Track- / Scenario-based Trajectory Generation for Testing Automated Driving Functions

Aaron Heinz*, Wolfram Remlinger* and Johann Schweiger† *AUDI AG - Process and Method Development .
Email: {aaron1.heinz, wolfram.remlinger}@audi.de †Technische Hochschule Ingolstadt
Email: {johann.schweiger}@thi.de

***Abstract*—Test drives for validating automated driving functions are expensive and not always suitable. Therefore, simulation-based virtual test drives were invented. Both ways of testing offer pros and cons. Mixed reality test drive (MRTD) is an novel approach for combining the pros of reality and simulation. It uses real environmental conditions of real test drives and the accuracy and reproducibility of virtual test drive vehicle behavior descriptions (scenarios). Until now no method for the derivation of trajectories from a maneuver based scenario description exist. In this paper a methodology for the derivation of vehicle target trajectories from a scenario is introduced. The approach utilizes the OpenScenario description format for modeling dynamic vehicle behavior composed of sequential maneuvers. In particular, maneuvers can further be divided into conditions and actions. Due to physical restrictions of real vehicles, conditions and actions are evaluated regarding operability. Nonetheless, the composition of all maneuvers of a vehicle leads to its relative trajectory. Consequently, in combination with an OpenDrive geometric road-network, a geographically-referenced global target trajectory is derived. The presented methodology forms the basis of the MRTD approach by providing global target trajectories as input data for a vehicle motion controller. In addition, guidelines regarding practicable OpenScenario conditions and actions of maneuvers in the context of MRTD are defined. Finally, an overtake scenario is used to evaluate the trajectory generation with virtual test drive scenario simulation.**

***Index Terms*—OpenDrive, OpenScenario, Mixed Realty Test Drive, trajectory generation, digital maps, reference path**

I. Introduction

In the development of automated driving functions, testing and validation are indispensable. Validating new functions in complete systems by test drives on test tracks or public roads is widely used today. Nonetheless, test drives are expensive and not always suitable. Therefore, simulation-based virtual test drives where invented. Both offer pros and cons.

One approach for combining theses two worlds is vehicle in the loop (VIL). VIL combines a virtual visual simulation with the kinesthetic, vestibular, and auditory feedback of a real car [1]. The visual simulation is used for stimulating the driver and also the sensors of a vehicle. Since autonomous driving functions intend to release a driver from the driving task the visual simulation is not longer uppermost.

Another approach of combing real and virtual worlds for vehicle test purposes is the mixed reality test drive (MRTD). Unlike VIL, MRTD focuses on the controlled vehicle motion of multiple real vehicles within a traffic situation (scenario). The initial idea comes from a sub-project of the research

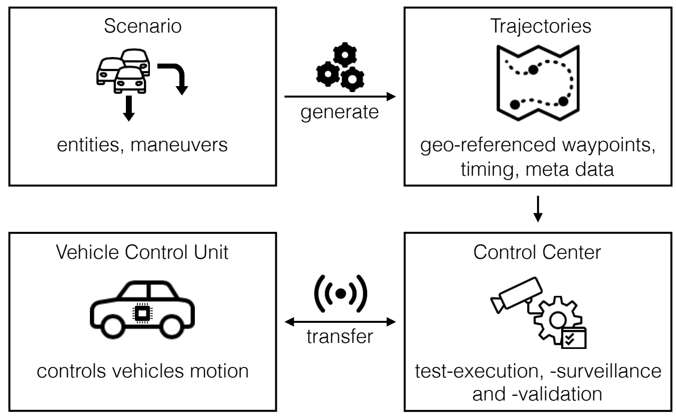


Fig. 1. Components of the MRTD approach. The *Scenario* and *Trajectories* elements are addressed in this paper. *Vehicle Control Unit* and *Control Center* are focused by the PEGASUS sub-project

project PEGASUS[1]. Sub-project proving ground tests aim to control multiple traffic simulation vehicles in order to stimulate a test vehicle, (which runs a automated driving function), on a real test track. For the MRTD approach, this use case is extended by former scenario descriptions since they are used commonly for virtual test drives. Thus the labeling mixed reality test drive arises from the virtual vehicle control and the resulting vehicle motion of a real car. The special aspect of MRTD is the use of real environmental conditions from real test drives combined by the accuracy and reproducibility of virtual test drives. An essential part in this context is the motion-planning of prepared test vehicles based on scenario descriptions. To ensure patency between virtual test drive and MRTD, the same description formats for environment and behavior are used. Figure 1 displays the components of the MRTD approach. The scenario defines the behavior of all vehicles. From this behavior description, a global trajectory for every vehicle is generated. A control center sends the global trajectories and additional control information to the specific test vehicles. Finally the test vehicles execute the trajectories by a controller during the test run.

In general MRTD provides two use-cases for test runs: For homologation of automated driving functions the traffic vehicles around a homologating vehicle (Vehicle Under Test) are controlled by MRTD. The traffic vehicles stimulate the

[1]PEGASUS research project supported by the federal ministry for economic affairs and energy - URL: http://www.pegasus-projekt.info/en/

automated driving function of the VUT. In order to pass the homologation the VUT needs to interact with the traffic vehicles as defined in a test specification (PEGASUS goal).

For vehicle development purposes all vehicles within a scenario are controlled by MRTD. A VUT is lead into a critical situation which causes the automated driving function to interfere with control instructions.

As mentioned before, trajectories are used as level of abstraction which are processed by vehicles equipped with a trajectory controller. A trajectory is a time-ordered sequence of sampled locations along the route of a moving object. It contains not only spatial but also temporal information. As a result a trajectory defines a deterministic motion behavior.

