided there is no collision on the nominal trajectory and it is in a standstill.

V. Evaluation Methodology and Results

This section is split in two parts: first, we present the evaluation methodology and results of the individual modules in order as they are presented in Sec. IV and, second, we present the experimental evaluation and results of the overall system. Note, that additional results can be found in [39].

A. Path Planning: Evaluation Methodology

Although the path planning module, as described in Sec. IV-A.2, is capable of following various road structures, the objective for the experimental validation is to set the vehicle in a repeatable initial condition at the time of triggering AEB/AES. Therefore, the vehicle must be directed toward the center of the lane, regardless of the initial condition at the beginning of an experiment. We emphasize that the path planner must generate paths that are kinematically feasible for the trajectory generator to track. This may not be achievable solely on the basis of the initial lane center measurement of the sensors, especially if there are significant deviations from the vehicle to this reference point. However, we use the lane reference, obtained from measurement data, as a benchmark in this performance analysis. We quantify the path planner's performance with the following key performance indicators:

a) Lateral deviation between the terminal state coordinates and the lane center coordinates. This quantitatively describes the fact that the path should be long enough to steer the vehicle back to the center of the lane. Further, in the terminal state, the vehicle should be aligned in both its lateral position and orientation with the lane center.

b) Update rate: Experimental validation has shown that the sufficient update rate for replanning a path is 1 Hz. The second KPI is defined as the difference between the desired and the achieved replanning frequency. If the desired update rate is not achieved, the resulting path may be outdated with respect to the current situation, and system performance and repeatability cannot be guaranteed.

B. Path Planning: Results

a) Lateral Deviation: Fig. 8(a) shows all computed paths (red) together with all line observations over the entire recording time (blue), for one of the validation runs. The testing takes place in the center of a three-lane roadway. The left and center lanes, as shown in the top and middle in Fig. 8(a), serve as driving lanes and are each 4 m wide. Conversely, the right lane, shown at the bottom in Fig. 8(a), acts as an emergency lane with a width of 4.5 m. Note that the line observations are not always accurate, due to, among others, partial lane occlusion, faded paint, or observation delays. Further, the missing line measurements in Fig. 8(a) at an x-position of 55 m and y-position of 7 m are explained by the lane processing algorithm, belonging to the used camera sensor, which is only optimised to detect the ego's lane (road markings at a y-position of $\pm$ 2.25 m) and not the lane markers besides the ego lane. It is important to mention that good quality ego-lane readings are sufficient within the framework of this experimental study, as we focus solely on in-lane AES. For this analysis, the lane center estimations from the perception system are considered ground truth. The lateral distance to the center of the closest lane at the end of the calculated path is depicted in Fig. 8(b). At time instance k = 95, the right lane marker is not observed, leading to a temporary absence of the correct lane center and consequently to an increased lateral error of a full lane width. We omitted the resulting outliers in Fig. 8(b), because they were caused by an erroneous perception. The maximum error is approximately 0.2 m and the average error is 0.033 m.

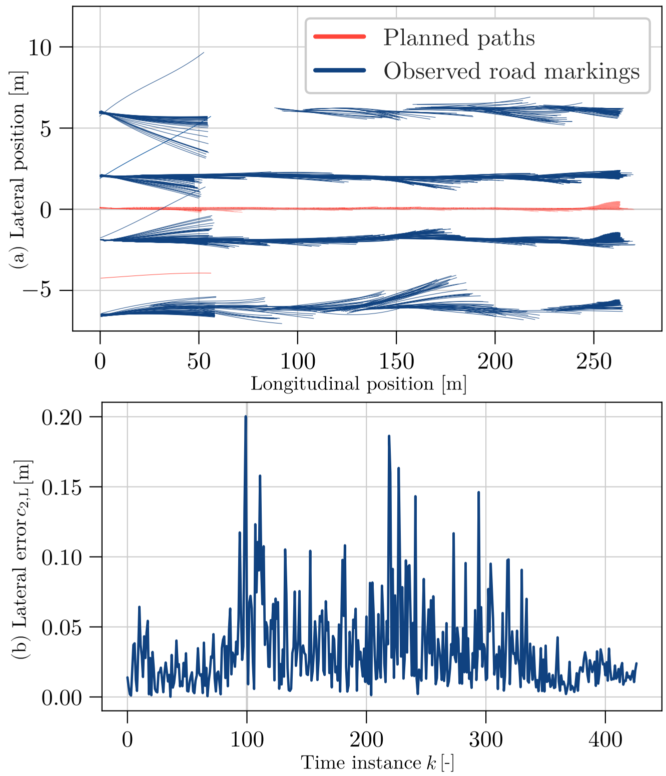


Fig. 8. (a) All lane observations and planned paths from a single validation run. (b) Path planning lateral error with respect to the lane center with omitted outliers at $k = 95$, for a single validation run.

b) Update Rate: The required update rate for the planner is 1 Hz.[4] The mean frequency achieved for 10 test runs is 0.99995 Hz (std. 0.00061 Hz).

Given the context of our application, the lateral deviation is sufficiently small and the update rate is adequate, as the path only accounts for static objects (see Definition 2.1).

C. Pedestrian Prediction: Evaluation Methodology

To evaluate the performance of the developed prediction model on a wide variety of pedestrian trajectories, we use the Waymo Open Motion Dataset (WOMD) [40], a publicly available dataset for this purpose. During training, WOMD's training split and two-thirds of the validation split are used, while one-third of the validation split is reserved for the final evaluation, consisting of 40930 pedestrian trajectories. To assess the robustness of the models, similar levels of noise as measured in-vehicle are simulated and added to the validation data, and the baseline and robustified models are compared. The baseline and robustified models refer to the optimized architectures of LSTM1 and LSTM2, respectively, as described in Sec. IV-B. We report the Euclidean distance between the predicted and true trajectories, commonly known as the final displacement error (FDE) [28], and the standard deviation (std.) of the prediction errors. For a more complete

[4]Note that this is not the maximum update rate of the software, as the planner has been running on 10–15 Hz as well on the same embedded hardware. However, to save computational resources, without hindering the overall performance of the system, the update rate is limited to 1 Hz.

