SAFE-UP

D3.3 Vehicle demonstrator for trajectory planning and control for combined automatic emergency braking and steering manoeuvres including system for VRU detection, motion planning and trajectory control to enhance real world performance

Primary Author(s)	Christian Löffler, Timm Gloger | Robert Bosch GmbH

Related Work Package	WP3
Version/Status	2.1 | Final version
Issue date	29/11/21
Deliverable type	Demonstrator
Dissemination Level	PU

Project Acronym	SAFE-UP
Project Title	proactive SAFEty systems and tools for a constantly UPgrading road environment
Project Website	www.safeup.eu
Project Coordinator	Núria Parera | Applus IDIADA
Grant Agreement No.	861570

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

Co-Authors

Name	Organisation
Emilia Silvas	TNO (Netherlands Organisation for Applied Scientific Research)
Manuel Muñoz Sánchez	TU/e (Eindhoven University of Technology)
Leon Tolksdorf	THI (Technische Hochschule Ingolstadt)
Daniel Weihmayr	THI (Technische Hochschule Ingolstadt)
Volker Labenski	Audi AG
Markus Koebe	Audi AG
Johann Stoll	Audi AG
Carina Vogl	Cariad SE
Hiroki Watanabe	Cariad SE
Robin Smit	TNO

Reviewers

Version	Date	Reviewer (Name/Organisation)	Action (Checked/Approved)
1.0	15/11/2021	Jorge Lorente Mallada / TME	Checked
1.0	10/11/2021	Stella Nikolaou / CERTH	Checked
1.0	15/11/2021	Christian Birkner / THI	Checked

Document Distribution

Version	Date	Distributed to
2.0	23/11/2021	Coordination Team
2.1	29/11/2021	Submission in the EC System
		Approved by the EC

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

Document history

Version	Date	Author	Summary of changes
0.1	09/23/2021	Christian Birkner (THI)	Basic template draft. Sent out for distribution to partners
0.2	11/10/2021	Manuel Muñoz Sánchez (TU/e)	Subsystem VRU Intent & Trajectory Prediction
0.3	13/10/2021	Emilia Silvas (TNO)	Software architecture, subsystem Path Planning
0.4	15/10/2021	Christian Löffler (Bosch)	Scenario selection
0.5	20/10/2021	Timm Gloger (Bosch)	Subsystems Trajectory Generation, Vehicle Control, Sensor Input and Object Fusion & Tracking
0.6	25/10/2021	Leon Tolksdorf (THI)	Subsystems Crash Prediction & Avoidance Estimation, Safety Decision
0.7	25/10/2021	Carina Vogl, Johann Stoll, Volker Labenski (Cariad SE/Audi AG)	Data Base for scenario selection
0.8	27/10/2021	Timm Gloger, Christian Löffler (Bosch)	First complete draft
0.9	28/10/2021	Robin Smit (TNO)	Path planning details
1.0	03/11/2021	Christian Löffler (Bosch)	External review version
1.1	19/11/2021	Christian Löffler, Timm Gloger, (Bosch)	Incorporate external review comments, add initial test results
2.0	22/11/2021	Christian Löffler (Bosch)	Final version
2.1	29/11/2021	Christian Löffler (Bosch)	Final version ready for submission

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

Copyright statement

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

Executive summary

This Deliverable falls under the SAFE-UP Project Work Package 3 “Active safety systems for vehicle-VRU interaction” and specifically under the Task 3.4 “Advanced intervention functions to avoid critical events”. It is a purely technical document that targets to support the efficient monitoring of the technical developments for Demonstrator 3 “Vehicle demonstrator for trajectory planning and control for combined automatic emergency braking and steering maneuvers including system for VRU detection, motion planning and trajectory control to enhance real world performance”.

The present document is the first of two deliverables related to Demo 3 and focuses on the scenario selection method, initial scenario selection results, and a description of the Demo 3 algorithms under development. Initial test results show the current status and maturity level of the algorithms.

A detailed description of the Demo 3 architecture and technical specifications can be found in the deliverable report D3.1 (SAFE-UP, Deliverable report D3.1, 2021).

This report is organized as follows: an overview of the hardware and software architecture are presented in section 2. In section 3, the scenario selection method, data base and the simulation results are described. The latter are used for an initial Demo 3 scenario specification. Section 4 gives an overview about the developed algorithm subsystems and their current development status. Initial test results are presented in section 5, followed by a discussion, conclusion and a description of the next steps in section 6.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement 861570.

Table of Contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

List of figures

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

List of tables

List of abbreviations

Abbreviation	Meaning
AEB	Autonomous emergency braking
AES	Autonomous emergency steering
LSTM	Long short-term memory
VRU	Vulnerable road user
PCM	Pre-crash matrix
P-CLwoSO	Pedestrian crossing left without sight obstruction
RNN	Recurrent neural network
ROS	Robot Operating System
GIDAS	German In-Depth Accident Study
TTC	Time to collision
UTYP	Unfalltyp (accident type identifier)
GNSS	Global navigation satellite system
RTK	Real time kinematics
IMU	Inertial measurement system
UTM	Universal Transverse Mercator coordinate system
ODD	Operational design domain
MPC	Model predictive control

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

1. Introduction

This deliverable reports on the current development status of WP3 Demo 3. The scope of Demo 3 is to develop advanced vehicle dynamics intervention functions to avoid or mitigate critical events. The demonstrator will include a vehicle with trajectory control algorithms for both emergency braking and steering.

The purpose of this document is mainly to support the technical coordination and monitoring of the Demo 3 development. It is therefore working as a technical document, supporting the work of the system developers throughout the process, as well as the related work that will be performed in T3.6 focusing on technical verification. This version of the deliverable focuses on the scenario selection method, initial scenario selection results, and a description of the Demo 3 algorithms under development supported by initial test results showing the current status and maturity level of the algorithms.

An updated version of this Deliverable is scheduled for Month 26 of the project (July 2022), when Demo 3 has completed its development phase.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

2. Architecture

2.1 Demonstrator hardware architecture

A Bosch development vehicle as depicted in Figure 1 is used as Demo 3 integration platform. The vehicle contains several sensors and actuators with enhanced interfaces as well as a computing platform utilizing the Robot Operating System (ROS) as middleware to facilitate communication between different subsystems.

A detailed description of the Demo 3 hardware architecture and technical specification can be found in the deliverable report D3.1 (SAFE-UP, Deliverable report D3.1, 2021).

Figure 1: Demo 3 integration platform. A Bosch development vehicle featuring a radar/video sensor set and steering and braking interfaces with enhanced dynamics.

2.2 Demonstrator software architecture

The software for Demo 3 consists of several functionalities developed by different partners. Figure 2 shows the high-level interactions between the functionalities developed by the partners for Demo 3. These functionalities are implemented in the ROS2 (Robot Operating System) framework, which acts as a middleware and facilitates communication between the functionalities. Within the ROS framework, the functionalities are implemented as separate executables, or nodes, and these nodes communicate with each other via topics, using a publish-subscribe pattern or services, using a server-client pattern.

A detailed description of the Demo 3 software architecture and technical specification can be found in the deliverable report D3.1 (SAFE-UP, Deliverable report D3.1, 2021).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

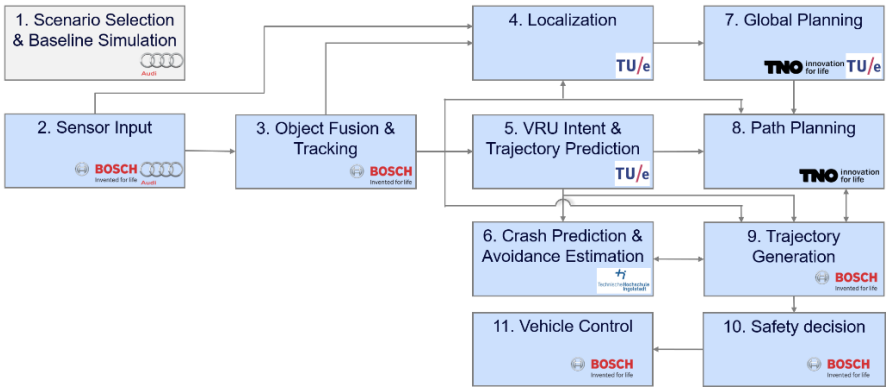

Figure 2: High level interaction layouts between functionalities (light blue) and inputs needed from other work packages (grey block).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

3. Scenario selection

With the main goal of developing advanced active safety systems including autonomous emergency steering (AES) as a novelty, special focus is given in understanding the potential field of effect of such a system, especially in comparison to current state-of-the-art active safety systems. Therefore, a simulative analysis to quantify a theoretical field of effect is performed.

The goal of the following Demo 3 scenario selection process is to identify scenarios that cannot be avoided by state-of-the-art systems and have the theoretical potential to be avoided by AES. These scenarios are then used to steer Demo 3 development towards a real-world safety benefit by directly addressing accident types that are not yet covered by any active safety system.

3.1 Scenario selection method

The scenario selection method is based on a simulation of generic implementations of Autonomous Emergency Braking (AEB) and Autonomous Emergency Steering (AES) systems. Those systems are then simulated using Pre-Crash-Matrix (PCM) accident data (German In-Depth Accident Study, n.d.). Based on an assessment of the accident avoidance potential, accident clusters are formed and specified by their parameter distributions. Figure 3 shows an overview of the simulation process.

The following chapters describe the data base, as well as the simulation assumptions and workflow.

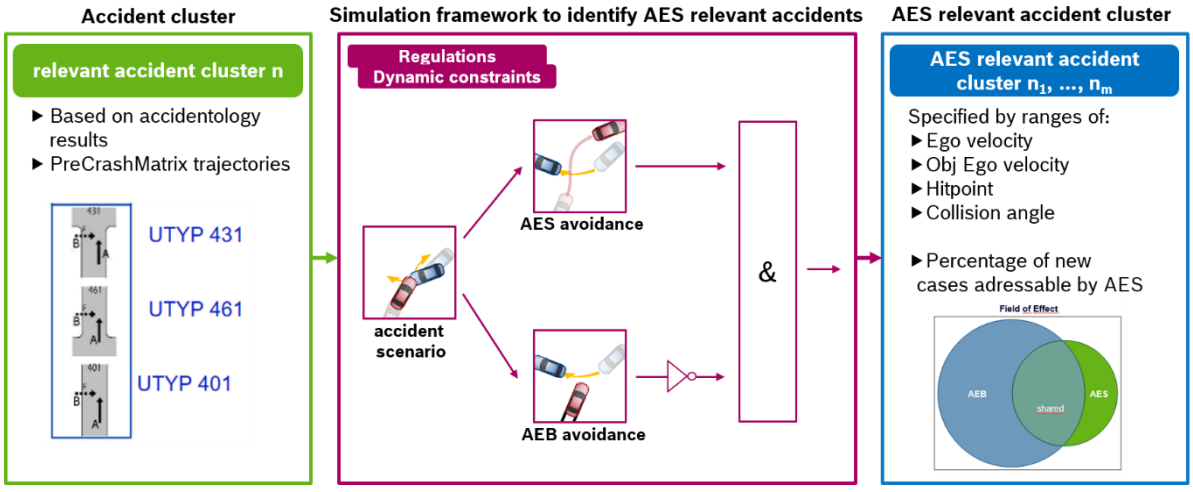

Figure 3: Overview of the simulation process for scenario selection. Clusters derived from accident data are used to simulatively assess the accident avoidance potential of an AES maneuver and build clusters of AES relevant traffic situations.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

3.1.1 Data base

In Task 2.1 of the SAFE-UP project, various analyses of crash data and naturalistic driving data were performed to derive scenarios for further work in the project. For car-to-VRU crashes in adverse weather conditions four use cases were selected, which are shown in Table 1. The focus of this analysis was on pedestrians and bicyclists as VRU types and on precipitation as it is significantly more prevalent in crashes with VRUs than other weather phenomena like fog. For the use cases, conflict scenarios with a high relative occurrence of precipitation, as well as conflict scenarios with a high absolute occurrence of precipitation were selected. The highest absolute number of cases under precipitation is in the conflict scenario cluster P-CLwoSO (Pedestrian crossing from left without sight obstruction), which is therefore selected with priority for the analysis in this document.

