This project has received funding from the ECSEL Joint Undertaking under grant agreement No. 737469.

This Joint Undertaking receives support from the European Union's
Horizon 2020 research and innovation programme and the ECSEL member states.

Auto Drive

Deliverable	Report on integration of the stochastic traffic simulation		
Involved SCs	SC10	Deliverable type	Confidential
Project	AutoDrive	Grant Agreement Number	737469
Deliverable File	D5.13	Last Modified	22.06.2020 09:46
Due Date	30.06.2020	Actual Submission Date	
Status	Final	Version	1.0
Contact Person	Marcus Mai	Organisation	TUD
Phone	+49 351 463 34058	E-Mail	marcus.mai@tu-dresden.de

Document history			
V	Date	Author	Description
0.2	29.05.2020	Christian Siebke (TUD)	Initial Version
0.4	10.06.2020	Marcus Mai (TUD)	Review 1
0.6	10.06.2020	Marcus Mai (TUD)	Rework
0.8	22.06.2020	Felix Elrod (AVLS)	Review 2
1.0	22.06.2020	Marcus Mai (TUD)	Final Version

Publishable executive summary

As part of the AutoDrive project, the OpenPASS framework is used to develop a cognitive-stochastic traffic flow simulation for urban intersection scenarios described in deliverable D1.14. This framework was adapted and further developed.

The deliverable D5.13 deals with the construction of the stochastic traffic simulation. At this point of the process, the theoretical design aspects of D4.20 are implemented. D5.13 explains the operating principles of the different modules. This includes the foundations, boundary conditions, and mathematical theory of the traffic simulation.

Table of contents

Publishable executive summary.....	3
1 Introduction.....	5
2 OpenDRIVE	6
3 Construction of the stochastic traffic simulation	7
3.1 Perception	7
3.2 Gaze determination.....	8
3.3 Cognitive map.....	9
3.4 Localisation.....	9
3.5 Navigation	10
3.6 Right of way determination.....	11
3.7 Collision detection.....	12
3.8 Action decision	12
3.9 Lateral controller	13
3.10 C-AEB system.....	13
4 Outlook.....	14
5 List of figures	15
6 Bibliography.....	16

1 Introduction

Within the AutoDrive project, OpenPASS is used to develop a stochastic traffic flow simulation for urban intersections, which is described in deliverable D1.14. The stochastic traffic flow simulation is supposed to cover the entire range of traffic events, from normal driving to critical situations up to road accidents. The traffic flow simulation is based on a human driver behaviour model. This model describes the cognitive structure of the driver and represents the human decision making process.

The term “stochastic simulation approach” means that all important driver and vehicle parameters are chosen by stochastic distribution functions. The distributions are ensured by studies of the Chair of Traffic and Transportation Psychology of TUD and the current state of research. The focus lies on the driver’s cognitive structure. The vehicle models are simple point masses and the sampling rate is moderate, which means that there is an update every 100 milliseconds. Therefore, the simulation time is short and many simulation runs are possible. Simulations with more than 100,000 runs are planned. In this way, the simulation is able to simulate a large part of the realistic traffic flow within the entire traffic flow population.

Objectives:

The objective of TUD during the AutoDrive project is the evaluation of the selected safety function C-AEB from the partner VIF, regarding the aspect of its impact on road safety. Based on the evaluation process, TUD will develop a recommendation for system requirements to guarantee a minimum standard of safety. In addition, TUD and AIT try to develop test cases for the considered safety functions. These test cases are supposed to be recommendations for possible standardization tests with regard to the evaluation of the according safety functions.

The deliverable D5.13 explains how the theoretical aspects of the deliverable D4.20 are implemented. Furthermore, it presents the operating principles of the different modules on which the construction builds on.

2 OpenDRIVE

For the internal representation of the infrastructure, the traffic simulation uses the OpenDRIVE standard as already mentioned in D2.16. For a better understanding of the subsequent implementations, a short insight in the OpenDRIVE format is given in this chapter. Further information can be found on the OpenDRIVE website [1].