TABLE II

PREDICTION ERRORS (FDE AND STD. IN METERS) OF THE BASE AND ROBUSTIFIED MODELS WITH HEALTY AND NOISY INPUT DATA

Prediction horizon [s]			0.1	1.1	3.1	5.1	7.6
Base model	Healthy	FDE	0.07	0.18	0.6	1.19	2.16
		std.	0.08	0.19	0.66	1.29	2.17
	Noisy	FDE	0.27	1.01	2.63	4.33	6.48
		std.	0.16	0.67	1.91	3.13	4.55
Robustified model	Healthy	FDE	0.18	0.2	0.61	1.19	2.11
		std.	0.06	0.19	0.68	1.34	2.31
	Nosiy	FDE	0.29	0.33	0.78	1.41	2.37
		std.	0.15	0.22	0.69	1.34	2.31

TABLE III

PREDICTION PERFORMANCE DEGRADATION OF THE BASE AND ROBUSTIFIED MODELS WITH NOISY INPUT DATA

Prediction horizon [s]		0.1	1.1	3.1	5.1	7.6
Base model	$\%\Delta_{\text{FDE}}$	285.7	461.1	338.3	263.9	200
	$\%\Delta_{\text{std.}}$	100	252.6	189.4	142.6	109.7
Robustified model	$\%\Delta_{\text{FDE}}$	61.1	65	27.9	18.5	12.3
	$\%\Delta_{\text{std.}}$	150	15.8	1.5	0	0

assessment, these metrics are reported at multiple prediction horizons, ranging from 0.1 to 7.6 seconds.

Additionally, to get an intuition of the performance degradation of a model, we report the relative increase of the model's FDE and std. when using noisy input data. Let $\mathcal{M}$ denote any metric of interest, in our case either FDE or std. Then the relative increase of $\mathcal{M}$, $\%\Delta_{\mathcal{M}}$, when using noisy input data, is given by

$$\%\Delta_{\mathcal{M}} = \frac{\mathcal{M}_{\text{noisy}} - \mathcal{M}_{\text{healthy}}}{\mathcal{M}_{\text{healthy}}} \cdot 100. \tag{6}$$

With this metric, a relative FDE increase of 100% would signify that the model's FDE increased 100% its original value. In other words, when using noisy data, the FDE is twice as much as when using healthy data.

D. VRU Trajectory Prediction: Results

Tab. II and Tab. III present the results of the comparison between the baseline model and the robustifed model. In Tab. II, the FDE and std. of the prediction errors are summarized. Tab. III summarizes the relative performance degradation of the models given by (6). As can be seen in the tables, the base model outperforms the robustified model for short- to medium-prediction horizons when using healthy data. It seems the robustified model learned it cannot fully rely on the accuracy of the perception module and erroneously offsets the predicted object's position. For longer prediction horizons (i.e. 7.6 s), the robustified model outperforms the base model. However, with imperfect data the performance of the base model rapidly degrades, reaching an FDE of up to 6.48 m (+ 200%) and std. of up to 4.55 m (+ 109.7%). However, its greatest performance degradation is seen at shorter prediction horizons of 1.1 s, with an FDE and std. increase of + 461.1% and + 252.6% respectively.

Similarly, the robustified model reaches its greatest performance degradation at short prediction horizons. At 0.1 s, its std. changes +150% and at 1.1 s, its FDE changes +65%, which is significantly lower than the performance degradation of the base model. Additionally, as the prediction horizon increases, the performance degradation is reduced, reaching a change of +12.3% in FDE and no change in std. for the longest horizon of 7.6 s. Summarizing, the robustified model presents a significant improvement over the base model, especially under noisy input data, as found in the presented real-world driving scenarios.

E. Ego Trajectory Generation: Evaluation Methodology

The evaluation of the trajectories generated for the ego vehicle as described in Sec. IV-C is performed in the context of trajectory tracking control. The evaluation assesses the amount of error that needs to be regulated by a trajectory tracking controller. For doing so, the generated trajectories along with the generated control input are treated as a reference. The evaluation uses AES trajectories only, as those exhibit the highest lateral dynamics. Thus, the results should indicate the upper limit to the potential errors that a trajectory tracking controller would be faced with regulating. A dataset of 44 AES maneuvers was collected [39]. The maneuvers were executed at a vehicle speed of 13.8 m/s, the lateral displacement of the evasive maneuver is set to 1 m, and the trajectories are sampled with a sampling time of 20 ms. The AES maneuvers in the dataset are driven without replanning, as such the generated control input is applied feedforward throughout the entire duration of a maneuver (approximately 1500 ms). As higher replanning frequencies are typical in practice, we calculate the errors occurring in a moving horizon with a length of $N = 10$ by the following measures. The planned state is provided by the single-track model from Section IV-C.1, i.e., $\boldsymbol{x}_k^{\mathrm{ST}}$ at time-step k. Denote the measured state vector at time-step k as $\boldsymbol{x}_k^{\mathrm{meas}}$. To estimate the error occurring in a moving horizon, we denote $\Delta\boldsymbol{x}_{n|k} = \Delta\boldsymbol{x}_{n|k}^{\mathrm{meas}} - \Delta\boldsymbol{x}_{n|k}^{\mathrm{ST}}$, with $\Delta\boldsymbol{x}_{n|k}^{\mathrm{meas}} = \boldsymbol{x}_k^{\mathrm{meas}} - \boldsymbol{x}_n^{\mathrm{meas}}$ and $\Delta\boldsymbol{x}_{n|k}^{\mathrm{ST}} = \boldsymbol{x}_k^{\mathrm{ST}} - \boldsymbol{x}_n^{\mathrm{ST}}$, as the error at time-step n with $n \in \mathbb{Z}_k^{k+N}$. Within the horizon, each $\Delta\boldsymbol{x}_{n|k}$ is computed and sorted into bins of a histogram. The horizon is then shifted forward by one time-step, and the errors $\Delta\boldsymbol{x}_{n|k}$ for all instances in the horizon are again computed and sorted into the histogram's bins. Once the horizon is moved over the entire trajectory, the process is repeated for the next trajectory of the dataset. Therefore, the histograms estimate the potential errors a trajectory tracking controller is faced with, given a control cycle of up to 200 ms.

F. Ego Trajectory Generation: Results

The state variables for this evaluation are the lateral displacement y_{L}, the yaw angle ψ, and the steering angle δ, to assess if the vehicle can perform the planned steering behavior. The histograms for these variables are depicted in Fig. 9. The bin sizes are set for an accurate representation of the data. Further, the data is presented in relative frequency by dividing the number of entries in each bin by the total number of entries per time instance. In addition to the estimated relative frequency for the deviations, the overlayed solid lines indicate the medians of the respective states and the dashed lines display the bounds of the 68.28 % confidence intervals ($\mathrm{CI}_{68}(\cdot)$) centered around the medians.