In this work a methodology for the derivation of vehicle trajectories from virtual road networks and scenarios is introduced. The approach utilizes the OpenDrive format (*.xodr) as road-network database and the OpenScenario format (*.xosc) for modeling dynamic vehicle behavior composed of sequential maneuvers. Both formats are evaluated regarding spatial and temporal influences to trajectories and guidelines regarding MRTD are developed. Next, a architecture of the trajectory generation is introduced. Within this section the task of a trajectory generation is separated into different subsequent levels: the route planning, the maneuver sorting and at least the trajectory generation itself. In Section IV-C a use case example of an overtaking-scenario illustrates the introduced architecture. Thereby the tasks derived from the architecture levels are processed step by step by example values from the overtaking-scenario. Finally the generated global trajectories are evaluated in Section V regarding geographical localization and compared to a scenario simulation.

II. Description Languages

In order to simulate automated driving functions virtual model of environments including static and dynamic objects are required. Therefore specific description formats were invented. The following section will focus on the OpenDrive format for road network descriptions and the OpenScenario format for describing the behavior of dynamic objects within a scenario. A detailed definition of the term scenario in the context of traffic situations is given by Ulbrich et al. [2].

A. OpenDrive

OpenDrive [3] is an open file format for the logical description of road networks. It enables the description of track-based road networks using Extensible Markup Language (XML) syntax. OpenDrive files specify the geometry of roads as well as features along roads influencing the logics (e.g. lanes, signs, signals). Hereafter not all features of the format will be explained in this work. The paper focuses on relevant features regarding spatial trajectory information.

1) Coordinate Systems: Track and inertial coordinates are the two most frequently used coordinate systems of the Open Drive Format. The inertial system is a right-handed coordinate system according to ISO 8855 [4]. The track system is also a right-handed coordinate system, applying along the reference line of a road. Consequently, the s-coordinate defines the position from the beginning of a road calculated in the xy-plane, the t-coordinate defines the lateral position (positive to the left) and the h-coordinate defines a position above the road.

2) Geographic Referencing: OpenDrive provides geographic referencing for real road networks derived from measured data or building plans. Setting a projection definition formatted as Proj.4-string specifies a reference to the real world. Proj.4 is a standard Unix filter function which converts Cartesian coordinates into geographic longitude and latitude coordinates and vice versa [5]. Reference locations embedded in a Proj.4-string enables precise conversion from road network coordinates into GPS locations.

3) Road Layout: The geometric shape of a road is defined by its reference line in the xy-plane. Along the reference line, various properties of the road may be defined. These are, e.g. elevation profile, lanes, traffic signs etc. Every road within a road network owns a unique id. Thus roads can be linked to each other either directly (when there is only one connection possible between two given roads) or via junctions (when more than one connection is possible from a given road to other roads). A reference line is defined by consecutive geometries such as straight lines, spirals (clothoids), arcs, cubic polynomials and parametric cubic polynomials.

B. OpenScenario

OpenScenario [6] (currently in the making[2]) is an open file format for maneuver based behavior description of dynamic traffic contents. The format provides a common base for describing catalog-based vehicle scenarios using XML syntax. Referred to the scenario defintion from Ulbrich et al. [2] a scenario consists of *actions&events*, *goals&values* and a scene which further contains *dynamic elements*, *scenery* and *self-representation*. Consequently, these components are also found in the OpenScenario format. Dynamic elements within a scenario are called *entities* e.g. vehicles, pedestrians, cyclists or animals and are organized in catalogs for vehicles, drivers, pedestrians and miscellaneous objects. The scenery of a scenario includes the road network, time and weather conditions and is organized in a environment catalog and a OpenDrive road network. Self-representation is indicated by the *ego-vehicle* in context of MRTD the ego-vehicle stands for the VUT. Actions&events and goals&values are represented and organized by catalogs of maneuvers in the OpenScenario format. The mapping between maneuvers and entities is organized in a so called *storyboard*. Consequently, a storyboard represents the chronology of all maneuvers within a scenario.

1) Positions: OpenScenario provides world-, road- and lane-positions for the determination of a entity location within a road network. All positions are based on OpenDrive's coordinate systems and either absolute or relative to another entity defined. World positions are defined by inertial coordinates. Road positions are defined by road-ids and track-coordinates, additional lane-ids define lane positions.

[2]OpenScenario revision 0.9 schemata, mindmap and examples URL: http://www.openscenario.org/download.html

2) Routing: A route is defined by a sequence of way-points. Every way-point contains a position (world, road or lane) an a routing strategy. The routing strategy defines a routing method between the current and following way-point. Available routing strategies from the OpenScenario format are: fastest, shortest, least intersections or random. Routes are used to navigate entities within a road network via relevant junctions or infrastructure.

3) Trajectories: OpenScenario format facilitates the definition of relative trajectories. In this context a trajectory consists of a sequence of vertexes, which further contain a position and a shape e.g. polyline, clothoid or spline. The trajectories are used for custom lane changes or user defined longitudinal and lateral motions.

4) Storyboard: The storyboard consists of an initialization, the story itself and an end. Within the initialization the initial situation of a scenario is represented. In the context of MRTD this definitions represent the preconditions of a test run. The scenario story itself contains different maneuvers and their correlating actors. Maneuvers are further divided into events, in which every event holds a priority. To be more precise, an event consists at least of one action and also one corresponding condition-block containing at least one start-condition. At last, the end of a storyboard contains an optional condition, which could be used to define a particular end condition of a scenario.

A example scenario story is described as follows: Ego starts $40m$ in front of Overtaker by same speed of $50km/h$. At world position wPos1, Overtaker accelerates at $4.00m/s^2$ till $75km/h$. After this action Overtaker should keep his speed until longitudinal distance to Ego is lower than $20m$. If a lower distance is achieved, Overtaker should perform a lane change to left within $2.00s$. If longitudinal distance between Ego and Overtaker is greater than $20m$ Overtaker should perform a right lane change within $3.00s$.

5) Actions: Actions are used to manipulate a scenario. As illustrated in the pseudo code above actions affect vehicle motion and thus their trajectory. This section introduces all predefined actions of the OpenScenario format and evaluates them in the context of a MRTD practicability. In particular, actions are classified into *private*, *global* and *user defined*.