Three layers (reference line, lanes, and features) display the OpenDRIVE format. The first layer is the reference line, which represents the road. The reference line describes the mathematical geometry of the road. To describe the position along the road, the $s(t)$ -coordinate system according to Figure 1 is used. In this way, the position of vehicles or properties like traffic signs can be defined. A road network is obtained by the connection of different roads. The second layer *lanes* lies on top of the reference line system. It encodes the lane geometry along the road reference line. The last layer describes features like static objects (parked cars or traffic signs).

FIGURE 1: LAYERS OF OPENDRIVE [1]

The OpenPASS framework calculates inertial coordinate points (x, y) and the s -coordinate and heading angle for each lane on top by a fixed sampling rate. This information is stored in arrays representing the lane system.

3 Construction of the stochastic traffic simulation

This chapter explains the implementation of the modules of the stochastic traffic simulation. The focus lies on the realization of the driver behaviour theory of D4.20 as core part of the traffic simulation.

3.1 Perception

This section describes the implementation of the perception of the agents in the simulation. The agents perceive visual and vestibular information. The driver directly receives the ground truth acceleration data of the vehicle and the vestibular perception process is not explicitly modelled. However, the visual perception is modelled at a high level of detail. A great emphasis is put on modelling the cognitive structure and the right top-down and bottom-up mechanisms, which are already explained in D4.20. Subsequently, the implementation details of the visual perception are explained.

The useful field of view (UFOV) represents the area, where visual information can be received. It is depicted by a two-dimensional funnel (see Figure 2). The funnel possesses an opening angle and a maximum detection range. Agents, which are obstructed by other agents, cannot be detected.

FIGURE 2: REPRESENTATION OF THE USEFUL FIELD OF VIEW (UFOV)

The UFOV receives two categories of visual perception data:

- Object information
- Infrastructure information

Object information refers to the geometry information of the bounding box, position, indicator state, and brake light state of the observed agent. They are obtained by getting access to the perceived agent data. In addition, the current ego agent calculates the lateral displacement and heading error over the lane coordinates and the coordinates of the vehicle bounding box. The displacement and heading error are the input values for the lateral controller.

The road network, traffic signs, and buildings belong to the infrastructure information. The infrastructure is not detected by the UFOV. It is assumed that the agent peripherally perceives the current road and its successor. In cases in which the succeeding road is an intersection, all incoming intersection roads are also perceived. All other information along these roads is also detected.

3.2 Gaze determination

This module implements the gaze distribution of the driver. Four values describe the gaze state. The first value, the UFOV angle, defines the middle axis of the gaze direction. The opening angle and the view distance are self-explanatory. The fixation duration represents the time, in which the gaze state is valid. If the current gaze state is expired, a new gaze state has to be calculated. The driver behaviour model distinguishes between three different glances:

- Observation glance
- Control glance
- Scan glance

The ego agent performs observation glances in two cases. On the one hand, it does so when a collision is imminent. Then, the ego agent fixates the collision opponent. On the other hand, the ego agent performs an observation glance in cases in which it follows a leading vehicle. In these situations, it observes the leading vehicle according to the underlying probability. The UFOV angle is calculated by means of the orientation vector between the ego agent and the fixated agent.

The control glances belong to the rule-based behaviour. They are consciously performed top-down processes. The driver behaviour model considers two types of control glances. First, there always is a control glance while approaching an intersection. These glances are performed to evaluate the right of way at the upcoming intersection. The agent fixates the incoming intersection roads and looks out for approaching vehicles. Therefore, the fixation point along the incoming intersecting roads is determined by interpolating the lane coordinates at a certain view depth. The second type of control glances is performed at intersections with the aim to look out for vulnerable road users (VRUs). The fixation points are located at the intersection corners, where the VRUs cross the road. This type of control glances is determined by the calculation of the intersection points of the bordering sidewalks. Further, there is one special fixation point, which is used during left turns. While turning left, the agent secures himself whether vehicles come from the opposite incoming intersection road. For this purpose, the agent performs a control glance to the crossing opposite lane. Figure 3 displays the fixation points for both of the control glance types.