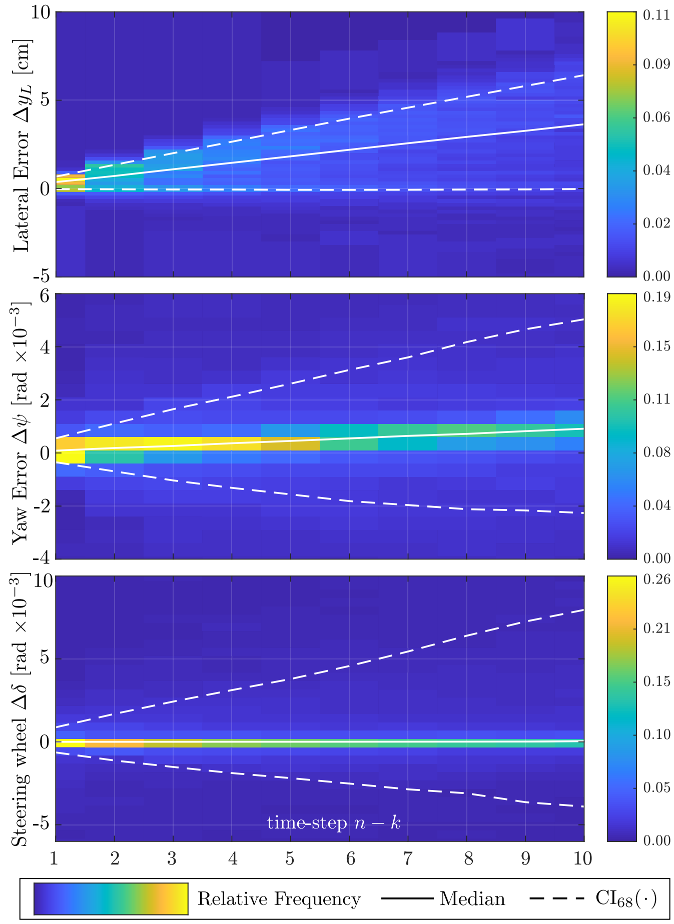


Fig. 9. Time evolution of the deviations from planned to measured AES trajectories within an assumed replanning cycle of 200 ms.

The deviations of the steering angle show a distribution with a median of about 0 for all times and yield a maximum confidence interval of $\mathrm{CI}_{68}(\Delta\delta) = [-3.9, 8] \times 10^{-3}$ rad. The deviations of the vehicle's yaw display an approximately linear increase of the median over the time of the assumed replanning cycle up to about 9×10^{-4} rad ($\mathrm{CI}_{68}(\Delta\psi) \in [-2.3, 5] \times 10^{-3}$ rad). We find a systematic error between the planned and measured trajectory, which is also reflected in the distributions of the lateral deviations. Here, the median increases up to 3.6 cm and the 68 % confidence interval is $\mathrm{CI}_{68}(\Delta \mathrm{y_L} \in [-7 \times 10^{-2}, 6.4]\mathrm{cm})$. The cause of this error is an offset in a sensor reading of the test vehicle that was used to drive the AES maneuvers and has been discussed in [39]. Even including this systematic error, the planned trajectories are estimated to be followed within a lateral accuracy of better than 7 cm using the planned steering angle as a feedforward control input only. We, therefore, conclude that the planned trajectories and control inputs are well suited as a reference for a feedforward trajectory tracking controller.

G. Risk-Based Decision-Making: Evaluation Methodology

The objective of the following analysis is to evaluate the decision-making performance of the proposed algorithm (see

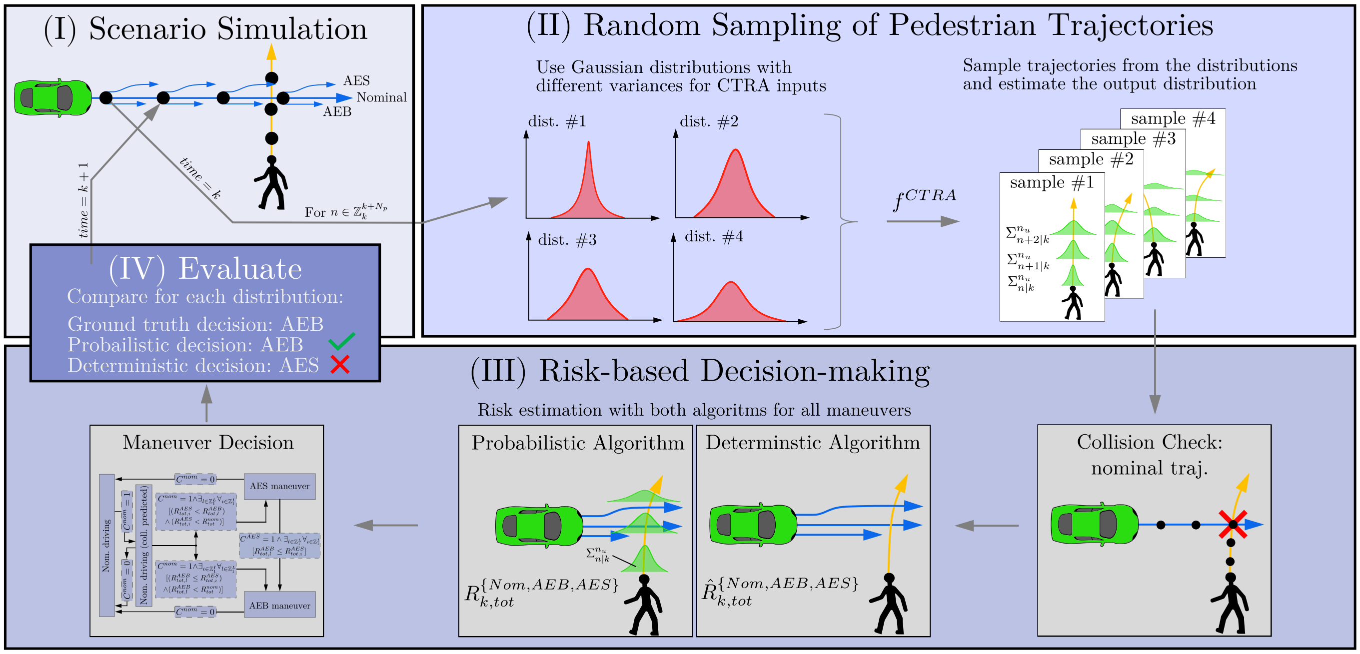


Fig. 10. Simulation workflow to asses the decision-making performance under uncertainty.

Sec. IV-D) under varying levels of uncertainty in the pedestrian prediction stand-alone. To evaluate the performance of the proposed probabilistic risk estimation algorithm, we compare the performance against a deterministic baseline algorithm. The baseline is retrieved from [41], where the risk, denoted $\hat{R}_k$, is described as a potential field depending on the distance to and the severity of the risk-inducing object at time k. Distance-based or time-based (e.g., time-to-collision) risk measures are state-of-the-art in many algorithms in AVs [42]; however, none of them considers uncertainties in predictions and estimations. Note that we used the absolute value of [41, Eq. 2] and applied the same parameterization and severity measure as with our probabilistic approach.