A user defined action executes a corresponding script. If a custom action (which is not provided by OpenScenario) is needed the action could be implemented within this script. How a entity is affected by this action depends on the implemented script. Therefore user defined actions may not be taken into account for the trajectory generation unless their impact according to its entity's motion is known.

Global actions are used to manipulate the scenery. Available global actions facilitate the manipulation of environment (time, weather, road-conditions), entities (add, remove), parameters (set, modify), infrastructure (signal states) and traffic (source, jam). Because global actions have no effect on lateral- or longitudinal-motion of an entity they are not relevant for the trajectory generation.

Private actions are used to manipulate entities. In particular not all private actions of OpenScenario are applicable in order to generate a continuous trajectory geometry. Therefore only the following private actions are considered in this work. A position action sets the position of an entity at a trigger time. In order to provide a continuous trajectory geometry of an entity, as it is required for a MRTD, position actions are only allowed at the initialization of a scenario. Longitudinal actions are used to change an entity's speed or force an entity to gain a given longitudinal distance between two entities. The dynamics of this action is defined by a shape (e.g. linear, cubic, sinusoidal or step) and a rate, time or distance whereby only one attribute could be set at the same time. If no dynamics is given, the entity is set to the given target value (speed or distance). This means that only longitudinal actions with a valid dynamics definition are applicable by MRTD. Dynamic of longitudinal actions result in vehicle requirements considering the execution of a generated trajectory. Lateral actions are used to change an entity's lane on a road or force an entity to gain a given lateral distance between to entities. They behave similar to longitudinal actions, except dynamics misses the rate attribute and the target value refers to an absolute or relative position. A missing dynamics definition according to a lateral distance will cause a break in the geometrical smoothness of a trajectory. For the trajectory generation suchlike definition is only at the initialization of a scenario allowed.

Compared to path planning of automated vehicles the parameters for a longitudinal or lateral action are not sufficient enough to remodel all maneuvers. A lane change for example may be defined by a fifth order polynomial [7], [8] whereby a OpenScenario lateral action only linear, cubic, sinusoidal or step shapes provides. In this case a routing action is used to trigger a entity following a custom relative trajectory which could be defined by a fifth order polynomial.

6) Conditions: Conditions determine if an corresponding action is either started, canceled or stopped, depending on its use. At latest a condition defines the when and where an action is triggered in the process of a scenario. For this reason predefined conditions of the OpenScenario format are very important for the trajectory generation.

Conditions are classified into *byEntity*, *byState* and *byValue*. In addition, every condition holds a delay value and an edge value. The edge can be set as rising, falling or any and is used for byValue conditions.

ByValue conditions trigger on absolute values of a scenario environment e.g. given parameter values, simulation time or time of day.

ByState conditions trigger on a specific state of a controller (e.g. a gear state) or the state of a signal (e.g. traffic light state) or user defined commands. This conditions can also be used to trigger at start or after termination of an scene, act, maneuver, event or action.

ByEntity conditions refer at least to one defined entity and trigger on relative or absolute positions or entity relationships (e.g. Speed, Distance, TimeToCollision).

In order to generate trajectories for a MRTD, a scenario definition needs to be predictable and thus deterministic. In this context user defined byState command conditions are only allowed if they are predictable. All other byValue or byEntity conditions are analyzable and always led to the same result, unless their parameters are changed.

III. Architecture of Trajectory Generation

According to Zhang et al. [9] vehicle navigation in general can be divided into three different levels to carry out different tasks including route planning at macro level, maneuver planning at meso level and trajectory planning at micro level. Transferred to the goal of controlling a vehicle based on a scenario description, these three levels are used to separate the different tasks. As a result a deterministic vehicle motion description in form of a trajectory is generated (figure 2).

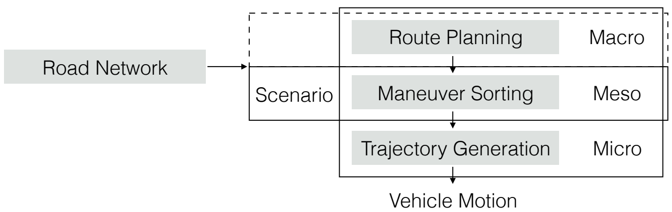


Fig. 2. Proposed architecture of trajectory generation

A. Route Planning

Route planning in general defines an optimal route or journey between two locations under a given road network. The road network is part of a scenarios environmental definition. Routes of traffic vehicles represent the motion of dynamic elements, and is therefore also defined within a scenario. At least the route of the ego-vehicle is part of the self-representation and thus also defined within a scenario. Consequently, for simulation based testing, a road network and routes are a unique selling point of a scenario. Changing the specification of a road network or a defined route of a vehicle could lead to another scenario.

Scenarios for testing use cases are often derived from measurement-data [10]. In this process the underlying road network of a scenario is specifically modeled from captured vehicle trajectories. As a result, every scenario refers to its own road network. For the MRTD approach a scenario has to be mapped onto a given road network, caused by the availability of only a view real test tracks. By mapping a scenario on another road network all test specifications regarding the road has to be considered. This means if a scenario is defined on a straight highway with three lanes and a length of four kilometers or a specific curvature, a real test track needs to fulfill at least a minimum of these attributes. The minimum of attributes depends on the test specification. Otherwise a simulated scenario and a real driven scenario are not comparable. As an example if a test does not focus on a specific road detail, a country road scenario could also be performed on a real test track with more then two lanes.

Since a OpenDrive road network description provides the geometry of a road, all roads of a planned route result in the geometry of a trajectory along this route. Summarized the route planning task analyzes the minimum of required road attributes along a route. As well as mapping this requirements onto a given road network of a test track. As a result, the spatial reference line of a trajectory is defined.