For more detailed information on the analysis and the selected use cases, refer to Deliverable D2.6 of the project SAFE-UP (SAFE-UP, Deliverable report D2.6, 2021).

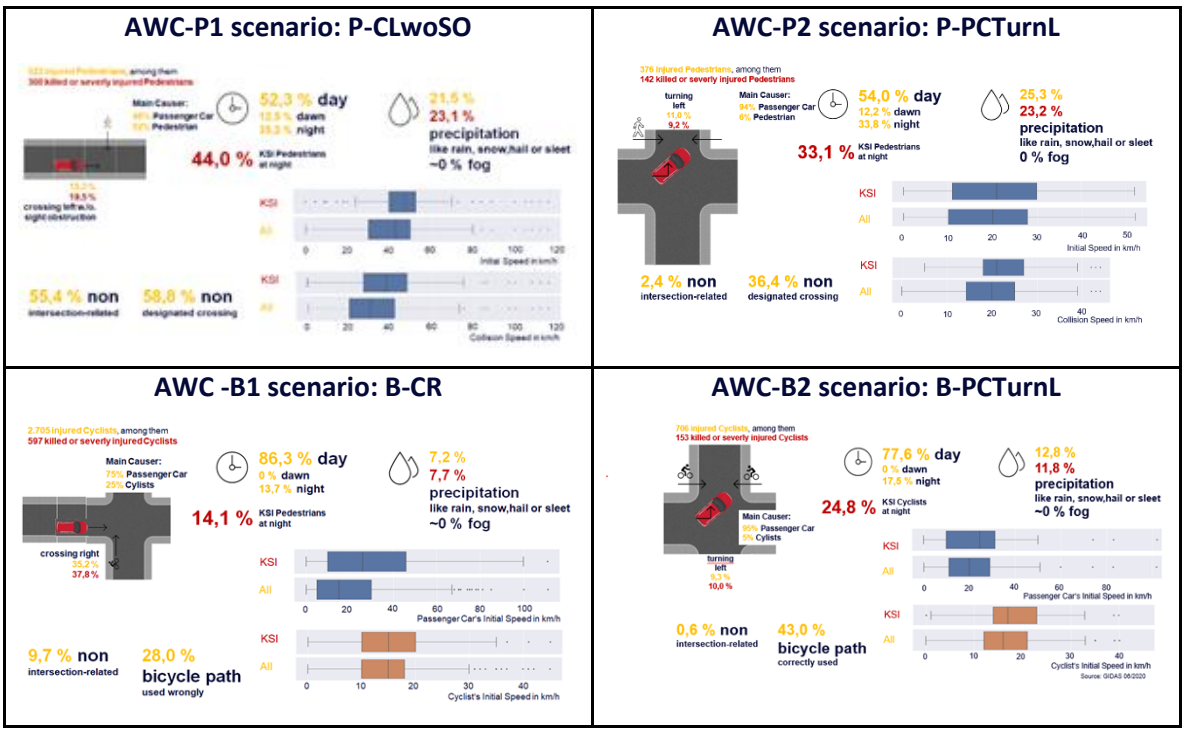

Table 1: Car-to-VRU scenarios recommended in D2.6 for consideration for safety systems with improved sensor performance (SAFE-UP, Deliverable report D2.6, 2021).

For the first iteration of the demonstrator development, special focus is given to the three most common UTYPs (401, 431, and 461) in the cluster P-CLwoSO, which are displayed in Figure 3. For these, 137 cases in the PCM database could be identified, 109 of which could be used for the simulation. The GIDAS-PCM database is a subset of the GIDAS database, which contains the kinematic information of the pre-crash phase, including trajectories and

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

speed information from the initiation of the conflict situation until the actual collision. Based on information from the crash reconstruction, the pre-crash phase is described at least approximately TTC=5 seconds before the collision and stored as a pre-crash matrix format (PCM).

In further iterations, the present method is planned to be applied to all UTYPs within the clusters and to the other relevant accident clusters from the SAFE-UP project scope.

3.1.2 Simulation assumptions

To be able to generate useful and realistic simulation results, several assumptions must be made. Figure 4 shows an overview of the general simulation assumptions.

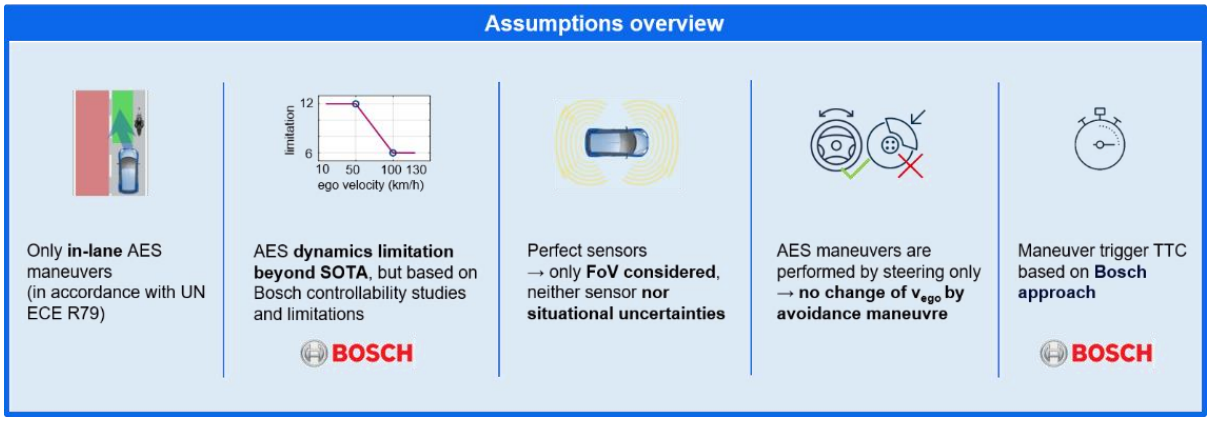

Figure 4: Simulation assumptions for the assessment of the accident avoidance potential of an AES maneuver.

The main limiting factors for the AES field of effect are expected to be the available space for an evasive maneuver, the allowed lateral dynamics and the trigger time of the system. The AES system under investigation is considered to be a SAE Level 2 system (SAE, 2018) and has to fulfill the legal requirements of the UNECE R79 regulation (UNECE, 2018), resulting in the following general simulation assumptions:

1. In accordance with the UNECE R79 regulation, the AES maneuver is limited to evade within the current ego lane only.
2. Based on Bosch controllability studies (Schneider, Schmitz, Ahrens, Löffler, & Neukum, 2018), the lateral dynamic interventions are limited to a maximum lateral acceleration of $5\,\frac{\mathrm{m}}{\mathrm{s}^2}$.
3. Sensor characteristics are considered by a field of view model only, sensor detection or situational uncertainties are not considered.
4. AES maneuvers are performed by steering only. A combination of AES and AEB is not considered in this initial step.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

5. The system trigger time is limited by a Bosch internal approach, which, if a driver does not trigger an intervention, delays the system trigger until it is estimated that a driver would not able to perform an avoidance maneuver on her own.

To assess the AES systems performance, criteria for collision avoidance have to be defined. An illustration of the chosen collision avoidance criteria is shown in Figure 5.

The first and most obvious criterion is *full collision avoidance*. In this case there is no contact of the ego vehicle and the pedestrian at any time during the AES maneuver.

As the velocity of the crossing pedestrian from the accident reconstruction is always constant and the simulation does not make any assumptions regarding reactions of the pedestrians to the ego vehicles evasive motion, cases occur where a collision of the pedestrian with the ego vehicles front is avoided, but the pedestrian gets in contact with the side of the vehicle.

Because of the missing pedestrian reactions, the fact that a pedestrian is able to reduce its velocity to stand-still on quasi-instant timescales and the assumption that side collisions may be less critical than frontal collisions, a *frontal collision avoidance* criterion is used as a second performance measure.

Figure 5: Avoidance criteria for the assessment of the accident/collision avoidance of an emergency steering or braking maneuver.

3.1.3 Simulation workflow

The simulation workflow is depicted as a flow chart in Figure 6.

The initial step of the simulation is the extraction of ego and VRU trajectories and dynamics as well as lane information from the PCM accident scenario. Based on the trajectories and dynamics, the Time-To-Collision (TTC) and the collision overlap are calculated for each

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

timestamp and then fed into the calculation of the system trigger. At the timestamp where TTC falls below the system trigger threshold AES and AEB maneuvers are initiated.

For the AEB maneuver, a longitudinal acceleration profile is used to calculate the vehicle's future motion until stand-still.

For the AES maneuver, two trajectories are calculated for every accident scenario: One evading to the left and one to the right. Both trajectories end on the maximum lateral displacement possible (given by the ego lane information) and use the maximum allowed lateral dynamics.

For both maneuvers, the vehicle dynamics of the component of motion which is not affected by the maneuver (lateral in case of AEB, longitudinal in case of AES) are assumed to be constant. For an AEB maneuver lateral dynamics are calculated with a model of constant acceleration, starting at the time of the system trigger. Likewise, for the AES maneuvers longitudinal dynamics are calculated with a model of constant acceleration, starting at the time of the system trigger.

The trajectories and dynamics of the ego maneuvers are then used to perform collision checks with the pedestrian trajectory to decide if *full collision avoidance* or *frontal collision avoidance* can be realized.

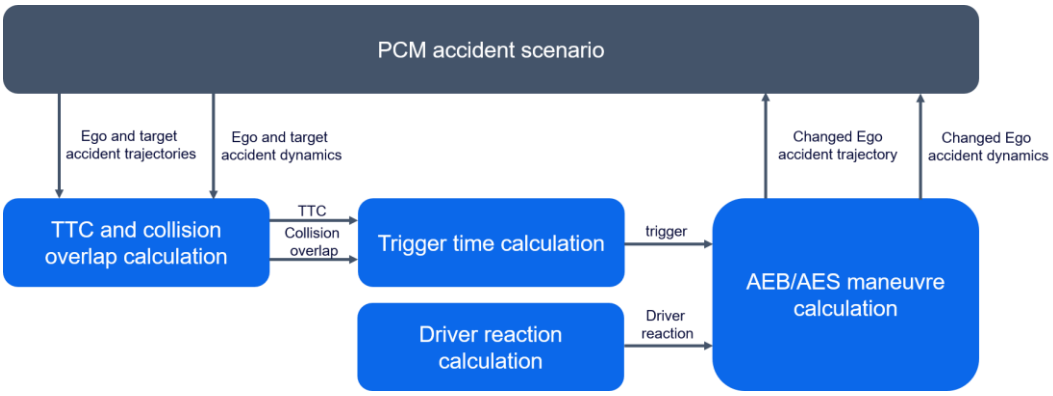

Figure 6: Logical view of the simulation workflow.

3.2 Selected scenarios for Demo 3

3.2.1 Overall results

The overall simulation results are separated into evaluations for the two collision avoidance criteria: *full collision avoidance* and *frontal collision avoidance*.

Figure 7 shows an illustration of the AEB and AES field of effect for *full collision avoidance*. Out of the 109 simulated PCM cases, 76.15% can be avoided by AEB and 54.45% only by

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

AEB and not by AES. 36.7% can be avoided by AES and 11.93% only by AES and not by AEB.

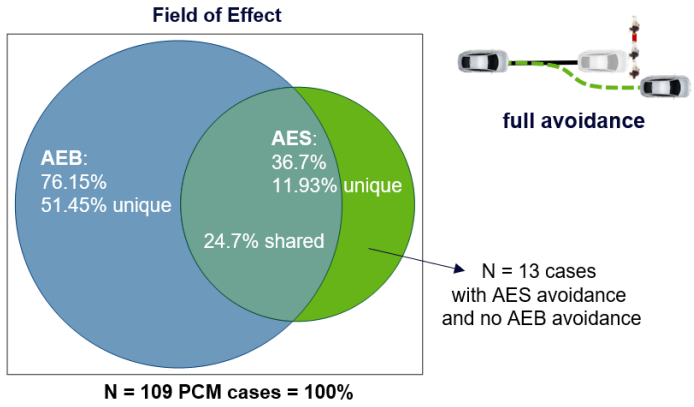

Figure 7: Field of effect estimations for AEB and AES maneuvers using the *full collision avoidance* criterion.