FIGURE 3: FIXATION POINTS FOR CONTROL GLANCES

(1 – EVALUATE RIGHT OF WAY REGARDING OTHER VEHICLES, 2 – CHECK FOR VULNERABLE ROAD USES)

The scan glance maps the random gaze behaviour of the human. He monitors the surrounding environment to look out for hazards. The scan glances are parameterised by the conducted driving simulator study of the Chair of Traffic and Transportation Psychology of TUD (see D6.23a). The ego agent draws scan glances according to the gaze distribution of the underlying phase he finds himself in. The different areas of interest (AOI) that can be fixated are divided into *straight*, *left*, *right*, *down*, and *other*. The AOI *straight* marks the area of the upcoming trajectory. The AOIs *left* and *right* mark the areas next to trajectory. The AOI *down* represents the dashboard or interior of the car and the AOI *other* includes gazes, which could not be assigned. Within the simulation, the AOIs *down* and *other* were combined to AOI *none*. At first, in order to modulate the scan glances, the view direction vector of the driver is calculated. The agent observes his upcoming trajectory. Therefore, it is assumed that he fixates a point on his trajectory that is two seconds ahead. With the help of the current velocity, the distance on the trajectory is determined and the fixation point is calculated. By subtracting the current position of the agent and the fixation point, the view vector is received. The UFOV angle for a scan glance at the AOI *straight* is given by the angle of the current view vector. The UFOV angles for the AOIs *left* and *right* are simulated by adding $\pm 30^\circ$ to the UFOV angle for *straight*. The AOI *none* is represented by a UFOV with an opening angle of zero degrees. In this way, no information can be acquired.

3.3 Cognitive map

The cognitive map stores all perceived information of the ego vehicle and the surrounding environment. D4.20 includes all theoretical aspects of the cognitive map. Subsequently, the implementation of the cognitive processes is shown.

The agent holds a mental representation of the surrounding infrastructure and agents. By getting new visual information, the currently represented world is updated. The perception module provides new object and infrastructure data at each time step. The cognitive map compares the stored data with the new information and updates the old mental agent representation. This happens by comparing the stored mental agent IDs with the new perception data. If the new perceived perception data includes an agent ID that is already stored, the old data is replaced by the new perception data. If no new perception data is presented for a particular mental model agent (e.g. in cases in which the ego agent is not looking at the area where the mental agent is located) the mental model data of the agent is extrapolated. The extrapolation is based on the latest present velocity and acceleration values. The updated data is provided to all other modules of the driver behaviour model.

3.4 Localisation

This section shows how the ego agent assigns his current phase. Deliverable D4.20 already introduced the different driving phases (see Figure 4). The current simulation state adds a further phase by dividing the *Deceleration* phase into *Deceleration I* and *Deceleration II*. The additional phase was added according to the results of the conducted driving simulator study (see D6.23a).

FIGURE 4: SEGMENTATION OF TASK ANALYSIS INTO: APPROACH, DECELERATION, CROSSING, TURNING, AND EXIT. EACH SEGMENT IS CHARACTERIZED BY SUBTASKS THAT ARE ALL LEADING TO ONE GOAL AND AT EACH SEGMENT, ONE CONSCIOUS DECISION HAS TO BE MADE. (ACCORDING TO [4, P. 85])

The ego agent locates himself with the help of the current distance to the stopping points. Therefore, the distances between the current s-coordinate of the agent and the considered stopping points are compared. The agent is located in the *Approach* phase when his distance to the first stopping point is smaller than 75 m and greater than 50 m. The phase *Deceleration I* starts at 50 m and ends at 25 m to the first stopping point. The phase *Deceleration II* continues up to the first stopping point. The agent is located at the *Crossing I*-phase when his s-coordinate is between the second and third stopping point. The *Crossing II*-phase for turning left is reached when the agent is between the third and fourth stopping point. The last intersection phase *Exit* starts behind the last stopping point and ends when the ego agent leaves the intersection and enters the next road. In all other cases, the agent is located at the phase *None*.