B. Maneuver Sorting

In general, maneuver planning generates a series of vehicle behavioral demands based on a detected traffic situation to carry out part of the overall driving tasks defined by a planned route. The traffic situation and all behavioral demands along a route are defined as maneuvers within a scenario. In particular behavioral demands are defined by action&events and goals&values. Since the OpenScenario format defines maneuvers not in a sequence of actions but events triggered by conditions, their execution order needs to be sorted chronologically. Moreover the interdependency between maneuvers of different actors (vehicles) needs to be evaluated. Depending on the quantity of actors referring to one another the sorting may become arbitrary complex. To reduce the complexity a longitudinal velocity profile of every actor is created. The profile allows combinations of constant velocities, acceleration or deceleration caused by longitudinal or lateral actions and their conditions along a planned route [11]. Lateral actions cause a change in the spatial reference line of a trajectory. Therefore a lateral offset profile for every actor is created. Both profiles are used to determine condition triggering.

Summarized the maneuver planning is transformed into a maneuver sorting task. It generates velocity profiles and lateral offset profiles along a planned route according to chronologically ordered maneuvers. As a result, trajectories are spatial and temporal defined.

C. Trajectory Generation

The trajectory generation revers to the trajectory planning task of an autonomous vehicle navigation which is to provide a dynamic path for a vehicle to follow with time scale often in milliseconds. The planned trajectory in general combines geometric path information with time stamps to perform the driving tasks defined by a maneuver. As an example one common straightforward approach for a lane change maneuver trajectory is the calculation of fifth order polynomials, since they can establish smooth transitions between the vehicles actual and desired final positions [12]. In this context, calculated trajectories refer relative to the actual vehicle position. From a global perspective a trajectory is consequently, the total of all relative trajectories. Snider et al. compared a variety of different path tracking methods for controlling a car-like robot along a predetermined path with the conclusion that control performance is strongly dependent on the curvature continuity and smoothness of the reference path [13]. In order to control a car for MRTD by a generated trajectory, the geometry of the trajectory needs to be smooth and curvature continuous.

The trajectory generation task in this work aims to provide global trajectories covering a whole scenario. This is achieved by the combination of maneuver planned velocity- and lateral offset-profiles and the geometric path description from the route planning. Since global trajectories contain many way-points a representation by one polynomial is not possible. Either a global trajectory is defined as a total of higher order polynomials or, such as in this work, by a time-ordered sequence of way-points defined by time, location, heading, velocity and acceleration.

IV. Implementation

In this section a method for a scenario based trajectory generation based on the proposed architecture of Section III is introduced. As case example the overtaking scenario from Section II-B is used. The scenario contains route-following maneuvers, one acceleration maneuver and two lane-change maneuvers. The route-following maneuvers defines the basic road-course. The acceleration maneuver combines a world-position condition and a longitudinal speed-change action. The lane-change maneuvers combine relative-distance conditions and lateral lane-change actions. The used database is a map of the Audi Driving Experience test track (figure 3).

A. Route Planning

The given scenario takes place on a straight country road with two lanes. Both lanes with a width of $3.75m$. The total required road length of the scenario is $392m$. Whereat the road length in this case only revers on the scenarios maneuvers. In order to perform a MRTD additional space for acceleration and deceleration is required. Since this aspect is not captured by a scenario definition, it is not taken into account for the trajectory generation in this work. As a result, the required road attributes are $392m$ of a straight road with two lanes ($3.75m|3.75m$).

If a given test track does not completely match the required road attributes of a scenario, the road attribute requirements on maneuver level may be analyzed. For that matter the impact of a road geometry on a maneuver is analyzed. In this case the lane-change maneuver takes place on a straight road, if the maneuver is mapped on a curved road, the lane-change trajectories are effected by the curvature [14]. As a result, the calculated trajectory does no longer match the scenario description. Furthermore the trajectory of a lane-change maneuver depends on the lane widths of a scenarios road. As a result, the given test track must at least provide lanes of the same width or wider. The route-following and acceleration maneuvers are also located at a straight road. In case of a minimal lateral acceleration the distance offset caused by vehicle physics between a straight and a curved road are sustainable. The lateral acceleration ($acc_{lateral}$) depends on velocity and radius (r_{curve}) of a curved road as displayed in the following equation:

$$acc_{lateral} = v^2/r_{curve}$$

Consequently, for low velocities and large radius the lateral acceleration is minimal. To put in a nutshell rout-following maneuvers or acceleration maneuvers may also be mapped onto curved roads, if current and target velocity is low and curve radius high.

The road network used in this example is a map of the Audi Driving Experience Center test track in Neuburg an der Donau (figure 3). The road network contains a 2.200m handling track and a oval with 1.250m straights separated into 3 marked lanes. The OpenDrive database of this track was created on building plans and may not be as exact as a road network created on laser-scanned data [15]. But it is currently the only available road network of the Audi Driving Experience Center test track including junctions representing a real test track. The geographical referencing quality of this road network is evaluated in Section V.

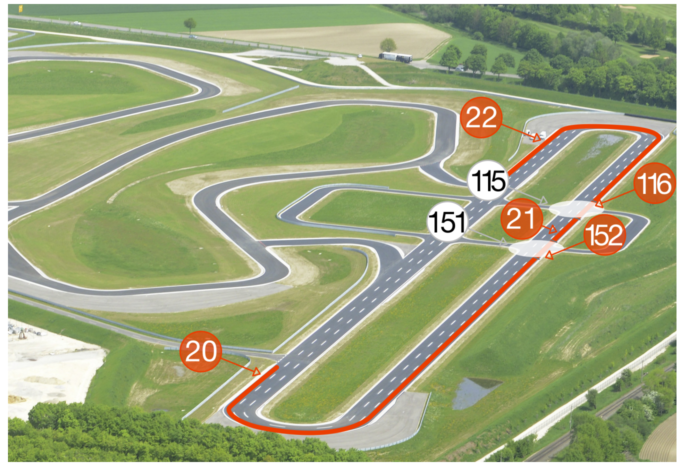


Fig. 3. Planned route displayed as line and OpenDrive database road-IDs (20, 21, 22, 116, 152), junction-IDs (115, 151) on Audi Driving Experience track, Neuburg an der Donau, Germany