These 11.93% of sole AES avoidance potential, equaling a number of 13 cases, are the ones of interest for the Demo 3 development, as addressing these cases with novel avoidance functions would have a direct impact on total accident avoidance numbers.

Figure 8 contains distribution plots of ego vehicle and pedestrian velocities, angles between ego vehicle and pedestrian (all at the time of the original accident and the time of the system trigger), overlap of the ego vehicle and the pedestrian in the original accident and the system trigger time (green distributions) as well as a potential mitigation effect of an AEB by the reduction of collision velocity (blue distributions).

The distribution plots contain the discrete values of each case, shown as dark green or blue dots, as well estimated distributions around them. Due to the limited data the validity of theses distributions can not be guaranteed and should only be taken to guide the eye.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

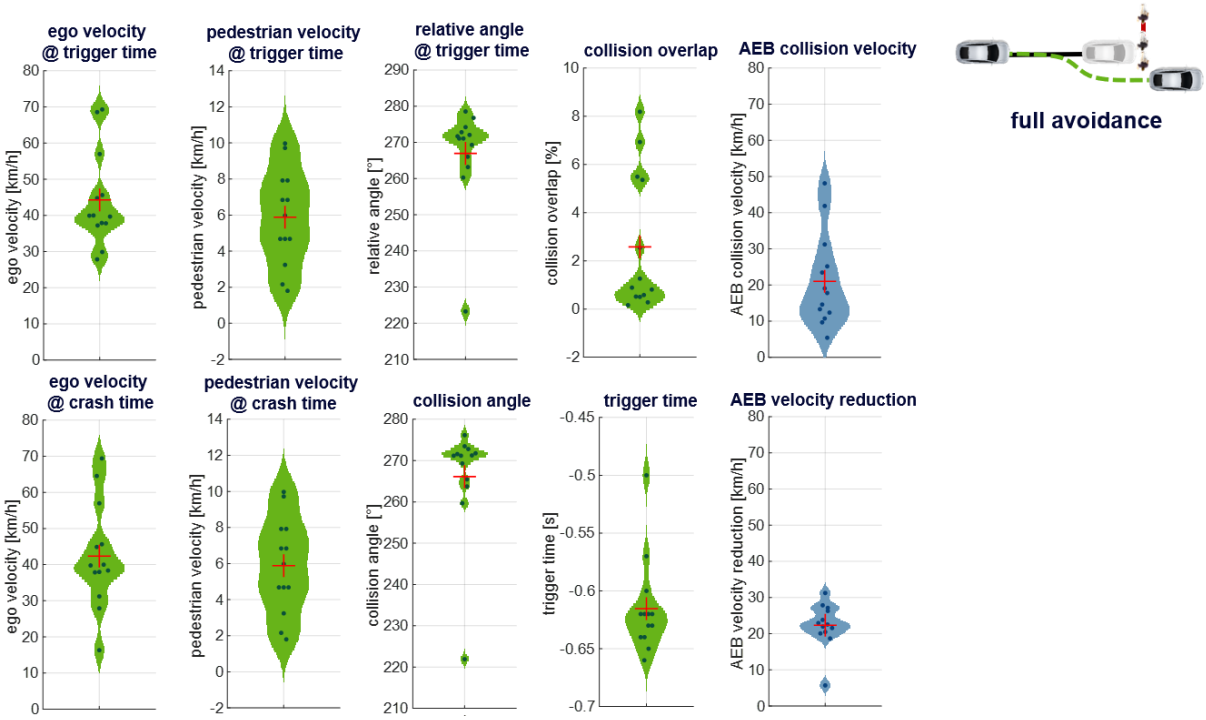

Figure 8: Parameter distributions for PCM cases with unique AES avoidance potential (green) and AEB mitigation potential (blue) using the *full collision avoidance* criterion.

The collision hitpoint of the original accident is a parameter of special interest here, as it already separates the original scenarios along the system performance criteria. The original scenarios can be clustered into frontal collisions and side collisions, where the latter cluster would meet the *frontal collision avoidance* criterion even without any emergency system. An illustration of the hitpoint definition can be found in Figure 9.

Six out of the 13 cases where the AES maneuver is able to realize *full collision avoidance* are frontal collisions with a hitpoint on the leftmost part of the ego vehicle's front (hitpoint = 0 for all cases). The remaining seven cases are side collisions with hitpoints predominantly distributed along the front-half of the vehicle side (see Figure 10).

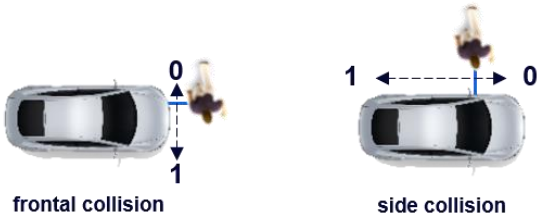

Figure 9: Hitpoint definitions for frontal and side collisions as used by the scenario selection simulations.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

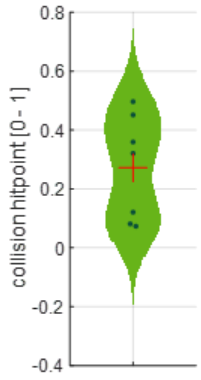

Figure 10: Distribution of original accident hitpoints in PCM cases with unique AES avoidance potential using the *full collision avoidance* criterion.

Using the *frontal collision avoidance* criterion 51.38% out of the 109 simulated PCM cases can be avoided by AES and 9.17% (equals ten cases) only by AES and not by AEB. The AEB field of effect remains identical. Cases with side-collisions in the original accident scenario are not considered here, since no intervention is needed to fulfill the *frontal collision avoidance* criterion. Figure 11 shows the AEB and AES field of effect for the *frontal collision avoidance* criterion.

Analog to the full avoidance analysis, Figure 12 contains plots of the parameter distributions for the *frontal collision avoidance* evaluation.

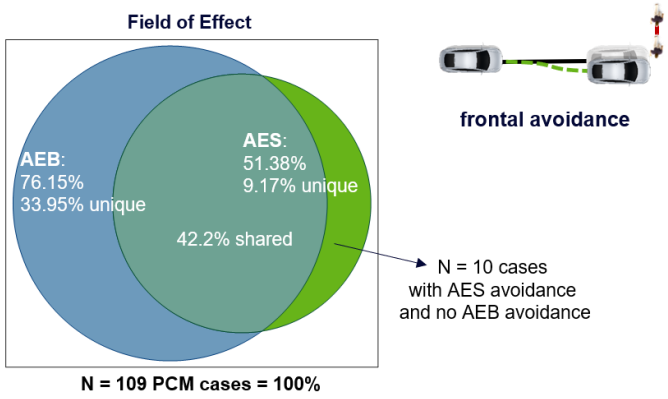

Figure 11: Field of effect estimations for AEB and AES maneuvers using the *frontal collision avoidance* criterion.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

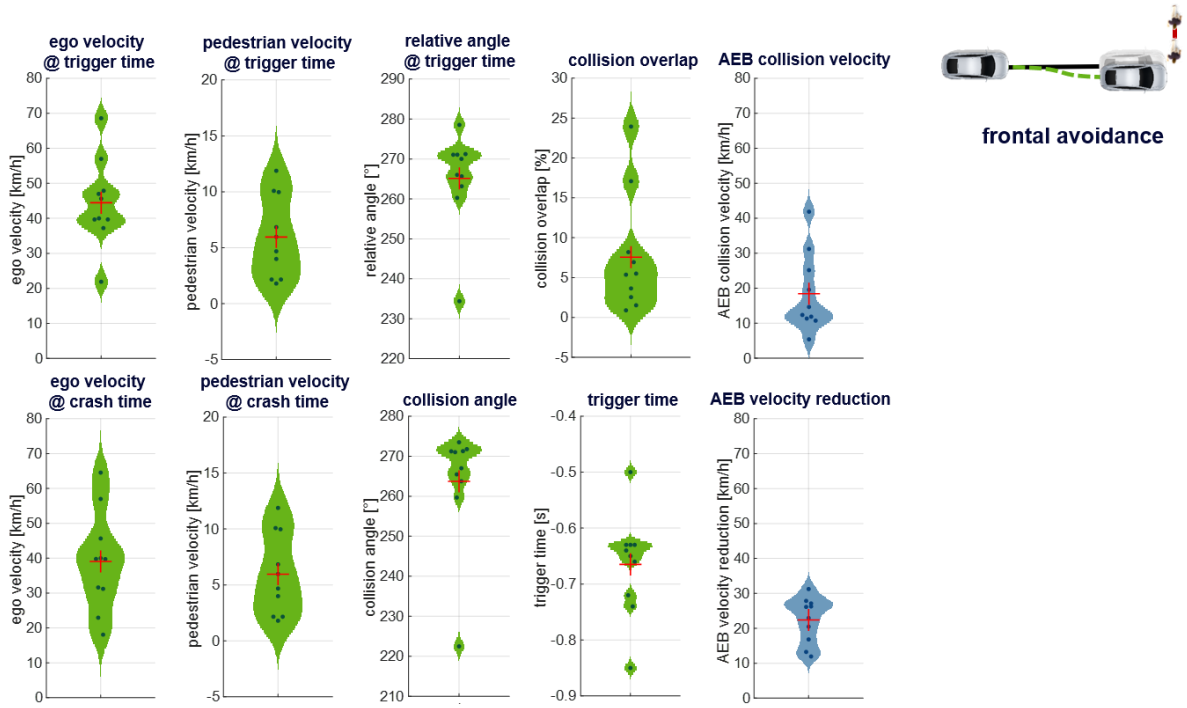

Figure 12: Parameter distributions for PCM cases with unique AES avoidance potential (green) and AEB mitigation potential (blue) using the *frontal collision avoidance* criterion.

With the *frontal collision avoidance* criterion, the overall AES field of effect increases by about 15%, while the percentage of AES avoidance only cases decreases by about 2%.

In four out of the ten relevant cases, frontal collisions are converted into side collisions. These cases enlarge the potential field of effect of the AES system, as they do not occur in the evaluation using the *full collision avoidance* criterion. In the remaining six cases, the frontal collisions can be fully avoided.

The distribution of hitpoints in the original accident is shown in Figure 13.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

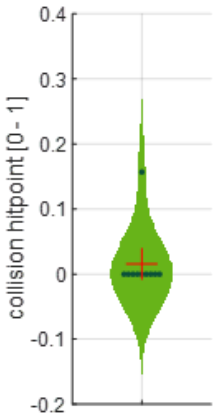

Figure 13: Distribution of original accident hitpoints in PCM cases with unique AES avoidance potential using the *frontal collision avoidance* criterion.

In conclusion, accidents of the three UTPYs 401, 431, and 461 with avoidance potential using the aforementioned AES system are characterized by:

1. Hitpoints on the front half of ego vehicle's left side,
2. Hitpoints on the leftmost part of the ego vehicle's front,
3. Initial ego vehicle velocities in the range of about 30 km/h to 60 km/h,
4. Initial pedestrian velocities in the range of about 2 km/h to 10 km/h and
5. Collision angles in the range of 260° to 280°.

3.2.2 Scenario description

Based on the simulation results the following scenario specifications for the first iteration of the Demo 3 AES development are chosen:

1. Ego vehicle velocity: 40 km/h, 50 km/h, 60 km/h,
2. Pedestrian velocity: 3 km/h, 6 km/h, 9 km/h,
3. Collision angle: 90° and
4. Collision hitpoint on the ego vehicle's front: 0, 0.2, 0.4.