The simulation setup is depicted in Fig. 10. The evaluated scenarios are pedestrian crossings from the right. The ego vehicle is traversing along the nominal trajectory (see Fig. 10 I), and at each discrete time instance, it is tasked with deciding whether to continue on the nominal trajectory or to perform either an AES or an AEB maneuver, hence at each time instance, steps II - IV in Fig. 10 are performed. However, for this simulation, the decision is never executed, and the vehicle continues on the nominal trajectory in the next time instance. Note that we infer the ground truth decision at each time instance from our setup, since the actual pedestrian trajectory is predetermined in the simulation, i.e., walking straight with constant velocity across the road. The ground truth decision is based on the ground truth collision check and if a collision is unavoidable, the maneuver with the least severity (i.e., lowest value of (4)) is chosen as the ground-truth decision.

The second step in the simulation is depicted in Fig. 10 II. Here, the objective is to generate predicted pedestrian trajectories with varying levels of uncertainty within the prediction. To obtain physically plausible trajectories, we employ a constant turn rate and acceleration (CTRA) model for the pedestrian (see, e.g., [43]). The inputs to this model are the turn rate and the acceleration (direct along the heading axis of the pedestrian). To model many different pedestrian behaviors, we choose to draw the pedestrian's inputs from Gaussian distributions and sample trajectories from these inputs for different uncertainty levels.[5] We generate different uncertainty levels by parameterizing Gaussian distributions with identical means but distinct standard deviations. The mean represents the pedestrian's ground truth input. Given that method, we generate a sample set of 100 pedestrian trajectories for each uncertainty level. For each sample set, we quantitatively determine the variance $\Sigma^{n_u}_{n|k}$ in the output, i.e, position and heading, at each time instance of the pedestrian's trajectory. Lastly, we draw a single trajectory from the sample set for each uncertainty level with its associated variances, representing a sample of an uncertain trajectory prediction.

The third step (see Fig. 10 III) performs the decision-making for each uncertainty level. Here, we embed both the probabilistic and deterministic algorithm. With the risk estimate of each algorithm and the collision check for the nominal trajectory, we decide separately with both algorithms according to our decision logic (see Fig. 7), which maneuver option to select.

The fourth and last step (see Fig. 10 IV) then compares the decisions of the probabilistic and deterministic algorithms with the ground truth decision, and thus determines whether an algorithm has made the correct decision. This comparison is done for each uncertainty level. Finally, the ego vehicle is traversed along the nominal trajectory for one time-step, as well as the pedestrian on its predetermined trajectory, and the entire process is repeated for the next instance.

[5]Note that we chose Gaussian distributions, as we wanted the pedestrian's trajectory to be close to the mean for low uncertainty and symmetrically varying around the mean with increasing uncertainty. After all, this remains a modeling choice, and other distributions may also be admissible.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

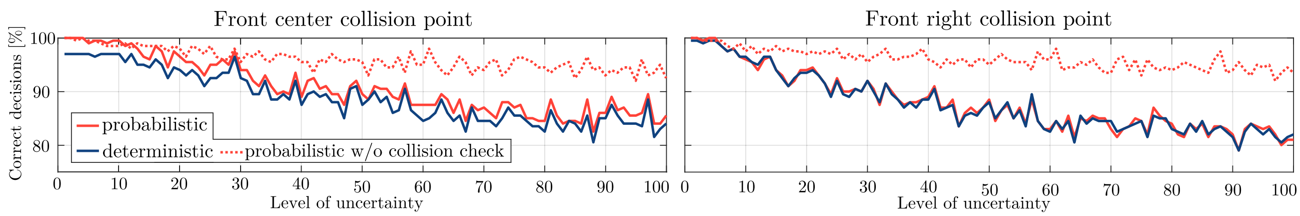


Fig. 11. Results risk-based decision-making, the dotted graph shows a deactivated collision check for the probabilistic algorithm. Each data point represents the percentage value of 200 decisions, i.e., the amount of time-steps in the simulation, which considers 20 s of driving with the ground truth collision at 9.9 s. We linearly grew the standard deviation in each input from 0.001 for uncertainty level 0 to 5 for uncertainty level 100.

H. Risk-Based Decision-Making: Results

The results of the risk-based decision-making module are displayed in Fig. 11. Here, the left plot simulates a scenario where the pedestrian collides in the center of the ego vehicle's front, whereas the right plot simulates a scenario where the pedestrian collides with the right front corner of the vehicle. While the solid lines display the results as discussed in the previous section, the dotted line corresponds to a deactivation of the collision check for the probabilistic algorithm. Although the collision check is embedded before the risk is estimated to reduce system load in real-world operation, we are interested in understanding the potential performance degradation such collision check may introduce. We reason that randomly drawing a pedestrian trajectory can lead to false collision checks, especially when drawing from high uncertainty levels. The deterministic algorithm cannot operate without the collision check, as it would always decide for AEB maneuvers, no matter how distanced the pedestrian is. This is because minimal deterministic risk is achieved when the ego vehicle's velocity is zero, as the distance-based risk never fully vanishes. In contrast, probabilistic risk estimation can operate without a collision check. Deactivating the collision check for the probabilistic algorithm increases the decision-making's robustness, as can be seen in the offset of the dotted graph in both plots. Given the centered collision point, it can be seen that the probabilistic algorithm always outperforms the deterministic algorithm. Generally, the deterministic algorithm tends to falsely decide for AES maneuvers, as those increase the distance to the pedestrian, explaining the difference in the left plot, where the collision is in most cases best avoided by AEB. Also, the same explanation holds for the lowest level of uncertainty, where the deterministic algorithm does not achieve 100 % correct decisions. For the front right collision point, both algorithms decide very similar with the nominal collision check, however, as with the center collision point, the potential of the probabilistic algorithm is vastly improved when the collision check is deactivated. Concluding, the probabilistic algorithm without collision checking is remarkably robust against uncertainty.

I. Experimental Evaluation

In the following, we present our test scenario in which the subsequent run-time measurements are conducted and the overall system is evaluated.[6]

[6]Footage from experimental testing can be seen here: `https://www.youtube.com/watch?v=SjPPo1OMHc8`

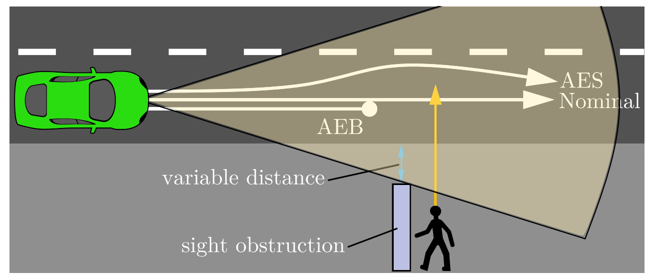


Fig. 12. Test setup, pedestrian crossing with variable sight obstruction.