The required road attributes of the scenario are found on the oval of the test track. It provides long straights and curves with large radius and 3 lanes. For low velocity the curves are used for route-following maneuvers. The red line on figure 3 marks the planned route on the oval, the orange areas display junctions. In addition the mapping of a scenarios maneuvers along the route may be adjusted by a s-coordinate offset from the route start. First of all, the routing over the desired road-IDs from the OpenDrive database was done manually by selecting the road from a start point and linking through its predecessor information to the next road-ID. Junctions linking roads on lane level, this means a incoming road on lane 2 is linked over a connection road on lane 1 to a outgoing road on lane 2. In order to prevent offsets between reference lines the connection over a junction needs to be performed on lane 1 or -1, as they are next to the reference line on lane 0. The continuity of all reference lines along a route is especially important since they represent the basic geometry of a generated trajectory. As a result, table I represents the planned route from the route planning task. The route is represented in a geometric ordered sequence of desired road-ids and additional successor, predecessor, type and lane information (road-IDs and junction-IDs also displayed on figure 3).

TABLE I
OpenDrive database routing table of planned route

Road ID	Type	Lane	Successor	Predecessor	Incoming
20	road	-1	Start	151	n/a
151	junction	n/a	n/a	152	20
152	road	-1	20	21	n/a
21	road	-1	151	115	n/a
115	junction	n/a	n/a	116	21
116	road	-1	21	22	n/a
22	road	-1	116	End	n/a

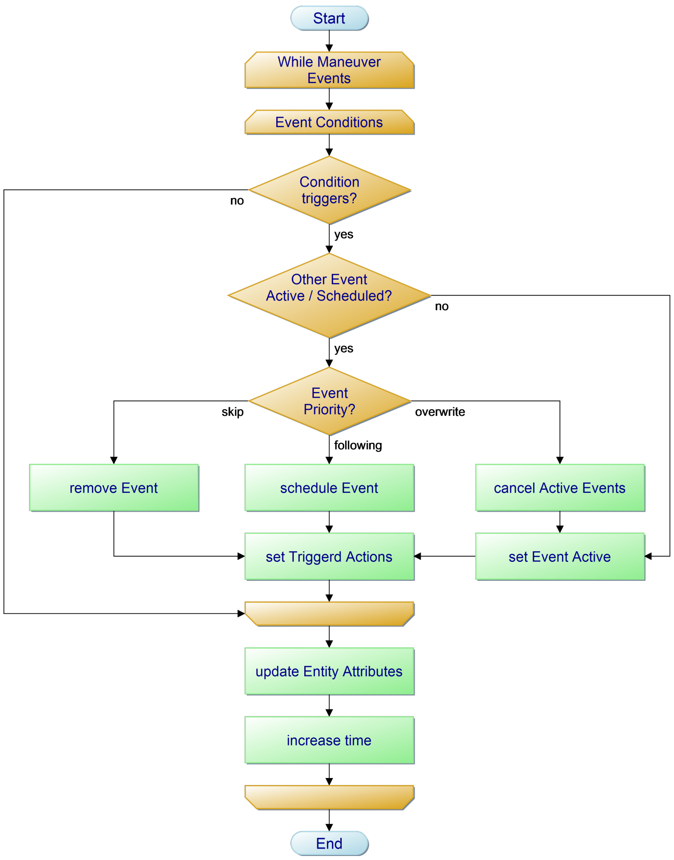
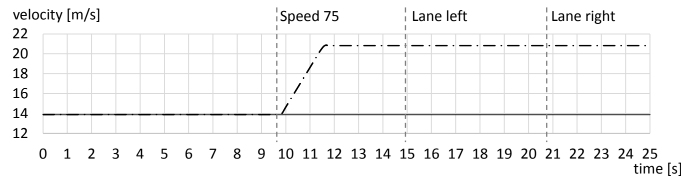


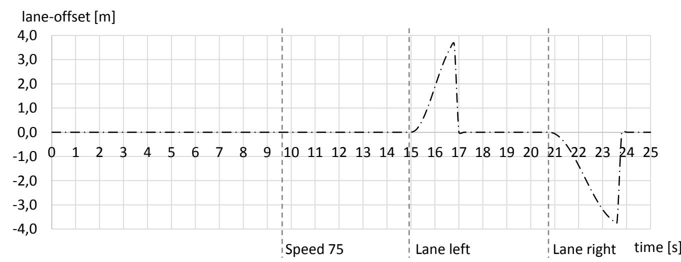
Fig. 4. Flowchart of maneuver sorting task

TABLE II
ACTION TRIGGER TIMES SORTED CHRONOLOGICAL

Name	Type	Start Condition	Start	End	Duration
Speed 75	lat	Pos. World	$9.81s$	$11.54s$	$1.73s$
Lane left	lon	rel. Dist. 1	$14.99s$	$16.99s$	$2.0s$
Lane right	lon	rel. Dist. 2	$20.75s$	$23.75s$	$3.0s$



(a) Velocity Profiles



(b) Lane-Offset Profiles

Fig. 5. Velocity and lane-offset profiles with marked trigger times from the example scenario. Solid line: ego, dot-dashed line: Overtaker

B. Maneuver Sorting

In general, the maneuver sorting tasks orders the following-route maneuvers, acceleration-maneuver and lane-change maneuver of the scenario according to its conditions. As mentioned in Section II-B6, the OpenScenario format provides different conditions, which may be used for starting, canceling or ending a maneuver-event. All conditions depend at least on one entity. During the chronological course of a scenario the attributes of a entity e.g. position or speed may change caused by a triggered action. In particular, the detailed lateral or longitudinal behavioral demand is defined by the dynamic description of an action. In the case example scenario the speed-change action is defined by an acceleration of $4.00m/s^2$ and a step action-shape. The left lane-change action is defined by a duration of $2.00s$ and a sinusoidal action-shape. The right lane-change action is defined by a duration of $3.00s$ and also by a sinusoidal action-shape. The lateral offset during a lane-change action from the current lane-middle to the next lane is defined as lane-offset. All entity attributes (velocity, acceleration, position lane, lane-offset) are calculated over the scenario duration. As a result, velocity profiles and lane-offset profiles from the entities are generated. In the process the conditions are evaluated regarding execution. In addition, the start- and end-timing of maneuver events are logged. Figure 4 displays the flowchart of the maneuver sorting task.