3.5 Navigation

The *navigation*-module implements the driver route decisions at intersections. Each agent gets a *turning decision vector* via the scenario input file. The scenario input file defines the spawn point values, such as velocity and position of the agent. It relies on the OpenSCENARIO format [2] and was extended by the *turning decision vector*. The turning states of the vector are *left*, *straight*, and *right*. While approaching the intersection, the present turning state of the *route decision vector* is chosen and the indicator state is set accordingly. The *navigation*-module also has to find the right successor lane, which the agent plans on following. Therefore, the direction vector of the current lane is calculated by the subtraction of the two last coordinates of the lane. Also, the direction vector of the first successor lane is calculated by means of the first and last coordinate of the lane. Afterwards, the angle between both of the vectors is calculated by the vector dot product. If the angle between the two vectors is larger than the chosen boundary condition for *straight*, it has to be evaluated whether the direction vector of the successor lane is pointing to the left or to the right. Therefore, the cross product of the two vectors is determined. If the cross product is negative, the successor lane leads to the incoming left intersection road. Otherwise, the successor lane leads to the right. In this way, the *navigation*-module is able to detect the successor lane that matches with the turning state left, straight, or right.

3.6 Right of way determination

The module *traffic rules* determines the right of way for all stored mental model agents of the cognitive map. The priority map saves the outcome of the evaluation. The priority status is saved by means of two Booleans; the first one reflects the priority state of the ego agent compared to the observed agent and the second Boolean considers the priority state the other way around. This is true when the observed agent has priority over the ego agent. At the current state, the traffic simulation can only process unsigned intersections, where the driver that comes from the right side has the right of way. To evaluate the priority, several conditions over all mental model agents have to be examined.

At the beginning, the easiest case between the mental model agent and the ego agent has to be verified: “both agents are driving in a row”. Therefore, it is checked whether the observed agent is situated on the predecessor, current, or successor lane of the ego agent. Is this case confirmed, both agents own the same rank of priority and so they both get priority. Is this case not confirmed, the *traffic rules*-module examines the trajectories of the observed and ego agent for overlapping. If the trajectories do not intersect, both agents do not affect each other and get priority.

Until now, it was only checked whether the trajectories of the agents affect each other. In the next phase, the actual right of way-determination will start by considering the type of agent. The simulation currently distinguishes between pedestrians and vehicles. Consequently, four cases are obtained:

1. Ego agent = vehicle, observed agent = vehicle
2. Ego agent = vehicle, observed agent = pedestrian
3. Ego agent = pedestrian , observed agent = vehicle
4. Ego agent = pedestrian , observed agent = pedestrian

Regarding the first case, it is initially determined whether the observed agent comes from a road that is located on the right from the ego agent’s point of view or not. Therefore, the current direction vector $\vec{d_e}$ of the incoming intersection lane of the ego agent is calculated. Next, the direction vector $\vec{d_o}$ originating from the ego lane pointing to the incoming intersection lane of the observed agent is calculated. By means of the algebraic sign of the z-coordinate of the vector cross product of both direction vectors, it is determined whether the observed agent comes from a left or right road. Subsequently, the calculation is performed the other way around. The point of view is changed and it is investigated whether the ego agent lane is located to the right or left of the observed agent lane. If one of the calculations shows that one road is located on the right, the right of way is obvious. Further investigations are only needed if both calculations lead to the outcome that the considered lane is located on the left side. This special case can only occur if both agents come from opposite lanes. Now, the indicator state has to be taken into account. The agent who is going straight ahead gets priority. If both agents have set their left indicator, no one gets priority because both Booleans are false.