1) Experimental Test Scenario: Our test scenario is in accordance with Sec. II-B and depicted in Fig. 12. To ensure the best possible repeatability, we employ a pedestrian dummy system. This system operates on a freely moving board, which follows preprogrammed trajectories leading to a predetermined impact point with the test vehicle. Additionally, the dummy system receives live positioning data of the test vehicle. A controller adjusts the dummy's velocity such that minor deviations of the velocity of the oncoming ego vehicle can be compensated for and the impact point remains consistent over multiple test runs. As depicted in Fig. 12, the scenario represents a pedestrian crossing, where a sight obstruction determines the detection time of the pedestrian. The sight obstruction is placed at varying distances, such that early and late detections are tested. For all tests, the nominal ego vehicle velocity is kept at 13.8 m/s and the pedestrian velocity at 1.67 m/s. Two lane widths are tested, i.e., 3.5 m and 4 m.

2) Run-Time Analysis of Risk Estimation and Trajectory Planning: The computational effectiveness of the risk estimation and trajectory planning framework is assessed by the analysis of extensive runtime measurements using the demonstrator vehicle. The analysis presented here is based on [32] and [39]. The data consists of 86 AES scenario test drives, resulting in 288 executions of the AES planner or, respectively, the calculation of 5708 trajectories. Trajectories are calculated with a time-discretization of 20 ms and an average planning horizon of 3.22 s (std. 0.78 s). By analysis of information from the logging system, an average runtime per calculated trajectory of τ_{traj} := 176 mus (std. 34 mus) is determined. The evaluation of the risk function $R_{k,tot}$ for each trajectory sample at each time-step additionally adds significant to the overall runtime performance. An average runtime per trajectory of τ_{risk} := 930 mus (std. 240 mus) is determined for those evaluations in the demonstrator vehicle. As a conservative assessment, the average runtime per sampled

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

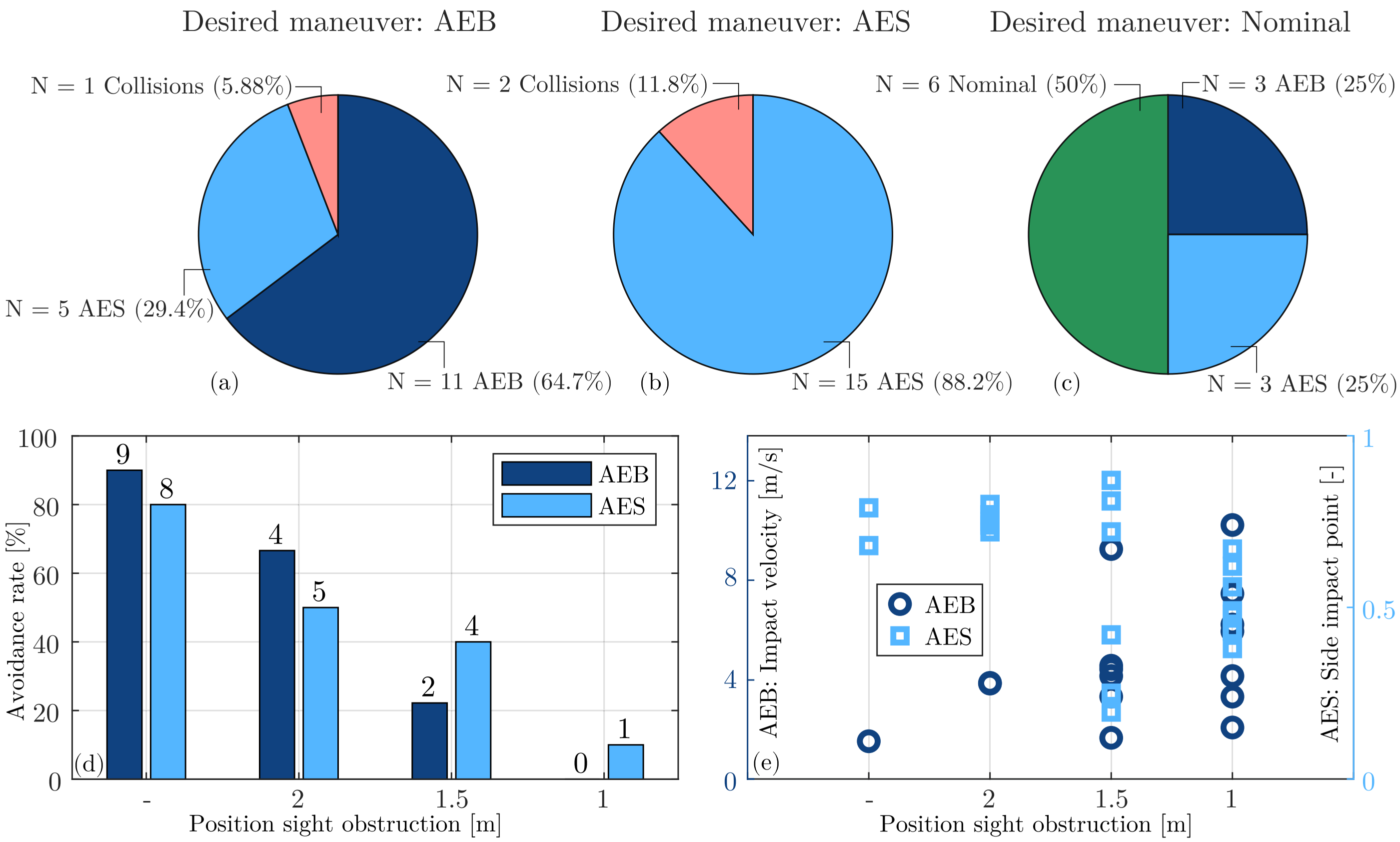


Fig. 13. Results: Overall system performance. The upper row (a) - (c) corresponds to the system response for accident avoidance maneuvering with sight obstruction (4 m lane width), and the lower row (3.5 m lane width) represents the accident avoidance rates (d) and collision impact conversion (e). While the upper row tests the decision-making performance (i.e., the logic of Figure 7 enabled), the lower row corresponds to test runs with only one emergency maneuver option available to the vehicle (i.e., the logic of Figure 7 disabled and one maneuver pre-selected). Thus, the avoidance potential of AES and AEB could be evaluated separately. Note that the leftmost entries in (d) and (e) correspond to runs without sight obstruction. In (d), the numbers above the bars indicate the number of tests leading to avoidance in the respective setting. In (e), the left y-axis represents the vehicle's velocity in cases where the AEB did not avoid a collision, and the right y-axis corresponds to the side impact location for cases where the AES did not avoid a collision. Here, 0 represents the front end, and 1 represents the rear end of the vehicle's side.

trajectory can be estimated according to

$$\tau_{\text{traj}}^{\text{AES}} := \tau_{\text{traj}} + \tau_{\text{risk}} = 1.10 \text{ ms(std. } 0.27\text{ms)},$$

where perfect correlation of τ_{traj} and τ_{risk} is assumed, leading to a linear combination of the respective uncertainties. No specific measures for runtime optimization are implemented in the calculation of trajectories and risk function, such that there is a high potential for $\tau_{\text{traj}}^{\text{AES}}$ to be further decreased once optimizations to a specific computing hardware are performed. The AES trajectory replanning cycle of the demonstrator vehicle is 200 ms. It is thus assessed, that about 120 AES trajectories can be calculated per planning cycle at a confidence level of about 95 %. We conclude that the runtime is sufficiently low for time-critical maneuvering in the investigated use case.