The resulting action trigger times of the case example scenario are shown in table II. Figure 5 displays the resulting velocity and lane-offset profiles of Overtaker (dot-dashed line) and ego (solid line). The action trigger times are marked as well (vertical dashed lines). The velocity profiles illustrates booth entities start with the same velocity. After the acceleration-maneuver trigger time ($9.81s$) Overtaker accelerates up to a constant velocity of $75km/h$. The lane-offset profile shows the lateral offset from a lane middle caused by the lane-change maneuvers. As displayed Overtaker performs at first a left lane-change at $14.998s$ within $2.0s$ followed by a right lane-change at $20.758s$ within $3.0s$.

C. Trajectory Generation

The trajectory generation task uses the velocity profile and lane-offset profile from the maneuver sorting task and combines it with the road geometry based on the route planning task. As a result, a trajectory is defined by a chronological sequence of way-point coordinates and corresponding time, velocity, acceleration and heading information. Finally the inertial way-point-coordinates are transformed into GPS-coordinates.

As mentioned in Section II-A3, every road of an OpenDrive database is defined by a geometric function and uses the track coordinate system. The s-coordinate from a track-coordinate is calculated by a velocity-profile at a given time. The t-coordinate is calculated by lane-middle offset and lane-offset profile at a given time. Transforming this track-coordinate into an inertial-coordinate leads to an trajectory way-point (x, y, z) at a given time. Since the s-course (geometry) of a road is defined by a function, the time sample-rate is arbitrary. This method works until the curvature of a reference-line is

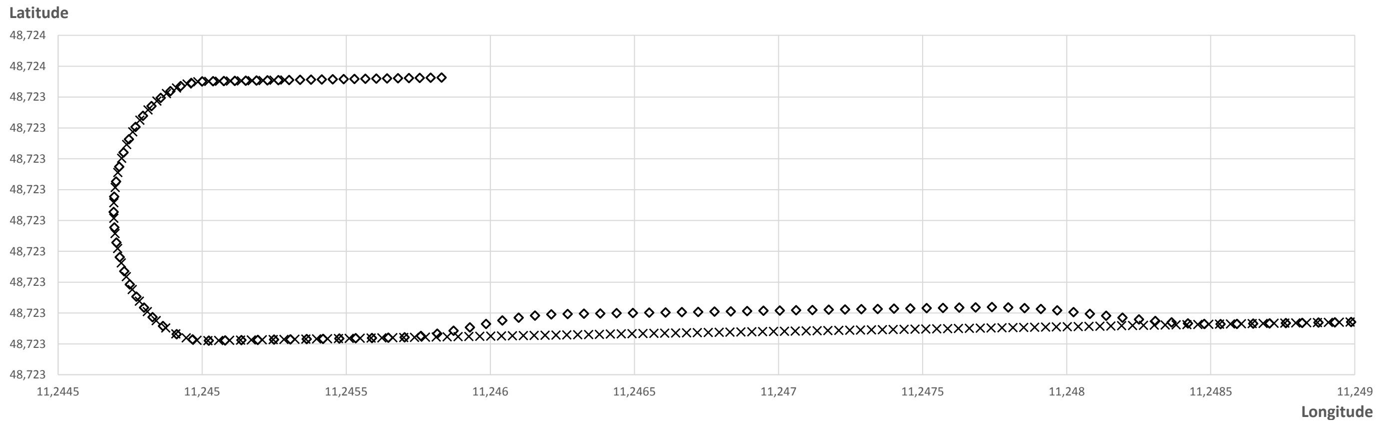
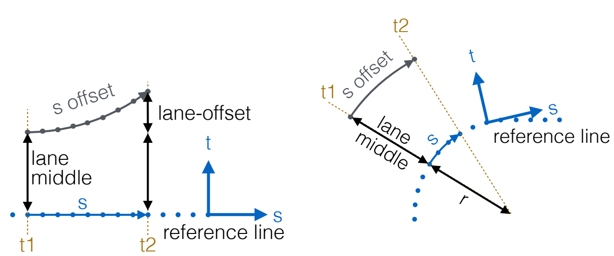


Fig. 6. Generated trajectory way-points of ego (crosses) and Overtaker (rhombuses)

constant. In case of a changing reference-line curvature the s-coordinate on a lane differs from the s-coordinate on a reference line. If a lane is on the outer of a curve's reference line (figure 7b), the s-coordinate offset between reference line and lane becomes greater. If a is on the inner of a curve's reference line, the s-coordinate offset becomes less. Furthermore, in case of a changing lane-offset value, caused by a lateral action (lane-change), the resulting curvature of a trajectory also changes (figure 7a). This s-coordinate offsets are fixed by length adjustments based on a lanes arc length.

Finally, all trajectory way-points are transformed into GPS-coordinates by the corresponding geographical reference of the OpenDrive database. This steps are repeated for every entity within a scenario. In this case example scenario two trajectories are generated. The trajectory of the ego entity is displayed by blue crosses, the trajectory of Overtaker is displayed by red rhombuses on figure 6.