The right of way for the second and third case is determined as follows: If the indicator is deactivated and the vehicle is driving straight, it has priority over the pedestrian. If the vehicle indicator is set to the right, the pedestrian has priority in two cases. Either, if the pedestrian crosses the intersection straight and comes from the right sidewalk on the same road or if the pedestrian comes from the left sidewalk of the opposite road. If the vehicle’s indicator is set to the left, the priority determination is similar except from the sides of the considered sidewalks, which have to be switched.

In the fourth case, both pedestrians have right of way.

3.7 Collision detection

The module *collision detection* modulates the driver estimation for an upcoming collision by means of a mental model agent. The estimation is based on the extrapolation of the agent positions. The *collision detection* estimates the future positions of the ego agent and the considered mental model agent for the upcoming three seconds based on their current velocities and accelerations. The geometries of both agents are calculated time-discrete by a given sampling rate (100 ms). At each time step, the geometries of the agents are checked for an overlap. If the geometries are overlapping, the agents will collide and the ego agent stores the collision data. This includes the collision opponent ID, the time to collision (TTC), and the distance to the collision point. Figure 5 shows the procedure.

FIGURE 5: PRINCIPLE OF THE COLLISION DETECTION

3.8 Action decision

The module *action decision* determines the wished acceleration. All mental model agents of the ego agent are considered one by one. After the calculation of all possible wished accelerations according to the varying underlying situations, the lowest acceleration is chosen. The calculation is based on the intelligent driver model (IDM) of Martin Treiber [3, p. 162]. The formula is based on the comparison of the current velocity v and distance s to the wished target velocity v_0 and distance s^* .

$$\dot{v} = a \left[1 - \left(\frac{v}{v_0} \right)^\delta - \left(\frac{s^*(v, \Delta v)}{s} \right)^2 \right]$$

$$s^*(v, \Delta v) = s_0 + \max \left(0, vT + \frac{v\Delta v}{2\sqrt{ab}} \right)$$

Parameter T represents the time headway of one vehicle following another. The parameters a and b reflect the maximum acceleration and the comfort deceleration of the driver. The exponent δ describes the way in which the acceleration approximates zero by reaching the wish velocity.

To parameterize the IDM model correctly, the first step is to set the target velocities of the underlying phases. These target values are the IDM wished velocities of the driver. Now, according to the present scenario, the wished velocity, acceleration, and distance values have to be set. Therefore, following states have to be checked:

Crash imminent: This state is reached, if the time to collision (TTC) has fallen below a certain threshold or if the maximum comfort acceleration is not sufficient to stop the vehicle until the collision point. If

the state is true, the maximum acceleration values are set to the emergency values and the acceleration part for free driving $a_v = a \left[1 - \left(\frac{v}{v_0} \right)^\delta \right]$ is not considered. If the ego agent is already beyond the stopping point, the distance s is set to the distance to the collision point. Otherwise the distance is set to its stopping point. T is set to zero.

Right of way: The already calculated right of way-map for all mental model agents is taken into account. If the ego agent has priority, his target velocity is set to the default value for his current phase and s is set to infinity.

Following state: If the ego agent is following the mental model agent, all parameters are applied according to the formula.

Gap estimation state: If the considered mental model agent has priority over the ego agent, it is considered whether the ego agent could pass the intersection before the observed mental model agent reaches the intersection or not. If the calculated gap time for the ego agent is high enough, the priority of the observed mental model agent is ignored. Therefore, the ego agent sets his target velocity to the default velocity of the underlying phase and s to infinity.