3) Decision-Making for Accident Avoidance: The objective is to experimentally evaluate the decision-making performance of the test vehicle. Therefore, we set up the sight obstruction (see Fig. 12) so that only one maneuver option out of nominal driving, AES maneuver, and AEB maneuver, is desired, similar to the simulation study from Sec. V-H. To trigger an AEB maneuver, the distance of the sight obstruction to the road is 3.65 m. For an AES maneuver, the distance is 1.25 m, since at this distance, the stopping time to a standstill for an AEB maneuver would exceed the time-to-collision. The distances of the sight obstruction is specified experimentally, as the exact response of the system is challenging to determine due to noise and delays in the perception, computation, and brake actuation. Therefore, the decision is subject to the response and environment of the entire system, compared to Sec. V-H, where the risk-based decision-making was evaluated stand-alone with a precisely known correct decision. For the AES and AEB test cases, the collision point is defined at the front right corner of the test vehicle. To test the nominal system response, the dummy is programmed to pass closely before the ego vehicle without colliding. The results are provided in Fig. 13(a) - (c). For the AEB scenario (Fig. 13(a)), the system did successfully avoid a collision with the desired AEB maneuver for 64.7 % of test runs. The system avoids most collisions for cases with significant sight obstructions, where AES is the preferred option (Fig. 13(b), wherein 88.2 % of cases, a collision is successfully avoided with the desired AES maneuver. In the case where the pedestrian walks closely before the vehicle passes (Fig. 13(c)), the vehicle responds with a 50 % false positive rate. Given the uncertainty, noise, and system delays, a significant false positive rate is expected when the dummy passes very closely to the vehicle. As collision avoidance is prioritized, we believe conservative decision-making in close encounters yields a safety benefit.

4) Accident Avoidance Rate Per Maneuver: To obtain the accident avoidance rates for the individual avoidance maneuvers with different obstruction distances, we restrict the system only to perform one maneuver option besides nominal driving, i.e., AES or AEB (contrasting to Sec. V-I.3, where all maneuver options where permitted). Such allows for analyzing the AES avoidance potential for cases where the AEB cannot avoid an accident, i.e., extending the field-of-effect of state-of-the-art advanced safety features. The sight obstruction is placed in such a way that 1 m, 1.5 m and 2 m of lateral distance between the sight obstruction and the lane border remain (see Fig. 12). As a reference, the scenario is also tested without any sight obstruction. Fig. 13(d) shows both AES and AEB avoidance rates for the different settings of sight obstruction. The exact number of corresponding runs is given at the top end of each bar. Note that the number of repetitions varies for the different obstruction settings. The figure shows a higher AEB avoidance rate than the AES avoidance rate for cases without sight obstruction and for 2 m sight obstruction distance. For the closer obstruction distance of 1.5 m, leading to a later trigger time for the systems, the AES avoidance rate exceeds the AEB avoidance rate. Only one AES intervention avoids an accident for 1 m obstruction distance. As expected, both AEB and AES avoidance rates decrease with closer obstruction settings.

5) Collision Impact Point and Velocity: We used the test setup from Sec. V-I.4, to analyze the impact constellation for cases, where the system could not avoid a collision with a given maneuver option. The impact velocity was reduced for all cases where the AEB did collide; likewise, for all cases where the AES did not lead to complete avoidance, the collision was converted from a front to a side collision. Fig. 13(e) displays the AEB impact velocity (left y-axis) and the AES side impact conversion (right y-axis). While for the base case, i.e., without sight obstruction, and for the 2 m sight obstruction case, the AES impact points were notably skewed towards the rear of the vehicle, for the 1.5 m and 1 m case the impact points where scattered across the vehicle's side, with the same approximate mean impact point.

VI. Discussion

The results are manifold; on the one hand, simulation studies show the benefits of our proposed prediction and risk-based decision-making module; on the other, we analyze the system's real-time capabilities, and demonstrate the system's functionality as well as safety benefits in a representative real-world driving scenario.

The prediction model (Sec. V-D) exemplifies that state-of-the-art models from literature display insufficient performance on noisy data, as found in real-world applications. Our proposed approach improves the prediction performance by robsutifying against perception noise, often caused by the lack of object tracking history, as found in our sight obstruction scenarios.

In Sec. V-H, we analyzed the decision-making performance of the proposed probabilistic risk-based method and compared it to a distance-based risk measure. We found that deterministic collision checking, in fact, reduces the decision-making quality for our probabilistic risk measure. Our proposed probabilistic risk estimation exhibits strong robustness against uncertainty whenever the collision check is removed, which, in that case, demonstrates a significant improvement over the comparative distance-based risk measure. We speculate that the removal of deterministic collision checking in uncertain environments, i.e., the real world, can thus be beneficial for the safety of the deployed system.

To account for real-world deployment, we also analyzed the run-time of our system. Our proposed MPC-based path planner (Sec. V-B) satisfies our update rate needs while displaying sufficiently low lateral errors in its planning tasks. Here, accounting for dynamic objects with the path planner would be feasible, as we could increase the update rate significantly. Similarly, it is argued that the generated trajectories and input signals are well suited as the expected planning errors remain small (Sec. V-F). As a pipeline, the risk estimation, trajectory planning, and decision-making execute their tasks in a satisfactory time (Sec. V-I.2). As such, the presented approach can be considered real-time capable for the investigated highly time-critical use cases.

The real-world accident avoidance performance (Sec. V-I.3) displayed the desired behavior in most test runs. Our system was tuned to be conservative, which exhibits a significant false-positive rate for close misses. Furthermore, the avoidance rate of the respective emergency maneuver was experimentally studied (Sec. V-I.4). Here, we found real-world evidence that AES can generate an additional accident avoidance potential in crossing pedestrian scenarios with sight obstructions compared to state-of-the-art AEB systems. Also, while the AEB reduced the impact velocity in unavoidable collisions, the AES converted these cases to side collisions in all cases, potentially decreasing the collision severity significantly. Due to using a pedestrian dummy on a non-stopping platform, we speculate that a human pedestrian is less likely to walk into a vehicle's side. Additionally, even if a human collides with the vehicle's side, we speculate that the associated harm is reduced (compared to a frontal collision), as accidentology [38] shows that the percentage of KSI pedestrians is approximately half that between a front and a side collision, located a meter behind the vehicle's front. For accidents to the rear half of the vehicle's side no data, i.e., no accidents, are among the $n = 1805$ cases recorded in the database. With the additional data, the results indicate that the AES system may reduce the harm incurred by pedestrians by converting frontal into side collisions.