Figure 14 shows an illustration of those first iteration scenarios. An obstruction of the pedestrian (black solid line) will be used for better demonstrability of the criticality of the accident scenarios and the resulting of rather late system trigger times.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

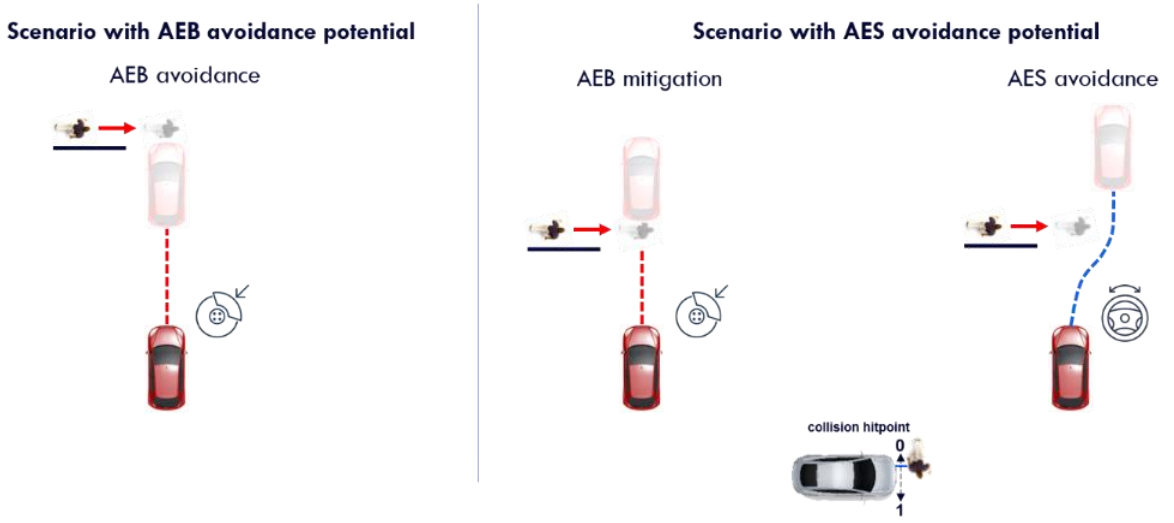

Figure 14: Illustration of the Demo 3 scenarios chosen for the first iteration of the demonstrator.

3.3 Applicability for future scenarios for Demo 3

The general method of scenario selection can be applied on any other data base or scenario set. So when definitions of future accident scenarios become available (SAFE-UP, Deliverable report D2.8, 2021) (SAFE-UP, Deliverable report D2.13, 2021), the scenarios that could be addressed by Demo 3 can be extended.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

4. Demo 3 development status

4.1 Overall demonstrator scope

In the first iteration of Demo 3 the focus will be on demonstrating the basic functionality of all algorithms and the whole algorithm chain. The general layout of the software is shown in Figure 2.

System integration and functional verification of the whole architecture will be done using the scenarios described in chapter 3.2.2. An extensive analysis of further relevant scenarios for Demo 3 based on the results of D2.6 (SAFE-UP, Deliverable report D2.6, 2021) using the scenario selection method described in section 3.1 will be performed in the further course of the project.

Preliminaries:

The terminology used within this report can have in literature slightly different interpretations. To avoid confusion, this paragraph defines the terminology and ontology used in this chapter, which is an extension on the review article of (Laurène Claussmann, 2020) where widely accepted terminologies are explained.
In a general hierarchical scheme of automated vehicles (see Figure 15) once the high level route and decision are known, the motion strategy includes generating and selecting a *path* and a *trajectory.*

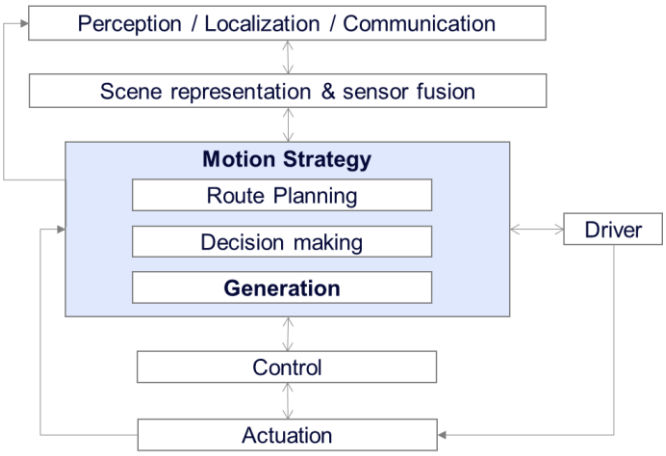

Figure 15: General hierarchical abstraction scheme of automated vehicles adopted from (Laurène Claussmann, 2020), where the generation block contains the path and trajectory generation.

Path here refers to the sequence of space-related states in the free space (also referred to as geometric waypoints) and *trajectory* refers to the sequence of spatiotemporal states in the free space (also referred to as time-varying waypoints).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

Furthermore, *maneuvers* refer to a predefined motion considered as a subspace of paths or trajectories, i.e. motion and actions or tasks refer to symbolic operations of maneuvers or intentions of road users.

4.2 Demonstrator subsystems

A general overview of the demonstrator subsystems is given in Figure 2. Detailed descriptions for each subsystem can be found in the following subsections.

With the goal of demonstrating the basic functionality of all algorithms and the whole algorithm chain within this first iteration of Demo 3, special focus is given on developing the most relevant subsystems. These are Path Planning, VRU Intent & Trajectory Prediction, Trajectory Generation and Crash Prediction & Avoidance Estimation, which will be described in more detail in the following sections. The subsystems Localization, Global Planning, Safety Decision and Vehicle Control are implemented in a way that the system works as a whole but kept rather simple. The subsystem Sensor Input and Object Fusion & Tracking mainly uses already available algorithms which are embedded into the radar sensor module.

4.2.1 Subsystem Localization

As localization is merely an enabling technology for the Demo 3 developments, validation and demonstration, the vehicle will be equipped with a high end GNSS-RTK sensor. This sensor will output high frequent and accurate GNSS data, which can be fused with odometry and/or IMU data if needed.

4.2.2 Subsystem Global Planning

Global goal points are provided from the global planning module for to the path planning module to plan towards. These goal points will be updated periodically and are always an approximate fixed distance away from the current vehicle position. The goal points can be extracted from a stored map (e.g. in OpenDrive format), or from a recorded GNSS trace. Besides the geometry point in a world fixed frame (e.g. UTM), the goal point can contain information on the threshold for which it is considered reached, or alternatively be a goal area, represented by a polygon.

Furthermore, the global planner is able to provide road information such as lane centers and/or boundaries, which can be used by the path planner module to find a better solution.

4.2.3 Subsystems Sensor Input and Object Fusion & Tracking

As Demo 2 covers the aspects of advanced sensor perception and Demo 3 is focused on the development of an evasive emergency maneuvering, the subsystems Sensor Input and Object Fusion & Tracking are realized using a single front-facing radar sensor module and

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

single front-facing video camera module with close-to-series hardware and software revisions.

Fusion of radar and video data is performed using radar locations (single reflexes detected by the radar detector) and video objects (object data derived from solely raw video signals by the camera module). Video objects are communicated by the camera module via a CAN FD bus to the radar sensor module. The sensor data fusion and object tracking algorithms are executed on hardware embedded into the radar sensor module.

Fusion objects (object data derived from the sensor data fusion and tracking algorithms) are then provided to the Localization and VRU Intent & Trajectory Prediction subsystems.

4.2.4 Subsystem Path Planning

To perform any automated driving task the basic idea is to safely traverse a state space to get from point A to point B. When given a driving goal by a route/global planner, if a path can be found then also sequentially a trajectory can be found that ensures the desired smoothness, continuity, comfort, speed and that obeys vehicles constraints.
The challenge of selecting a good path for automated driving tasks (e.g. lane changes, obstacle avoidance, car following, merging, etc..) can be solved with different algorithms each having their own advantages and disadvantages. The review by (D. González, 2016) distinguishes 4 main categories of motion planning algorithms: *graph search*, *sampling*, *interpolating* and *numerical optimization*. Other recent review papers, such as (Laurène Claussmann, 2020) highlight that depending on the architecture that is selected for motion planning and the scope of the automated vehicle, different algorithms have been used to find a feasible and optimal path for the vehicle.

Current challenges in real-time planning lie in dealing with dynamic environments in urban scenarios with multiple agents (i.e. pedestrians, cyclists, other vehicles) and dealing with extended ODDs (operational design domains), such as bad weather, complex road layouts, bad light and so on. This extra complexity requires the algorithms to become proactive rather than reactive, be robust to perception uncertainties and also be interaction-, risk- and context-aware.

To discover which algorithm performs better for the use cases considered here, two of the four classes of algorithms are implemented in this work: the *sampling based* and the *numerical optimization*. The sampling based planner is selected due to its wide usage by multiple previous research, its ability to deal with high dimensional spaces and allowing a fast planning in semi structured environments by executing a random search through the navigation area. The second algorithm selected, MPC (Model Predictive Control) was selected to investigate the currently existing trend of bringing more of the vehicle control tasks upwards towards the motion planner and to research if MPC can include predicted information of other road users and make the vehicle react more proactively. Compared to the sampling based algorithm which leads to suboptimal solutions, MPC has the potential to offer an optimal one. Both algorithms have the ability to consider non-holonomic constraints such as

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

vehicles' momentum and maximum turning radius and both algorithms consider the same inputs as shown in Figure 16.

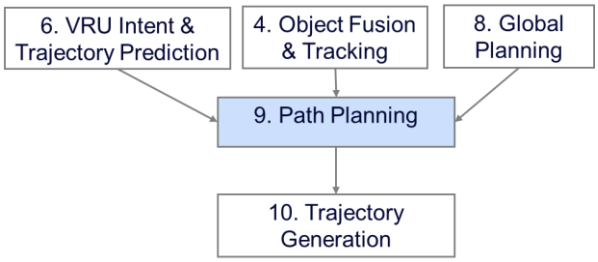

Figure 16: Path Planning subsystem (inter)dependencies.

4.2.4.1 Sampling-based Path Planning

Most sampling based planners sample a new state in the selected state space. Such state spaces can be $\mathbb{R}^2$, Dubin or Reeds-Shepp. However, planning in such space either puts a massive limit on the movement of the car (e.g. the constant curve radius in Dubin) or does not necessarily satisfy the differential and/or kinematic constraints of vehicle's motion model.

To facilitate the integration of kinematic and differential constraints while maintaining flexibility on the reachable space of the vehicle, action space sampling is an alternative. Contrary to state space sampling, action space sampling (alternatively called control space sampling) samples inputs to a dynamical system. Integrating this control input over a specified time interval given a fixed initial state (sample) will then result in a new state space sample, which satisfies the kinematic constraints. In this case, the following state and input are used:

$$X = [x\ y\ \theta\ v_{linear}\ t]^T,$$

$$U = [a_{linear}\ \varphi]^T,$$

with x, y being the positions in a world-fixed frame, θ the orientation in a world fixed frame, v_{linear} and a_{linear} respectively the forward speed and acceleration, φ the steering angle and t a (reference) time. For the forward propagation of the dynamic system, the following non-linear kinematic bicycle model discrete-time system is used:

$$X_{t+\Delta t} = f(X_t, U_t, \Delta t),$$

with

$$f(X_t, U_t, \Delta t) = X_t + \begin{bmatrix} v_{linear}\cos(\theta) \\ v_{linear}\sin(\theta) \\ \frac{v_{linear}}{wheel\ base} * \tan(\varphi) \\ a_{linear} \\ 1 \end{bmatrix} \Delta t.$$

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

The control inputs are sampled based on a Gaussian distribution with a parametrizable mean and standard deviation.