3.9 Lateral controller

The lateral controller calculates the wheel angle δ_{Sta} to keep the agent on his chosen trajectory. At the current simulation state, the chosen trajectory is the middle of the current road lane. The traffic simulation uses the Stanley control law, which has two terms. The first term considers the orientation error ψ . The second term considers the front lateral displacement e and the current velocity v_x . The variable k is the track error gain. Figure 6 displays the geometric principle.

$$\delta_{Sta}(t) = \psi(t) + \tan^{-1} \left(\frac{k e(t)}{v_x(t)} \right)$$

FIGURE 6: STANLEY CONTROLLER (ACCORDING TO [5, P. 584])

3.10 C-AEB system

The evaluated safety system is the cooperative autonomous emergency break (C-AEB), which is based on the model of VIF. The safety system is supposed to minimize the number of accidents and the collision speed with VRUs and vehicles in urban regions.

The requirements of VIF are: *“In a speed range up to 60 kilometres per hour, collisions with VRUs or objects, which cross the trajectory of the vehicle, are located on the trajectory, or are moving along the trajectory, should be detected. The VRU detection should be ensured even in adverse weather*

conditions. If a VRU is detected (in the immediate vicinity of the driving trajectory), the behaviour is recorded and predicted. The possibly dangerous situation is continuously analysed and if necessary, safety corridors are predicted. These can initiate an emergency braking or evasive manoeuvre, which should avoid a collision or reduce accident effects and the impact. In addition, subsequent vehicles are warned.”

The C-AEB was implemented in a reduced scope. The C-AEB is only able to perform longitudinal interventions by breaking. Also, the weather conditions are not simulated at present.

The C-AEB consists of two modules. First, there is the *sensor*-module and second, the *collision*-module. The sensor-module detects all simulation agents that are located in the sensor funnel and are not obstructed by each other. The opening angle is varied according to the safety evaluation study (see D6.23b). The collision module works similar to the collision detection of the driver behaviour module. It extrapolates the future position of the potential collision opponent and the ego vehicle. If a collision state is detected within the given extrapolation time, an emergency brake manoeuvre is performed. The C-AEB can be parametrised by the input file *profilesCatalog.xml*. Herein, an additionally safety zone can be added to the vehicle bounding box. If an object within the selected TTC threshold enters this safety zone, an emergency break manoeuvre is performed.

4 Outlook

It follows a small outlook of the further steps within the project: The project is in the final phase. So far, all tasks have been successfully implemented. The next deliverable D6.23b will contain the validation of the stochastic traffic simulation. Furthermore, the implemented safety system (C-AEB of VIF) will be evaluated.

5 List of figures

Figure 1: Layers of OpenDRIVE [1]	6
Figure 2: representation of the useful field of view (UFOV)	7
Figure 3: Fixation points for control glances (1 – evaluate right of way regarding other vehicles, 2 – check for vulnerable road uses)	8
Figure 4: Segmentation of task analysis into: approach, deceleration, crossing, turning, and exit. Each segment is characterized by subtasks that are all leading to one goal and at each segment, one conscious decision has to be made. (according to [4, p. 85])	10
Figure 5: principle of the collision detection.....	12
Figure 6: Stanley controller (according to [5, p. 584])	13

6 Bibliography

- [1] M. Dupuis, E. Hekele, and A. Biehne.a, “OpenDRIVE® ,Format Specification, Rev. 1.5,” 2019. [Online]. Available: <http://www.opendrive.org/docs/OpenDRIVEFormatSpecRev1.5M.pdf>.
- [2] ASAM, “ASAM OpenSCENARIO.” [Online]. Available: <https://www.asam.net/standards/detail/openscenario/>.
- [3] M. Treiber and A. Kesting, *Verkehrsdynamik und -simulation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.
- [4] M. Plavsic, “Analysis and Modeling of Driver Behavior for Assistance Systems at Road Intersections,” 2010.
- [5] M. Cibooglu, U. Karapinar, and M. T. Soylemez, “Hybrid controller approach for an autonomous ground vehicle path tracking problem,” in *2017 25th Mediterranean Conference on Control and Automation (MED)*, 2017, vol. 8, pp. 583–588.