VII. Conclusion and Future Work

This work focuses on automated vehicle technologies that have the potential to improve pedestrian safety. Therefore, car-to-pedestrian accidents are derived and, subsequently, a system design comprised of a model predictive control-based path planner and trajectory generator is proposed to avoid such accidents. We propose a robustified pedestrian prediction model, providing predictions with associated estimated uncertainties to a risk-based decision-making module, that works in a closed loop with the trajectory generator. We compare some of our algorithms against state-of-the-art algorithms, analyze run-times, and evaluate the system's accident avoidance capabilities in real-world testing.

We find: 1) our proposed system run-time is sufficiently fast for time-critical maneuvering, 2) our robustified prediction model outperforms a state-of-the-art model by accounting for noisy data and late detections, 3) the probabilistic risk measure improves the decision-making when compared to a deterministic model in the presence of uncertainty, 4) deterministic collision checking acts as a significant bottleneck to the decision-making performance under uncertainty, 5) the effectiveness of autonomous emergency steering (AES) is demonstrated, extending emergency maneuvering beyond braking, 6) we reduce accident severity in the presence of unavoidable collisions, AES is found to convert frontal collisions to side collisions, which are generally associated with less killed or severely injured pedestrians.

For future work, we suggest increasing computational efficiency in risk estimation; while sufficient in our implementation, Monte Carlo sampling will become a computational burden in more complex scenarios. The path planning algorithm could include dynamic objects and connected agents, such that navigation in complex multi-agent environments becomes viable by the path planner. While our AES functionality improves emergency maneuvering, the scope of the system could be extended (e.g., out-of-lane evasions or cyclists crossing). However, when doing so, it is important to ensure that hardware equipment, and in particular the sensor signal processing, satisfy the extended operational design domain of such a system. Regarding deployments in real-world environments, the software architecture would benefit from redundancies, and the perception and road user prediction algorithms must be suited, robust, and trained to accommodate more complex scenarios and road user behaviors.

References

[1] M. M. Morando, Q. Tian, L. T. Truong, and H. L. Vu, "Studying the safety impact of autonomous vehicles using simulation-based surrogate safety measures," *J. Adv. Transp.*, vol. 2018, pp. 1–11, Apr. 2018.

[2] A. Bálint et al. *Safe-Up Deliverable Report: D2.6 Use Case Definitions and Initial Safety-Critical Scenarios*. Accessed: Mar. 26, 2026. [Online]. Available: https://www.safe-up.eu/deliverables

[3] *GIDAS: German In-Depth Accident Study*. Accessed: Mar. 26, 2026. [Online]. Available: https://www.gidas.org/start.html

[4] *Care Database*. Accessed: Mar. 26, 2026. [Online]. Available: https://roadsafety.transport.ec.europa.eu/european-road-safety-observatory/methodology-and-research/caredatabase_en

[5] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443–58469, Mar. 2020.

[6] F. Camara et al., "Pedestrian models for autonomous driving part I: Low-level models, from sensing to tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 10, pp. 6131–6151, Oct. 2021.

[7] F. Camara et al., "Pedestrian models for autonomous driving part II: high-level models of human behavior," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5453–5472, Sep. 2021.

[8] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *Int. J. Robot. Res.*, vol. 39, no. 8, pp. 895–935, Jul. 2020.

[9] M. M. Sánchez, E. Silvas, J. Elfring, and R. van de Molengraft, "Robustness benchmark of road user trajectory prediction models for automated driving," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 4865–4870, Nov. 2023.

[10] R. McAllister et al., "Concrete problems for autonomous vehicle safety: Advantages of Bayesian deep learning," in *Proc. Twenty-Sixth Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 4745–4753.

[11] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar, "The impact of adverse weather conditions on autonomous vehicles," *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 103–111, Feb. 2019.

[12] D. Weihmayr, F. Sezgin, L. Tolksdorf, C. Birkner, and R. N. Jazar, "Predicting the influence of adverse weather on pedestrian detection with automotive radar and LiDAR sensors," 2024, *arXiv:2405.12736*.

[13] J. Müller, J. Strohbeck, M. Herrmann, and M. Buchholz, "Motion planning for connected automated vehicles at occluded intersections with infrastructure sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17479–17490, Oct. 2022.

[14] B. Ivanovic, Y. Lin, S. Shrivastava, P. Chakravarty, and M. Pavone, "Propagating state uncertainty through trajectory forecasting," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 2351–2358.

[15] *International Organization for Standardization: Road Vehicles-Safety of the Intended Functionality*, 2022.

[16] J. Vaicenavicius, T. Wiklund, A. Grigaite, A. Kalkauskas, I. Vysniauskas, and S. D. Keen, "Self-driving car safety quantification via component-level analysis," *SAE Int. J. Connected Automated Vehicles*, vol. 4, no. 1, pp. 35–45, Mar. 2021.

[17] S. Kolekar, J. de Winter, and D. Abbink, "Human-like driving behaviour emerges from a risk-based driver model," *Nature Commun.*, vol. 11, no. 1, p. 4850, Sep. 2020.

[18] M.-Y. Yu, R. Vasudevan, and M. Johnson-Roberson, "Risk assessment and planning with bidirectional reachability for autonomous driving," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 5363–5369.

[19] A. Pierson, W. Schwarting, S. Karaman, and D. Rus, "Navigating congested environments with risk level sets," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 5712–5719.

[20] M. Geisslinger, F. Poszler, J. Betz, C. Lütge, and M. Lienkamp, "Autonomous driving ethics: From trolley problem to ethics of risk," *Philosophy Technol.*, vol. 34, no. 4, pp. 1033–1055, Dec. 2021.

[21] *Regulation No 79-Uniform Provisions Concerning the Approval of Vehicles With Regard to Steering Equipment [2018/1947]*. Accessed: Mar. 26, 2026. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1416170084502&uri=CELEX:32014R0269

[22] A. Neukum, J. Paulig, L. Frömmig, and R. Henze, "Untersuchung zur Wahrnehmung von Lenkmomenten bei PKW," in *FAT-Schriftenreihe*, no. 222. VDA, 2010.

[23] S. Nikolaou et al. *Safe-Up Deliverable Report: D3.9 Active Safety Systems Specification and Risk Analysis Update*. Accessed: Mar. 26, 2026. [Online]. Available: https://www.safe-up.eu/

[24] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[25] C. van der Ploeg, R. Smit, A. Teerhuis, and E. Silvas, "Long horizon risk-averse motion planning: A model-predictive approach," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 1141–1148.

[26] C. van der Ploeg, M. Braat, B. Masini, J. Brouwer, and J.-P. Paardekooper, "Connecting the dots: Context-driven motion planning using symbolic reasoning," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2023, pp. 1–8.

[27] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-92-01, Jan. 1992.