First, the propagated state is checked for validity based on free space and collision with other targets (VRU's, vehicles). If feasibly, the states are awarded a penalty based on several, parametrizable, objectives such as:

- Desired speed,
- Path length,
- Driveable space,
- Distance to targets

and added to the tree. If multiple solutions are found from start- to end state, the solution is chosen with the lowest cumulative cost. The full algorithm is described in *(Yanbo Li, Zakary Littlefield, Kostas E. Bekris, Sampling-based Asymptotically Optimal Sampling-based Kinodynamic Planning, 2014).* For the implementation of this planner, the Open Motion Planning Library (OMPL) is used *(Ioan A. Şucan, Mark Moll, Lydia E. Kavraki, The Open Motion Planning Library,* IEEE Robotics & Automation Magazine, *19(4):72–82, December 2012. **https://ompl.kavrakilab.org**).*

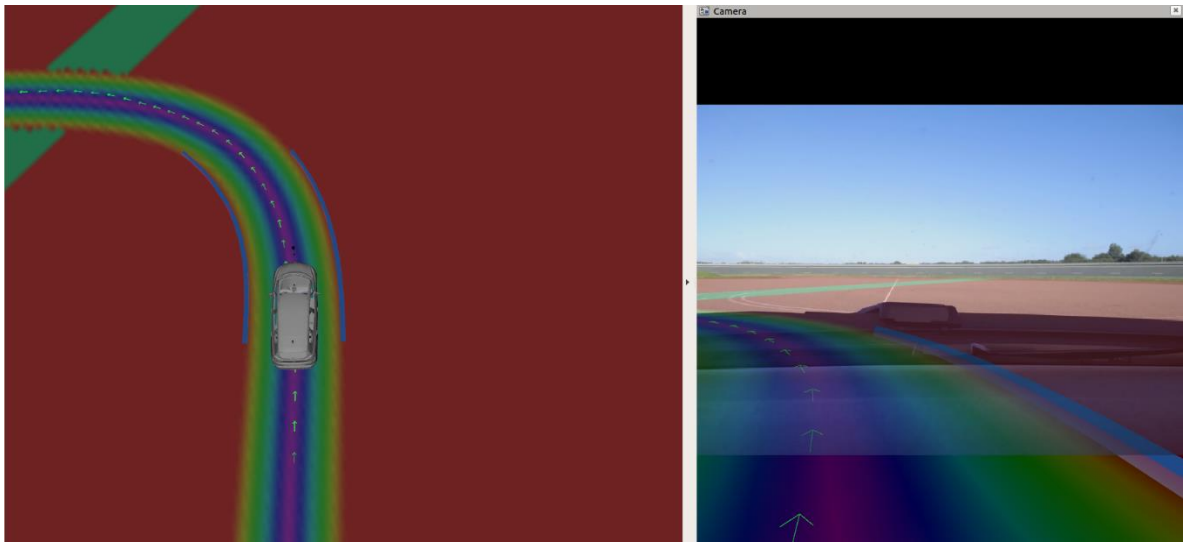

Figure 17: Sample based planner result.

Figure 17 shows a visual representation of the path planner output with the vehicle negotiating a left turn. Here a penalty grid map based on driveable space (following from the lane geometry) is shown, together with the virtual lane markings (in blue) and resulting path (green arrows).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

4.2.4.2 MPC-based Path Planning

Parallel to the sample based path planners, research is conducted involving the design of a local planner using a model-predictive control (MPC) algorithm. As opposed to a trajectory generator using MPC, the MPC formulation for a local planner usually looks slightly different since the goal is not to follow a certain trajectory, defined at each time-step, but to follow a discrete set of points generated by a behavior planner (e.g., using a Dijkstra algorithm). These points are usually spaced at a low resolution and primarily provide the information to, e.g., take a left turn or to keep driving straight. These discrete points work well in a scenario where the environment is unstructured and you simply need an algorithm to move from point A to B, collision-free, while satisfying constraints.

An example in which this information is not sufficient is the scenario in which you are expected to follow a lane (not necessarily straight), or have the possibility to follow multiple lanes. The exact reference to the center of the lane, or even which lane to drive on based on surrounding obstacle positions is not something which can be captured in these low-resolution positions.

On the public road, a driver and its vehicle is expected to always follow the lanes and respect regulations. For this purpose, many research papers have augmented the problem with a potential field or a "risk-assessment" field. This field captures analytically the risk of driving at certain positions. For example, driving off-center in an empty lane poses a higher risk than simply driving in the center. Using this risk-assessment field, the vehicle can guide itself through the environment in a potentially optimal way, up to the goal point.

The MPC formulation of the path planner is formulated as follows:

$$J^*_{0\to N_p}(X_s)=\min_u \sum_{k=0}^{N_p} l_X(X_k-r_k)+\sum_{k=0}^{N_c} l_u(u_k)+m\left(x_{N_{p+1}}-r_{N_{p+1}}\right)+U(X_k,o_k)$$

$$s.t. X_0 = X_s,$$

$$X_{k+1} = f(X_k, u_k), \forall k \geq 0,$$

$$h(X_k, u_k) \leq 0, \forall k \geq 0,$$

$$g(X_k, u_k, o_k) \leq 0, \forall k \geq 0,$$

where X_k and u_k represent the same state and input vectors as the sampling based planner, functions $g(.)$ and $h(.)$ represent constraint functions, $U(.)$ represents the artificial potential field function which could depend on state X_k and objects o_k, l_X and l_u are the stage cost of the state w.r.t. reference r_k and input respectively, $m(.)$ represents the terminal costs and N_p is the prediction horizon. The model for MPC synthesis considered here is the same non-linear kinematic bicycle model as used in the sample based planner.

Note, that the stage cost l_X may not be used due to an absence of a trajectory to follow at each time-step. The terminal cost, however, is used and is compared to the low-resolution discrete-points, in this case $r_{N_{p+1}}$. Moreover, the constraint $g(X_k, u_k, o_k) \leq 0$ may not be used due to the fact that the potential field could incorporate the presence of surrounding objects.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

This reduces the complexity at the cost of it becoming more difficult to guarantee a collision-free trajectory. The potential field is composed of 2 parts, one representing the infrastructural potential field (i.e. the road lanes) and one representing a potential field taking into account static and/or dynamic objects and predictions.

4.2.5 Subsystem VRU Intent & Trajectory Prediction

The ability of an automated vehicle to accurately sense its surroundings and anticipate how the driving environment will evolve is crucial for road safety. If the vehicle is fully aware of the current and future positions of all static (e.g. road infrastructure) and dynamic (e.g. other road users) objects, it can plan its own trajectory optimally and prevent any hazardous situations.

The task predicting the future positions of road users can be solved with different algorithms. These algorithms can be classified along two main dimensions: *(i)* the choice of modeling, and *(ii)* the contextual information they use (see Figure 18). The different modeling approaches have their own strengths and weaknesses. *Physics-based* approaches are accurate for short prediction horizons, but they don't capture the complexity of real-world dependencies and therefore their performance degrades rapidly for longer horizons.

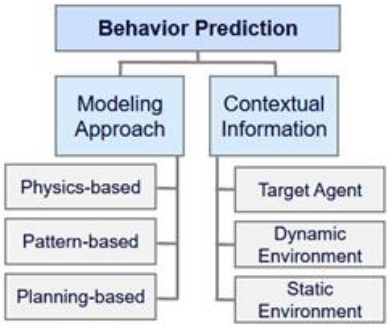

Figure 18: Taxonomy of behavior prediction models. Adapted from (Rudenko, et al., 2020).

Pattern-based approaches can approximate complex behavior without prior expert's domain knowledge, but if some behavior is not present in the data used to develop them, they will perform poorly on those situations. Finally, *planning-based* approaches capture well the goal-oriented nature of humans, but their running time scales exponentially with the number of objects in the scene, and automatic goal inference requires a complex understanding of the driving environment.

In recent years pattern-based prediction models have gained popularity due to their performance in a variety of applications (Lecun, Bengio, & Hinton, 2015). In particular for trajectory prediction, recurrent neural networks (RNNs) are at the center of most state-of-the-art methods (Rasouli, 2020). Long short-term memory (LSTM) is one of such RNN architectures which is commonly chosen due to its ability to capture temporal dependencies (Hochreiter & Schmidhuber, 1997). Thus, this architecture is also chosen for this submodule. Additionally, the more contextual information considered in the predictions, the more accurate the predictions tend to get. However, the required information is not always available or

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

accurate while driving, and in these cases the predictions of pattern-based models become unpredictable, so it is vital to have other simpler, fail-safe models that are at least reliable for short prediction horizons.

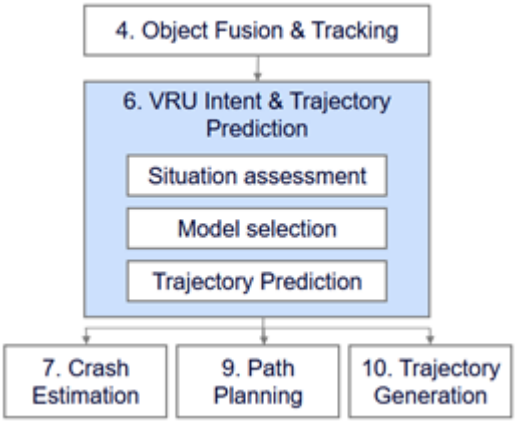

Figure 19: Overview of the VRU Intent & Trajectory Prediction subsystems (inter)dependencies.

To that end, this subsystem has been developed as overviewed in Figure 19. The inputs considered result from the object fusion & tracking module. An initial assessment is performed to verify if sufficient information is available for the implemented pattern-based model. If this information is not available (e.g. missing past target trajectory due to a late detection), the future positions are predicted with a constant velocity model, which is shown to describe pedestrian motion well in nominal conditions (Scholler, Aravantinos, Lay, & Knoll, 2020). The implemented pattern-based model is detailed next.

Pattern-based model – LSTM Autoencoder

The selected architecture for the first choice of prediction model is an LSTM Autoencoder, a flexible architecture allowing inputs and outputs of varying type and size.

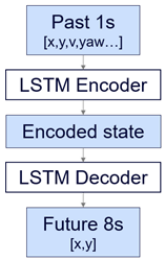

Figure 20: Overview of the LSTM autoencoder architecture.

Figure 20 provides an overview of the model's currently implemented architecture. Its inputs are 1 second of past states (positions, velocities, headings, etc.) of target(s) to be predicted, and its outputs are the target's estimated positions for the next 8 seconds. In future iterations,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

the model will also consider as input the static environment (i.e. road infrastructure) and dynamic environment (i.e. other road users).

To train and evaluate the implemented LSTM Autoencoder, the Waymo Open Motion Dataset was considered (Ettinger, et al., 2021). This dataset contains hundreds of thousands of annotated trajectories for three types of road users: pedestrians, cyclists, and vehicles.

4.2.6 Subsystem Trajectory Generation

The Trajectory Generation subsystem is responsible for the computation of vehicle trajectories (spatiotemporal information about the vehicles future motion) on the basis of the information provided by the Path Planning and VRU Intent & Trajectory Prediction subsystems. Vehicle trajectories are evaluated regarding a risk estimation associated with their potential realization by the Crash Prediction & Avoidance Estimation subsystem and handed over to the Safety Decision subsystem (see Figure 21). The Trajectory Generation subsystem constitutes of three subsystems of its own: *Nominal driving trajectory generation*, *AEB trajectory generation* and *AES trajectory generation*.

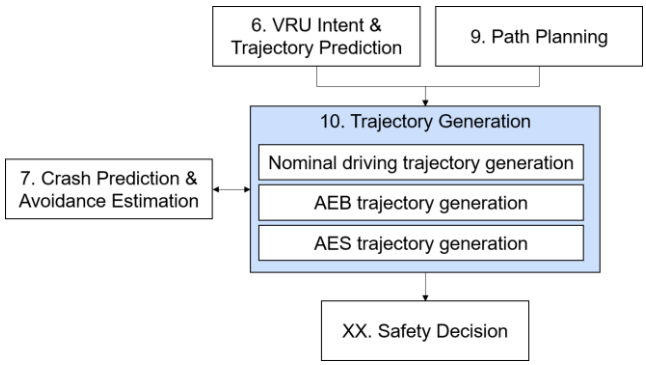

Figure 21: Overview of the Trajectory Generation (inter)dependencies.

Nominal driving trajectory generation is used to compute the vehicle's trajectories for driving under nominal conditions when there is no need for any emergency maneuvering and the driving goal is to follow the path planned by the Path Planning subsystem (see Figure 22). Here the trajectory generation utilizes an inverted stationary state approximation of the linear single-track model (Schramm, Hiller, & Bardini, 2010) to calculate a path-following trajectory as well as the control command for the vehicle's steering system based on the vehicles current velocity (assuming constant velocity for future motion) and the path information.

AEB trajectory generation performs the calculation of vehicle trajectories for an automated emergency braking maneuver (see Figure 22). Trajectories are generated using the same model inversion as used by the *Nominal driving trajectory generation*, but employs a fixed deceleration profile for the computation of the vehicles future motion. Thus, AEB trajectories are planned to be path-following with a deceleration of the vehicle until stand-still. For

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

determination of the spatiotemporal starting point of an emergency braking maneuver an adaptive sampling of starting times for the computation of the vehicle's deceleration is used. Trajectories are iteratively sampled with starting times chosen from the interval of the time of trajectory calculation to the time of a predicted collision with a target object (VRU) when the vehicle would follow the *Nominal driving trajectory.* For each iteration the sampled trajectories are handed over to the Crash Prediction & Avoidance Estimation subsystem for a collision and risk estimation, that is used to determine the sampling of starting times for successive iterations.