(a) Curvature change caused by lane-offset

(b) Curvature change caused by reference line

Fig. 7. S-coordinate offsets caused by changing curvature

D. Metadata

In addition, to the trajectory generation task scenario metadata is derived. This metadata refers on maneuvers within a scenario and the used road network. The metadata is independent from any vehicle dynamics and therefore universally valid. Metadata covers the following information:

- *Required Track Length* total required track length of a scenario
- *Required Lane Width* required lane width of a scenario
- *Required Radius* required curve radius of a scenario
- *Curvature* maximum curvature on a track
- *Curvature change* maximum change of curvature on a track
- *Heading* maximum Heading within a scenario
- *Velocity* maximum velocity reached within a scenario
- *Acceleration* maximum and minimum lateral and longitudinal acceleration within a scenario

This information may be used for a classification of different scenarios. In addition the derived metadata from a scenario represent track and vehicle requirements for MRTD.

V. Evaluation

The evaluation of the generated trajectories is split into two parts. The first part covers geometric referencing and quality of the OpenDrive database analysis. The second part compares the generated trajectories to a driving simulation.

A. Geographical Referencing

As mentioned in Section IV-A, the used road network was created on building plans. In order to evaluate the digital road network according geographical correctness and accuracy GPS-position, measurements were made. A vehicle equipped with a differential GPS localization unit was used to capture the vehicle position on different road lanes. The captured GPS data was firstly used to fine tune the geographic reference of the OpenDrive database. In addition, the captured GPS lane-positions of the two outer lanes from the oval are displayed as black lines on figure 8. The way-points of the generated trajectory of Overtaker (red rhombuses) overlap the captured GPS lane-positions before and after a lane-change maneuver almost identical.

B. Scenario Simulation

The generated Trajectories in this work strictly retain on a scenario description. In order to evaluate the offset between a generated target trajectory and a vehicle driven trajectory,

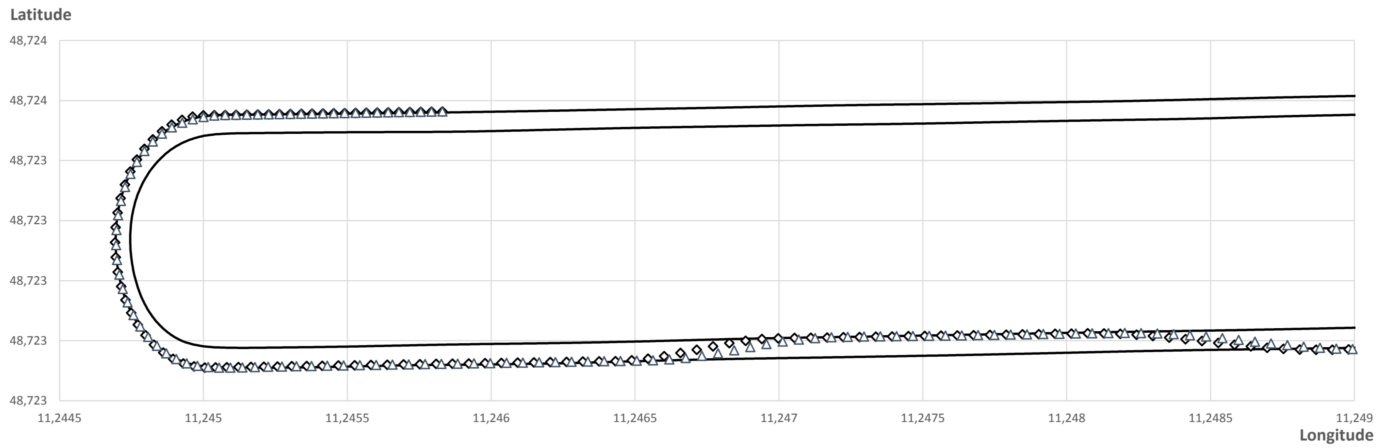


Fig. 8. Generated trajectory way-points of Overtaker (rhombuses) and simulated way-points of Overtaker (triangle)

a driving simulation is used. Virtual Test Drive (VTD)[3] is a driving simulation software with focus on environment (traffic, pedestrians, world) effects. VTD uses OpenDrive databases as static simulation-environment. Since VTD does not support the OpenScenario format yet, the example scenario was remodeled in a similar VTD specific dialect. During simulation the action trigger times and positions of ego and Overtaker are logged. Table III lists the simulated action trigger times.

TABLE III
SIMULATED ACTION TRIGGER TIMES

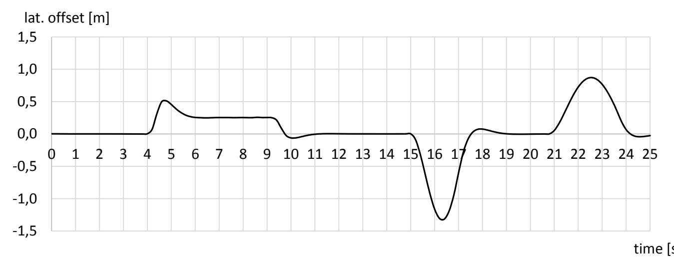
Name	Start Condition	Start	End	Duration
Speed 75	Pos. World	$9.74s$	$11.48s$	$1.74s$
Lane left	rel. Dist. 1	$14.94s$	$16.94s$	$2.0s$
Lane right	rel. Dist. 2	$20.68s$	$23.68s$	$3.0s$

Compared with the calculated trigger times from the maneuver sorting task the acceleration-maneuver in simulation is triggered $70ms$ earlier then in the trajectory generation. This timing-offset is caused by the fact, that the VTD driver model cuts the wide oval curve on entry and exit. As a result, the timing-offset of the first lane-change maneuver should be constant. The lessen timing-offset ($58ms$) comes from overshooting and adjusting at curve exit. The timing-offset of the second lane-change maneuver ($78ms$) depends on the lane-change trajectory. VTD's scenario dialect does not support a detailed lane-change definition. As a result, the performed lane-change trajectory of the driver model is not known. Figure 8 illustrates the trajectories of Overtaker. The triangles represent the simulated trajectory course, the red rhombuses the generated trajectory course.