[28] M. M. Sánchez, J. Elfring, E. Silvas, and R. van de Molengraft, "Scenario-based evaluation of prediction models for automated vehicles," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 2227–2233.

[29] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1696–1703, Apr. 2020.

[30] C. Löffler et al. *D3.6 Vehicle Demonstrator for Trajectory Planning and Control for Combined Automatic Emergency Braking and Steering Manoeuvres*. Accessed: Mar. 26, 2026. [Online]. Available: https://www.safe-up.eu/deliverables

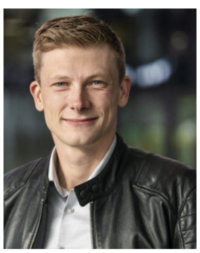
[31] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, Mar. 1964.

[32] C. Löffler, T. F. Gloger, and S. Joos, "Synthesis and application of constrained flatness-based real-time trajectory planning for autonomous emergency steering," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2023, pp. 1–8.

[33] S. Joos, M. Bitzer, R. Karrelmeyer, and K. Graichen, "Constrained online trajectory planning for nonlinear flat SISO systems using a switched state variable filter," *Automatica*, vol. 110, Dec. 2019, Art. no. 108583.

[34] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: Introductory theory and examples," *Int. J. Control*, vol. 61, no. 6, pp. 1327–1361, Jun. 1995.

[35] W. Kühnel and B. Hunt, *Differential Geometry: Curves Surfaces-Manifolds*, 2nd ed. Providence, RI, USA: American Mathematical Society, 2002.
[36] J. Ziegler and C. Stiller, "Fast collision checking for intelligent vehicle motion planning," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2010, pp. 518–522.
[37] L. Tolksdorf, A. Tejada, N. van de Wouw, and C. Birkner, "Risk in stochastic and robust model predictive path-following control for vehicular motion planning," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2023, pp. 1–8.
[38] N. Parera et al. (2023). *Safe-Up: D5.3 Active Safety System Assessment Results*. Accessed: Mar. 26, 2026. [Online]. Available: https://www.safe-up.eu/deliverables
[39] L. Tolksdorf et al. (2023). *Safe-Up: D3.8 Verification Report for Demos 2, 3 and 4*. Accessed: Mar. 26, 2026. [Online]. Available: https://www.safe-up.eu/deliverables
[40] S. Ettinger et al., "Large scale interactive motion forecasting for autonomous driving: The Waymo open motion dataset," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 9710–9719.
[41] J. Wang, J. Wu, and Y. Li, "The driving safety field based on driver–vehicle–road interactions," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2203–2214, Aug. 2015.
[42] Y. Li, K. Li, Y. Zheng, B. Morys, S. Pan, and J. Wang, "Threat assessment techniques in intelligent vehicles: A comparative survey," *IEEE Intell. Transp. Syst. Mag.*, vol. 13, no. 4, pp. 71–91, Apr. 2021.
[43] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *Proc. 11th Int. Conf. Inf. Fusion*, Cologne, Germany, Jul. 2008, pp. 1–6.

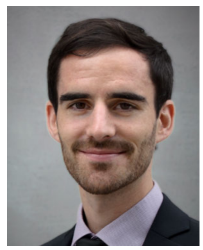


Leon Tolksdorf received the B.Eng. degree in automotive engineering and management and the M.Sc. degree in automotive engineering from Munich University of Applied Sciences, in 2017 and 2019, respectively, and the Ph.D. degree from the Dynamics and Control Group, Eindhoven University of Technology, Eindhoven, The Netherlands, in 2026. Since 2020, he has been a Researcher with the CARISSMA Institute of Safety in Future Mobility, Ingolstadt, Germany. His main research interests are stochastic optimization for motion planning, computation of stochastic risk measures, and risk-aware automated vehicle behavior generation in single and multi-agent control.

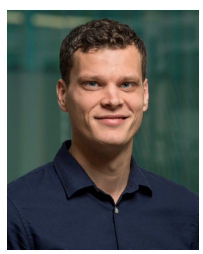
Manuel Muñoz Sánchez received the B.Sc. and M.Sc. degrees in computer science and engineering and the Ph.D. degree from Eindhoven University of Technology, Eindhoven, The Netherlands, in 2017, 2020, and 2025, respectively. His research interests include the applications of deep learning methods for automated driving, with a focus on thorough evaluations of trajectory prediction models and their integration within the autonomy stack.

Christian Loeffler received the B.Eng. degree in mechatronics engineering from the Esslingen University of Applied Sciences, Germany, in 2016. Since 2016, he has been with the Advanced Development Department, Cross-Domain Computing Solutions Division, Robert Bosch GmbH. His research interests include accident avoidance and mitigation in mixed-autonomy environments, human machine interactions in safety critical situations, and safe by design planning and control strategies.

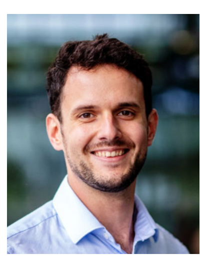
Timm F. Gloger received the B.Sc. and M.Sc. degrees in physics and the Ph.D. degree from the University of Siegen, Germany, in 2008, 2011, and 2025 respectively. From 2018 to 2024, he was with the Advanced Development Department, Cross-Domain Computing Solutions Division, Robert Bosch GmbH. His research interests include accident avoidance and mitigation in mixed-autonomy environments, human–machine interactions in safety critical situations, and safe by design planning and control strategies.



Yannick Fischer received the B.Sc. and M.Sc. degrees in mechanical engineering from the Technical University of Darmstadt, Germany, in 2014 and 2017, respectively. Since 2018, he has been with the Advanced Development Department, Cross-Domain Computing Solutions Division, Robert Bosch GmbH. His research interests include accident avoidance and mitigation in mixed-autonomy environments, human–machine interactions in safety critical situations, and safe by design planning and control strategies.



Robin Smit received the B.Sc. and M.Sc. degrees in mechanical engineering and system and control from Eindhoven University of Technology, Eindhoven, The Netherlands, in 2013 and 2016, respectively. Since 2019, he has been a Research Scientist with Netherlands Organisation for Applied Scientific Research (TNO), Helmond, The Netherlands. His work focuses on perception technologies of automated vehicles, including localization, ego state estimation, object tracking and prediction, and road modeling.



Chris van der Ploeg received the B.Sc. degree in mechanical engineering and the M.Sc. degree (cum laude) in systems and control from Delft University of Technology, Delft, The Netherlands, in 2016 and 2018, respectively, and the Ph.D. degree from Eindhoven University of Technology, Eindhoven, The Netherlands, in 2024. Since 2019, he has been a Research Scientist with the Integrated Vehicle Safety Department, Netherlands Organisation for Applied Scientific Research (TNO), Helmond, The Netherlands. His current research interests include fault diagnosis methods, fault mitigation strategies/methods, and risk-averse motion planning for connected and cooperative vehicles.