AES trajectory generation computes vehicle trajectories for an automated emergency steering maneuver (see Figure 22). Trajectories are generated by adaptively sampling deviations from the path planned by the Path Planning subsystem. Deviations are given in terms of deviation profiles that are generated using a switched feedback controller topology (Sira-Ramirez & Agrawal, 2004), that uses an extended linear single-track model (Uhler, 2021) to take higher (compared to nominal driving) lateral vehicle dynamics into account and features the possibility of an intrinsic handling of state variable and model input constraints (Joos, Bitzer, Karrelmeyer, & Graichen, 2018). The latter allows to consider external (e.g. legal, safety, human-factors) and system constraints at the level of trajectory planning which guarantees the feasibility of all planned maneuvers. The deviation profiles thereby maximize the usage of the vehicle dynamics potential within the constraints given by an emergency situation. The degrees of freedom of the sampling of deviation profiles are their spatiotemporal starting point and maximum lateral deviation. Profiles are iteratively sampled with starting times chosen from the interval of the time of trajectory calculation to the time of a predicted collision with a target object (VRU) when the vehicle would follow the *Nominal driving trajectory* and maximum lateral deviations chosen from the interval of vehicle overlap with a target object when the vehicle would follow the *Nominal driving trajectory* to the maximum possible lateral deviation (given by lane boundaries\legal constraints). As for *AEB trajectory generation* sampled trajectories of each iteration are handed over to the Crash Prediction & Avoidance Estimation subsystem for a collision and risk estimation to determine the sampling for successive iterations.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

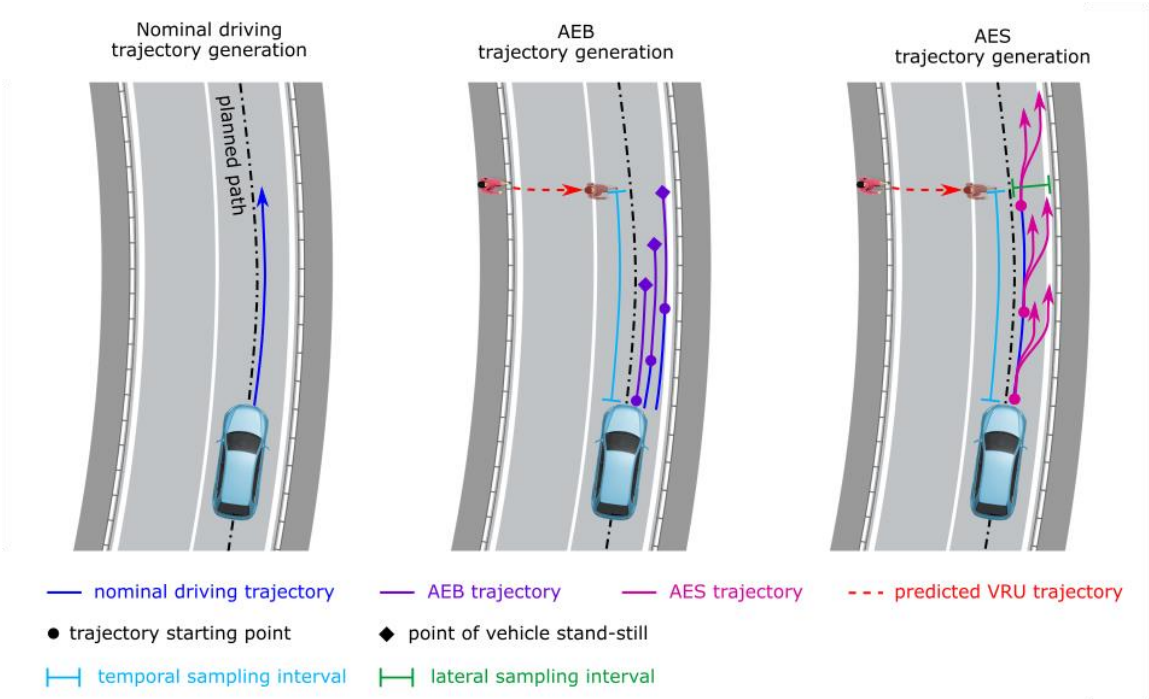

Figure 22: Overview of the trajectory generation by the *Nominal driving trajectory generation*, *AEB trajectory generation* and *AES trajectory generation* subsystems. Planned trajectories are displayed with offsets to the planned path for clarity. Details see text.

Iterative sampling of AEB and AES trajectories is terminated either by reaching the sampling tolerances of the respective subsystems or the cycle time of environmental information updates as provided by the Object Fusion & Tracking subsystem. The final sets of trajectories generated by the *Nominal driving trajectory generation*, *AEB trajectory generation* and *AES trajectory generation* subsystems and their collision and risk estimates are then handed over to the Safety Decision subsystem.

4.2.7 Subsystem Crash Prediction & Avoidance Estimation

The crash prediction and avoidance estimation are based on a risk assessment for a set of ego and object trajectories. Therefore, the inputs for this subsystem are an ego trajectory and the predicted trajectories of dynamic objects. Note that static objects are not considered since the ego trajectories are already collision-free for those. The output is a value of risk for the ego trajectory. Risk is defined as the product of collision severity and collision probability. The risk function is based on (Wang, Wu, Zheng, Ni, & Li, 2016). Due to the computational complexity of field-based functions, the work is adapted to evaluate single points on trajectories, reducing the computational complexity by multiple magnitudes.

The algorithm uses trajectories of the form $t \in [t_0, t_f] \times (x, y, \theta, v)$, where t_0 and t_f are the start and end times of the trajectory. The position is denoted as (x, y), and the heading angle is θ. Lastly, v denotes the velocity. From this information, a distance-vector $\vec{r}(t) = (x_{\text{ego}}(t) -$

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

$x_{\text{obj}}(t), y_{\text{ego}}(t) - y_{\text{obj}}(t))$ is computed. The vector represents the distance of points of the trajectory at the same time instances. Figure 23 depicts this pointwise comparison.

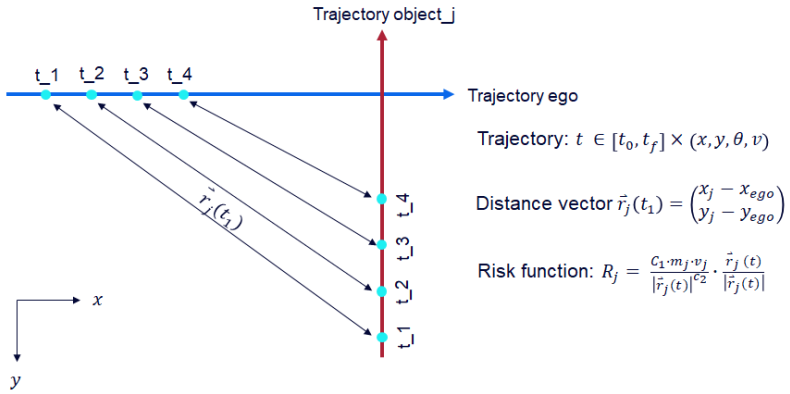

Figure 23: Illustration of distance vector calculations.

After computing the distance-vector, the risk is evaluated for each distance-vector. Since the risk function depends on the inverse absolute value of the distance-vector, the risk increases towards infinity when both trajectories intersect at the same time.

Since thus far only points are compared in the time domain, it is unlikely to detect a collision due to points not having a size. To ensure the computational feasibility, the object sizes are approximated by circles as (Ziegler & Stiller, 2010). As depicted in Figure 24, three circles approximate vehicles, and VRU's are approximated with one circle.

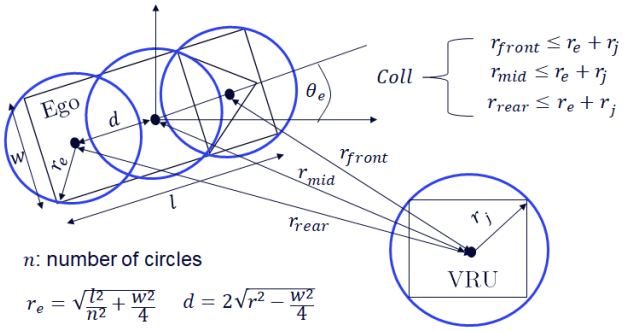

Figure 24: Illustration of circular bounding box estimations.

The collision function detects a collision if and only if the sum of the radii of one circle of the ego and the circle of the object is less or equal to the distance vector between both center points. Having three circles in the ego vehicle enables the risk function to distinguish between a front, mid and rear collision. Therefore, a factor is introduced that can model the differences in collision severity of each case.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

The chosen risk assessment methodology ensures that a subsequent decision module can make a reasonable decision based on potential harm and closeness to objects. Further, it can compute where and when a collision occurs.

4.2.8 Subsystem Safety Decision

The subsystem safety decision is implemented to decide between different motion alternatives. The sampling-based planner provides the motion alternatives (see Figure 22). For the first implementation of the proposed system design, the decision will be solely based on the risk value assigned to each trajectory by the crash prediction and avoidance estimation module (see Figure 21). Still, this subsystem will be designed such that it can be easily extended to include more information for a decision process.

4.2.9 Subsystem Vehicle Control

Vehicle control is executed via the vehicles steering and braking system. Steering and braking command references corresponding to the planned maneuvers (nominal driving, emergency braking, emergency steering) are generated by respective trajectory generation subsystems. These command references are used to directly control vehicle dynamics by means of feedforward control only, without a feedback path on short timescales. On longer timescales the cyclic trajectory replanning acts as a feedback path for vehicle control by taking updates of the vehicle state as well as environmental information into account. Due to the short timescales of higher (than nominal driving) dynamics during an emergency maneuver, errors in the vehicle control that accumulate within one trajectory planning cycle are estimated to be negligible for the first iteration of Demo 3. A dedicated short timescale feedback path for vehicle control is planned for the second iteration of Demo 3.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

5. Initial test results for Demo 3

As described in section 4.2, the goal of this first iteration of Demo 3 is to demonstrate the basic functionality of the whole algorithm chain with special focus on the subsystems Path Planning, VRU Intent & Trajectory Prediction, Trajectory Generation and Crash Prediction & Avoidance Estimation.

Initial test results for each of these subsystems are described in the following.

5.1 Initial test results Path Planning

The path planning algorithms are tested and benchmarked on one of the TNO Carlab vehicles, which are shown in Figure 25. To be able to test the system in closed-loop, dummy modules are created for the global planner and vehicle controller. The localization and object tracking & fusion modules are taken from TNO background IP.

Figure 25: TNO Carlabs.

The tests are performed at the RDW test track in Lelystad, the Netherlands. First, several virtual routes are recorded in an offline run representing different road layouts, e.g. straight road, corner, roundabout and intersection. From these virtual routes, virtual lanes and road layouts are created, as if there were physical lines on the asphalt. These are sent to the vehicle to mock the input from e.g. a road camera sensor.

Then, the path planner is enabled together with the other modules in the closed loop system and these different road layouts are navigated in closed loop. The results are recorded and stored for post-processing and the algorithms are evaluated based on several safety and comfort criteria, such as deviation from lane center and acceleration.

A sample of the results of the TNO Carlab negotiating an oval track is shown in Figure 26 for the sample based Path Planner and in Figure 27 for the MPC based path planner. Here, the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

vehicle is controlled and actuated based on the path planner outputs. In these figures, some relevant KPI's are shown (e.g. deviation from lane center).