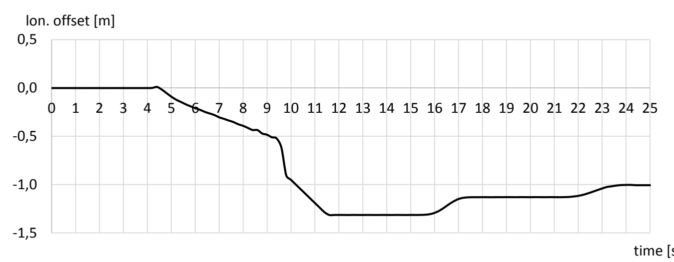
In detail the simulated and generated trajectories of Overtaker are evaluated regarding lateral and longitudinal offset. Since the ego entity stays in lane and velocity over simulation-time a detailed evaluation is not insightful.

Figure 9a displays the total lateral-offset over simulation-time. The offset between $4s$ and $12s$ cause from VTD driver model cuts at the wide oval curve on entry and exit. The

[3]Virtual Test Drive (by Vires GmbH) Driving simulation application - URL: https://vires.com/vtd-vires-virtual-test-drive/



(a) Lateral-offset



(b) Longitudinal-offset

Fig. 9. Lateral- and longitudinal-offset between simulated and generated trajectory of Overtaker

offsets amplitudes at $16s$ and $22s$ occur from different lane-change trajectories. The first lane-change trajectory duration is set to $2.0s$ the second lane-change trajectory lasts $3.0s$. For this reason the amplitudes differ from another.

Figure 9b displays the total longitudinal-offset over simulation-time. From entry of the oval's curve at $4.5s$ the offset increases approximately constant caused by the cutting of curves. At $9.74s$ the acceleration-maneuver causes a further increasing offset. This is explained by the VTD vehicle physics and the uniformly accelerated motion of the trajectory generation. The stagnating offset at $16s$ and $22s$ result from different lane-change trajectories.

VI. Conclusion

This paper presents a unified approach to a OpenScenario based trajectory generation. Scenarios in the form of Open-Scenario and road-networks in the form of OpenDrive are combined to derive global target trajectories for every dynamic scenario entity. The introduced architecture of trajectory generation matches spatial and temporal extracted information into three subsequent levels of a trajectory specification. The iteration through the route planning task, the maneuver sorting task and finally the trajectory generation results in a way-point based global target trajectory specification. The simulation results demonstrates the aberration between a generated target trajectory and a vehicle physics and driver model driven vehicle motion simulation.

Future work will focus on an automated route planning task based on road geometry and lane attributes. Mapping a predefined scenario on a given test track is essential for MRTD. Furthermore lateral and longitudinal maneuver actions needs to be defined as polynomials of fifth or higher order for the purpose to use generated trajectories as target motion control of real vehicles in the context of MRTD.

Acknowledgment

This work was supported by the Federal Ministry for Economic Affairs and Energy in line with the research project PEGASUS.

References

[1] G. Berg, V. Nitsch, and B. Färber, "Vehicle in the loop," in *Handbook of Driver Assistance Systems*, H. Winner, S. Hakuli, F. Lotz, and C. Singer, Eds. Switzerland: Springer International Publishing, 2016, ch. 10, pp. 199 –210.

[2] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, "Defining and substantiating the terms scene, situation, and scenario for automated driving," Institute of Control Engineering, Technische Universitt Braunschweig, Tech. Rep., 2015.

[3] M. Dupuis, *OpenDRIVE Format Specification Rev 1.4*, VIRES Simulationstechnologie GmbH, 2015.

[4] *ISO 8855:2013 Road vehicles - Vehicle dynamics and road-holding ability*, 2013.

[5] F. Warmerdam, "proj.4 geographic projection definition," 2017. [Online]. Available: http://proj4.org

[6] M. Dupuis, "Openscenario - bringing content to the road," in *2nd OpenSCENARIO Meeting*, 06 2016.

[7] J. X. Feng and J. H. R. et al., "Study on intelligent vehicle lane change path planning and control simulation," in *IEEE International Conference on Information Acquisition*, 2006, pp. 683–688.

[8] M. Y. and, "An experimentally confirmed driver longitudinal acceleration control model combined with vehicle lateral motion," *Vehicle System Dynamics*, vol. 46, pp. 129–149.

[9] S. Zhang, W. Deng, Q. Zhao, H. Sun, and B. Litkouhi, "Dynamic trajectory planning for vehicle autonomous driving," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2013, pp. 4161–4166.

[10] M. R. Zofka, F. Kuhnt, R. Kohlhaas, C. Rist, T. Schamm, and J. M. Zllner, "Data-driven simulation and parametrization of traffic scenarios for the development of advanced driver assistance systems," FZI Research Center for Information Technology, Karlsruhe, Germany, Tech. Rep., 2015.

[11] X. Li, Z. Sun, Q. Zhu, and D. Liu, "A unified approach to local trajectory planning and control for autonomous driving along a reference path," in *International Conference on Mechatronics and Automation*, 2014, pp. 1716–1721.

[12] A. Rupp and M. Stolz, "Survey on control schemes for automated driving on highways," *Automated Driving,*, pp. 43 –69, 2017.

[13] J. M. Snider, "Automatic steering methods for autonomous automobile path tracking," Robotics Institute, Pittsburgh, PA, Tech. Rep., 2009.

[14] L. Wang, X. Zhao, H. Su, and G. Tang, "Lane changing trajectory planning and tracking control for intelligent vehicle on curved road," *SpringerPlus*, vol. 5, no. 1, pp. 1150 – 1168, 2016.

[15] H. Guan, J. Li, Y. Yu, M. Chapman, and C. Wang, "Automated road information extraction from mobile laser scanning data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 194 – 205, 2015.