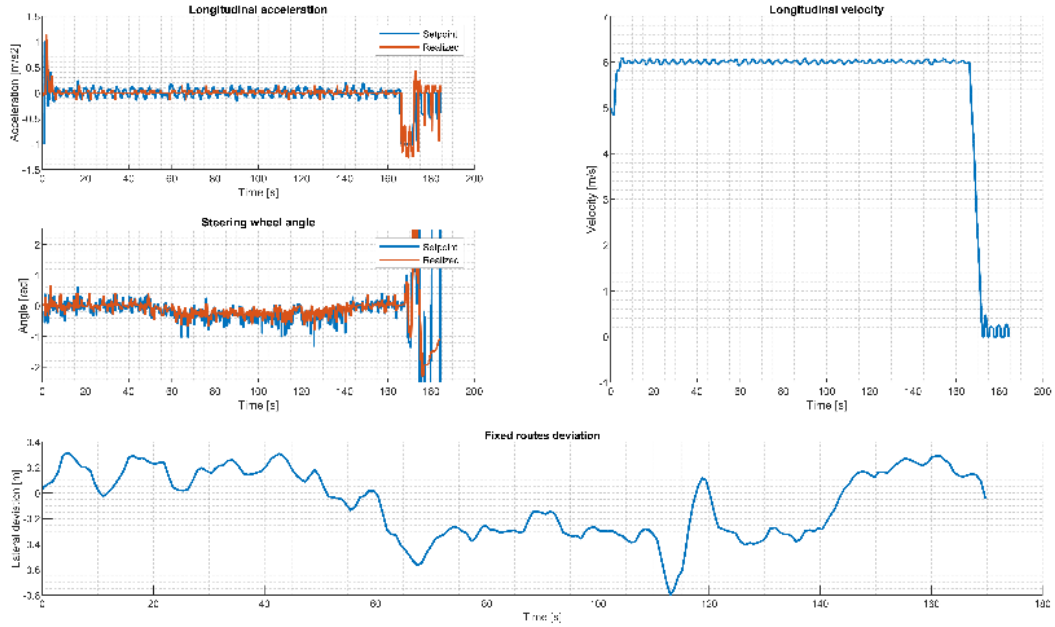

Figure 26: Sample of the TNO sample based path planning in closed loop.

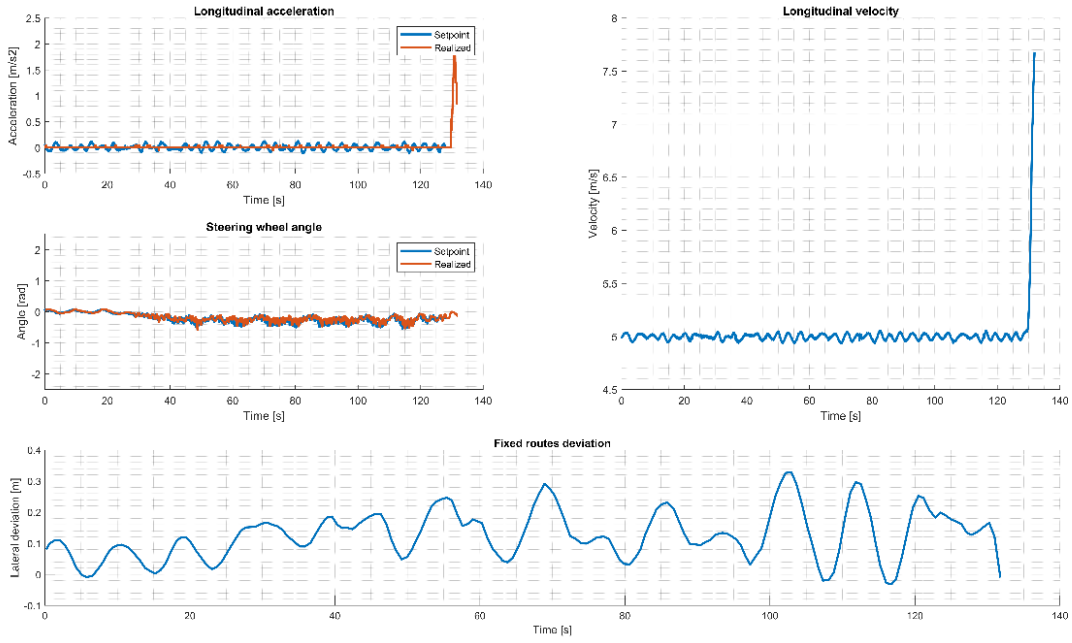

Figure 27: Sample of the TNO MPC based path planning in closed loop.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

5.2 Initial test results VRU Intent & Trajectory Prediction

A preliminary evaluation of the developed LSTM's predictive accuracy was performed using a part of the Waymo dataset reserved for this purpose (i.e. these trajectories were not used during model development), and a constant velocity model is used as a baseline for model comparison.

Figure 28 shows the final displacement error (FDE) of the predicted trajectory with respect to the actual trajectory for approximately 1 second in the future. As can be seen in the figure, the constant velocity model is more reliable for very short prediction horizons (i.e. <0.2 seconds), and after that point the LSTM results in a lower error. For predictions of up to one 1 second, both models present a mean error lower than 0.2 meters, although the variability of this error is considerably lower with the LSTM.

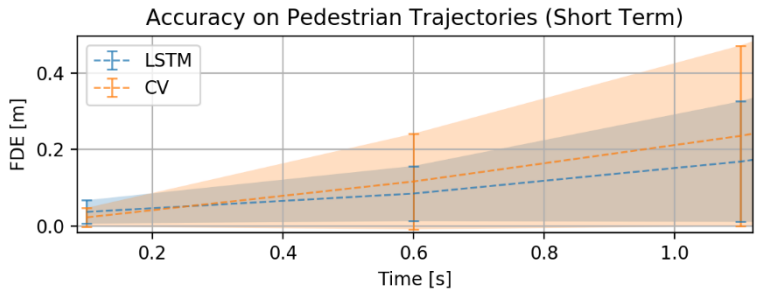

Figure 28: Short-term prediction accuracy of the LSTM and CV models on pedestrian trajectories. Vertical bars denote standard deviation of the errors.

Figure 29 depicts the same information for predictions of up to 8 seconds. Overall, the LSTM presents slightly superior performance in terms of accuracy and variability of the errors in the predictions. However, the benefits of the LSTM over a simple CV model are not directly visible from this model evaluation. The reason for such a minute improvement is the fact that most recorded pedestrian trajectories follow a straight path with approximately constant velocity. In other words, a simple CV model describes pedestrian motion reasonably well in most cases, and the advantages of more sophisticated models are most noticeable when a pedestrian suddenly turns, accelerates or decelerates.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

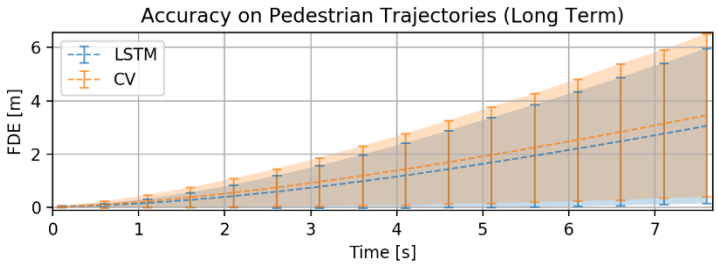

Figure 29: Long-term prediction accuracy of the LSTM and CV models on pedestrian trajectories. Vertical bars denote standard deviation of the errors.

Figure 30 depicts example predictions of both the CV and LSTM models for a crossing pedestrian that performs a minor change in direction. The LSTM is able to anticipate this minor turn and adapt its predictions accordingly, resulting in significantly higher predictive accuracy.

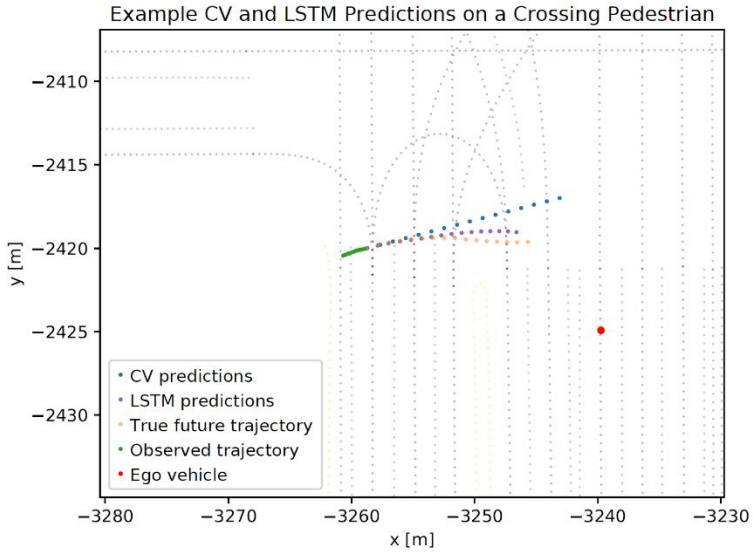

Figure 30: Example predictions on a scene where a pedestrian is crossing in front of the ego vehicle.

5.3 Initial test results Trajectory Generation

In the first iteration of Demo 3, special testing focus has been given to the AES trajectory generator, while testing of both the nominal and AEB trajectory remains open. For an initial evaluation regarding feasibility and potential unforeseen limitations, the AES trajectory generator has been tested within both a simulation framework and the demo 3 in-vehicle integration platform.

The simulation testing has the purpose of verifying and validating both the base trajectory generation algorithm, generating one trajectory from a predefined target lateral displacement

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

and predefined settling dynamics considering dynamic constraints, and the sampling algorithm, sampling target lateral displacement values and trajectory start times.

Figure 31 shows an exemplary case with a crossing pedestrian from the left side of the road, according to the defined scenarios in section 3.2.2. The ego vehicle is depicted as a black rectangle at the current timestamp, one second before a collision. The pedestrian is depicted as a red rectangle at the predicted collision timestamp, leading to a minimum required lateral displacement of 0.4m. Note that the x and y axes are not scaled equally for a better visibility of the trajectories, leading to a distortion of the objects. The right lane border is included to the figure as a dashed red line. Sampled trajectories are shown in blue. The sampling space in lateral direction is defined from the minimum required lateral displacement to the maximum possible lateral displacement, delimited by the right lane border. The sampling space for the trajectory starting time is given by the current timestamp and the collision timestamp. Trajectory sampling is activated when a collision is predicted by the crash prediction subsystem. Note that the length of each trajectory is automatically given by the predefined settling dynamics of the base trajectory generation algorithm. For this simulation example, five samples for both the sampling in lateral and longitudinal direction have been chosen, leading to a total sum of 25 trajectories. The specific starting times and lateral displacement values are displayed as black dashed lines. The collision timestamp is marked in red. Note that trajectory sampling will be repeated with each object data update, ensuring a fast adaptation to changes in the situation as well as adaptations due to prediction uncertainties.

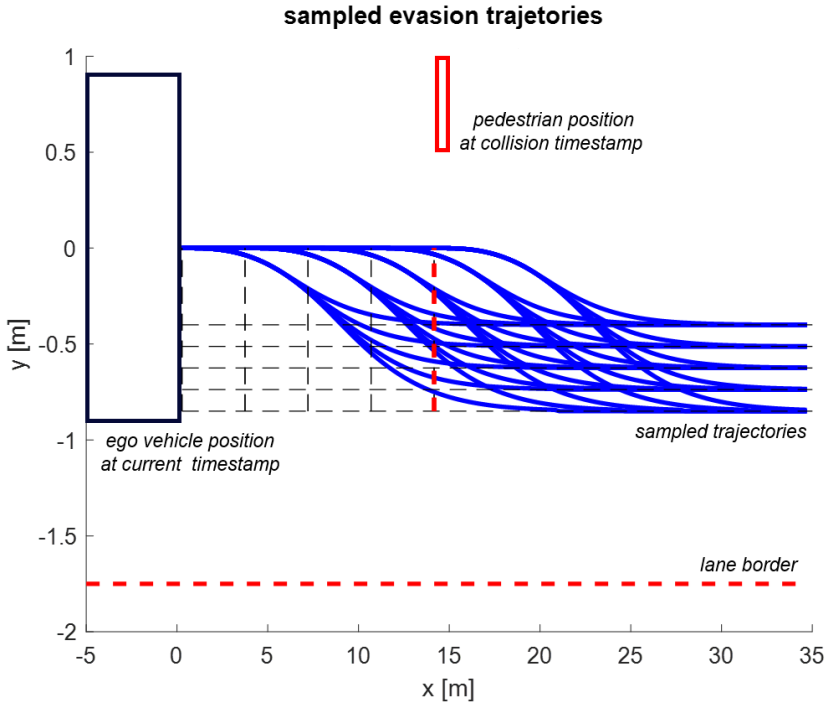

Figure 31: AES Trajectory Generation example in a crossing pedestrian case. The pedestrian position is displayed at the collision timestamp, while the ego vehicle position is displayed at the current timestamp. Sampled trajectories are displayed in blue.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

The simulation results show that trajectories are generated within the defined sampling space, considering the dynamic constraints as specified.

The in-vehicle testing has the purpose of tuning the trajectory dynamics in a way that the steering actuator can follow the required steering angle command. Furthermore, the steering actuator's dynamic constraints are determined and incorporated into the trajectory generation base algorithm.

Figure 32 shows the result of one planned AES trajectory with a target lateral displacement of 1m, including all planned vehicle states from the enhanced linear single track model described in section 4.2.6. Measured vehicle states are show in magenta, in case this was possible. Because no GNSS based reference system was available at the time of testing, lateral displacement, yaw angle and slip angle could not be measured. For the lateral displacement and the yaw angle, an estimation has been performed based on an integration of the measured yaw rate, displayed in dash-dotted magenta. The yaw rate of the maneuver is limited to a maximum of 12 °/s to ensure controllability by the driver, shown in dashed black. As described in section 4.2.9, the maneuvers are executed by means of feedforward only, without closed-loop control on short timescales for the first iteration of Demo 3. Because of the model based nature of the used trajectory generator, a feedforward steering angle command can directly be calculated using the model equations.

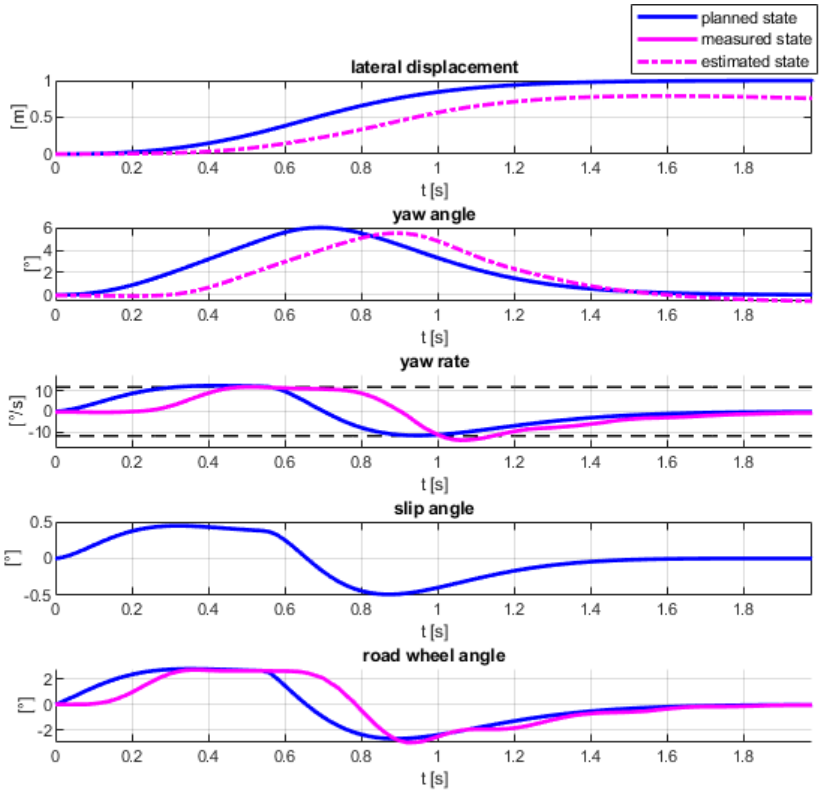

Figure 32: One planned AES trajectory, consisting of the displayed planned vehicle states in blue. The measured or estimated vehicle states are displayed in magenta.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

The in-vehicle testing results show a good following behavior of the steering actuator. A communication latency of ~100ms can be observed. Despite a short overshoot in the counter steering phase, the given yaw rate limitation can be fulfilled. The estimated lateral displacement shows values below the planned curve, leading to a stationary deviation of ~0.2m at the end of the maneuver. This is expected to be due to the fact that the slip angle of the vehicle is not considered for the lateral displacement estimation. Furthermore, a closed-loop trajectory controller could improve the trajectory following behavior.

The next step is the integration and test of the complete trajectory sampling algorithm within the demo 3 in-vehicle integration platform.

5.4 Initial test results Crash Prediction & Avoidance Estimation

Figure 33 shows two use cases: on the left side, no collision is occurring, but both trajectories come close to a collision. On the right side, a collision occurs, leading to the risk value's rapid growth.

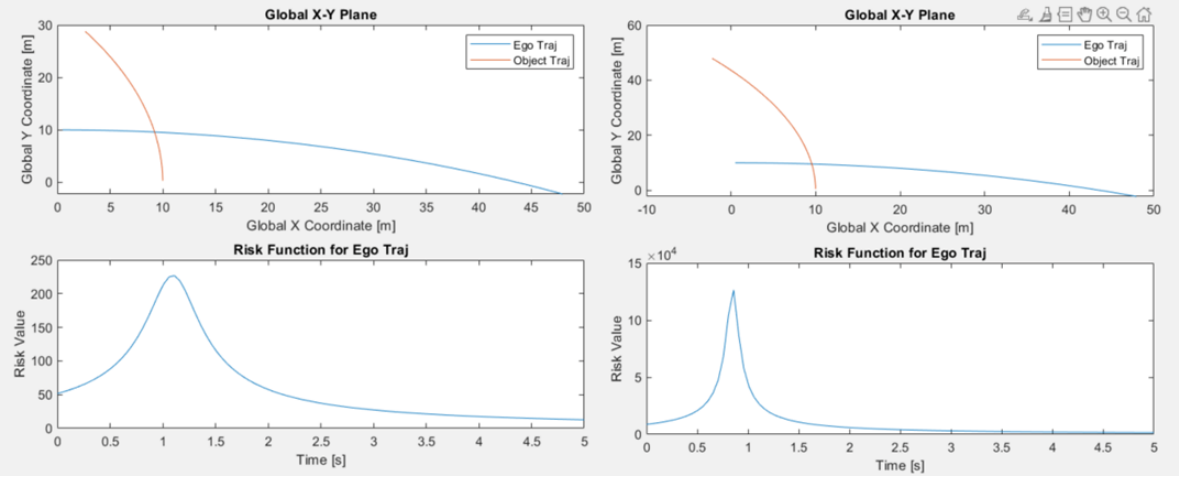

Figure 33: Evaluations of the time-resolved risk calculation for the use cases of no collision (left) and collision (right).

To return a single risk value for one ego trajectory, the accumulated risk value over time is computed. The inputs to the risk function are: One ego trajectory, one object trajectory (as described in 4.2.7) and the timewise length of the trajectory as well as the timewise step size for each discrete trajectory value. After executing the function, the risk function returns the accumulated risk value, if a collision occurs, where the collision occurs and when the collision occurs.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

6. Discussion, conclusions and next steps

In this first phase of the project several important development steps towards Demonstration 3 (Advanced Intervention Functions) have been done, which form a good basis for the previous and future developments to realize the active safety ambitions in WP3. These developments have started with the *high level vehicle design specification*, to *scenario selection*, *software architecture* and *interfaces definition* until *sub-system specification* and *design.*

Hereafter, all individual algorithms have been defined and developed in a first version for a subset of scenario variations. Each individual algorithm was evaluated in simulations and, when needed, through tests at each individual project partner. The first remote-integration of all software functionalities and interface verification has made an additional step towards the upcoming integration and in-vehicle testing.

The immediate next step will focus on applying, integrating, and porting all individual components to the Demo 3 vehicle. A set of (incrementally difficult) tests will be performed to ensure the desired integration and test the behavior according to the specified scenarios. After these first Demo 3 tests the algorithms will be further advanced and a new demonstration will be done in the next year.

The final system will be tested on the test track with VRU dummy systems under controlled environments with a subset of the selected scenarios, as well as in simulations with the complete set of scenarios.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

References

D. González, J. P. (2016). A Review of Motion Planning Techniques for Automated Vehicles . *IEEE Transactions on Intelligent Transportation Systems, 17*(4), 1135-1145 .

Ettinger, S., Cheng, S., Caine, B., Liu, C., Zhao, H., Pradhan, S., . . . Anguelov, D. (2021). Large Scale Interactive Motion Forecasting for Autonomous Driving : The Waymo Open Motion Dataset. Opgehaald van https://arxiv.org/abs/2104.10133

German In-Depth Accident Study. (sd). Opgehaald van https://www.gidas.org

Hamilton, I. A. (2019, 06 13). *Uber says people are bullying its self-driving cars with rude gestures and road rage.* Opgeroepen op 04 15, 2021, van Business Insider: https://tinyurl.com/pw7s7cs8

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation, 9*(8), 1735-1780. doi:10.1162/neco.1997.9.8.1735

Joos, S., Bitzer, M., Karrelmeyer, R., & Graichen, K. (2018). Priorization-based swiched feedback control for linear SISO systems with time-varying state and input constraints. *2018 European Control Conference (ECC)*, (pp. 2935-2940). doi:10.23919/ECC.2018.8550172

Laurène Claussmann, M. R. (2020). A Review of Motion Planning for Highway Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems, 21*(5), 1826-1248.

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*, 436-444. doi:10.1038/nature14539

Rasouli, A. (2020). Deep Learning for Vision-based Prediciton: A Survey. Opgehaald van arxiv.org: http://arxiv.org/abs/2007.00095

Rudenko, A., Palmieri, L., Herman, M., Kitani, K. M., Gavrila, D. M., & Arras, K. O. (2020). Human motion trajectory predition: a survey. *The International Journal of Robotics Research, 39(8)*, 895-935. doi:10.1177/0278364920917446

S. I. (2018). Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. Opgeroepen op 2020, van https://www.synopsys.com/automotive/autonomous-driving-levels.html

SAFE-UP. (2021). *Deliverable report D2.13.*

SAFE-UP. (2021). *Deliverable report D2.6.* Deliverable report.

SAFE-UP. (2021). *Deliverable report D2.8.*

SAFE-UP. (2021). *Deliverable report D3.1.* Deliverable report.

Schneider, N., Schmitz, M., Ahrens, L., Löffler, C., & Neukum, A. (2018). Der Toleranzansatz als Methodik zur Bewertung der Kontrollierbakeit höherer Eingriffsstärken bei

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.

Falschauslösung eines systeminitiierten Ausweichmanövers mit ESP-Aktorik. *12. Workshop Fahrerassistenzsysteme und automatisiertes Fahren* (pp. 67-78). Walting im Altmühltal: Uni-DAS e.V.

Scholler, C., Aravantinos, V., Lay, F., & Knoll, A. (2020). What te Constant Velocity Model Can Teach Us About Pedestrian Motion Prediction. *IEEE Robotics and Automation Letters, 5*(2), 1696-1703. doi:10.1109/LRA.2020.2969925

Schramm, D., Hiller, M., & Bardini, R. (2010). *Modellbildung und Simulation der Dynamik von Kraftfahrzeugen.* Heidelberg, Dordrecht, London, New York: Springer. doi:10.1007/978-3-540-89315-8

Sira-Ramirez, H., & Agrawal, S. K. (2004). *Differentially Flat Systems.* New York, Basel: Marcel Dekker, Inc.

Uhler, P. (2021). Trajektorienplanugnskonzepte für eine automatisierte Notausweichfunktion. Hochschule Heilbronn.

UNECE. (2018). Regulation No 79 of the Economic Commission for Europe of the United Nations (UN/ECE) — Uniform provisions concerning the approval of vehicles with regard to steering equipment [2018/1947]. *UN/ECE Regulation*. Opgehaald van http://data.europa.eu/eli/reg/2018/1947/oj

Wang, J., Wu, J., Zheng, X., Ni, D., & Li, K. (2016). Driving-safety field theory modeling and its application in pre-collision warning system. *Transportation Research Part C: Emerging Technologies, 72*, 306-324. doi:10.1016/j.trc.2016.10.003

Ziegler, J., & Stiller, C. (2010). Fast collision checking for intelligent vehicle motion planning. *2010 IEEE Intelligent Vehicles Symposion*, (pp. 518-522). La Jolla, CA, USA. doi:10.1109/IVS.2010.5547976

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 861570.