

Coupling SUMO with a Motion Planning Framework for Automated Vehicles

Moritz Klischat, Octav Dragoi, Mostafa Eissa, and Matthias Althoff

Technische Universität München, Fakultät für Informatik, Lehrstuhl für Robotik, Künstliche Intelligenz und Echtzeitsysteme,
Boltzmannstraße 3, 85748, Garching, Germany. moritz.klischat@tum.de

Abstract

Testing motion planning algorithms for automated vehicles in realistic simulation environments accelerates their development compared to performing real-world test drives only. In this work, we combine the open-source microscopic traffic simulator SUMO with our software framework CommonRoad to test motion planning of automated vehicles. Since SUMO is not originally designed for simulating automated vehicles, we present an interface for exchanging the trajectories of vehicles controlled by a motion planner and the trajectories of other traffic participants between SUMO and CommonRoad. Furthermore, we ensure realistic dynamic behavior of other traffic participants by extending the lane changing model in SUMO to implement more realistic lateral dynamics. We demonstrate our SUMO interface with a highway scenario.

1 Introduction

Automated driving requires extensive testing due to the complexity of driving tasks and the variety of possible situations. Therefore, virtual testing is important in order to accelerate the development of automated vehicles and reduce development costs. An integral part of testing motion planning algorithms under realistic conditions is the realistic simulation of surrounding traffic participants. Several simulation frameworks exist for testing functionalities of driver-assistance systems or automated vehicles; however, most of them are traditionally based on predefined scenarios. On the other hand, many simulators only consider microscopic simulation of traffic without automated vehicles.

To test automated vehicles under more realistic conditions, we couple the traffic simulator SUMO [18] with our open-source software framework CommonRoad¹ for benchmarking and motion planning of automated vehicles. Through our interface, arbitrary motion planners can be tested with traffic participants controlled by SUMO. The integrated vehicle models in SUMO enable a closed-loop simulation with traffic participants reacting to vehicles controlled by the motion planner, which we refer to as *host vehicles* from now on. Our interface is able to control multiple host vehicles and thus is able to handle collaborative planning tasks. We also improve the realism of other traffic participants in SUMO by extending existing lane-changing models to generate physically-feasible trajectories through a lightweight implementation.

1.1 Related Work

1.1.1 Coupling of Traffic Simulators

In previous works, traffic simulators are coupled with driving simulators for testing manually-driven vehicles, see e.g., [4,6,20]. Less work exists on coupling automated vehicles with traffic simulations. One of the few works combines SUMO with the simulator Virtual Test Drive (VTD) for driver assistance

¹<http://commonroad.in.tum.de>

systems [22,23]. VTD is used for high-fidelity simulations of vehicles close to the host vehicle, whereas SUMO is used for less relevant traffic participants.

Simulators focusing on generating behaviors for learning-based approaches are CARLA [9] and AirSim [24]. These simulators include modeling of sensor data and use game engines like Unreal and Unity. The recent project Apollo [5] aims at providing a complete framework for the development of automated vehicles. Simulators from the robotics community such as Gazebo [13] and VRep [10] can also be used for simulating automated vehicles. For instance, Gazebo is coupled with SUMO in [12] to simulate traffic participants. Further work enabling the simulation of the perception of the host vehicle through a combination of robotic simulators can be found in [8,21,27].

1.1.2 Lane-Changing Models in SUMO

Modeling of longitudinal motion in SUMO is already in a mature state, implemented in a variety of car-following models [16]. Also, there already exists much work on modeling high-level lane changes with a strong focus on the decision making process [11,15,17]. While it is possible to specify the duration of a lane change, the resulting trajectories are not generally physically feasible. Since motion planning algorithms typically only consider physically-possible motions, this can cause false positive errors. Semrau et. al. [23] developed a sub-lane model that enables a more detailed representation of lateral motion, including lane changes. Since this model is based on a lateral discretization of the lane, it increases the complexity of the simulation with regard to the number of considered maneuvers. Furthermore, it is not fully compatible with other functionalities of SUMO.

1.2 Contributions

Most of the existing simulators are based on complex software frameworks and aim at covering the whole workflow, from perception to a high-fidelity physics simulation. In contrast, our goal is to provide an interface to a lightweight, open-source simulation environment that especially targets research projects on motion planning. Furthermore, our approach for synchronizing lane changes of host vehicles with SUMO yields more realistic behaviors of other traffic participants compared to previous work. By extending lane-changing models in SUMO, vehicles feature a more realistic lateral movement with continuous acceleration instead of constant lateral velocity only.

2 Coupled Simulation

We first introduce the motion planning framework based on CommonRoad for interfacing motion planners and then present the realized interface to SUMO afterwards.

2.1 CommonRoad

CommonRoad is a Python tool for reading CommonRoad [1] benchmark files which provides basic functionalities for motion planning of automated vehicles. CommonRoad benchmarks consist of a map, static and dynamic obstacles, and a planning problem defined by an initial state and goal regions for one or multiple cooperating vehicles. The same structure is present in a CommonRoad scenario object. Based on CommonRoad, we have developed a software framework² for motion planning, including a collision checker, reachability analysis [25], and set-based prediction of traffic participants [2,14].

2.2 Map Conversion

Prior to starting the simulation, a common road network needs to be defined in the CommonRoad and SUMO formats. Road networks in SUMO are represented by *edges* and *junctions*. Edges connect two junctions and consist of one or multiple lanes whose order is specified by indices; lanes are defined by

²<http://commonroad.in.tum.de>

the lane width and a reference path consisting of an array of vertices (polylines). At junctions, edges are linked via *internal edges*, denoting which outgoing lanes are reachable from which incoming lanes of respective edges.

In contrast, road networks in CommonRoad are represented by lanelets [7], which are defined by their left and right bounds modeled by polylines. Lanelets are connected via successor-predecessor relations, as well as via adjacency relations. A lanelet is denoted as the successor of another lanelet if the starting vertices of the successor coincide with the final vertices of the predecessor. As shown in Fig. 1, junctions or merging lanelets are modeled by having multiple successors connected to a lanelet. Lanes of a road are also defined implicitly by the left and right adjacency relations of lanelets.

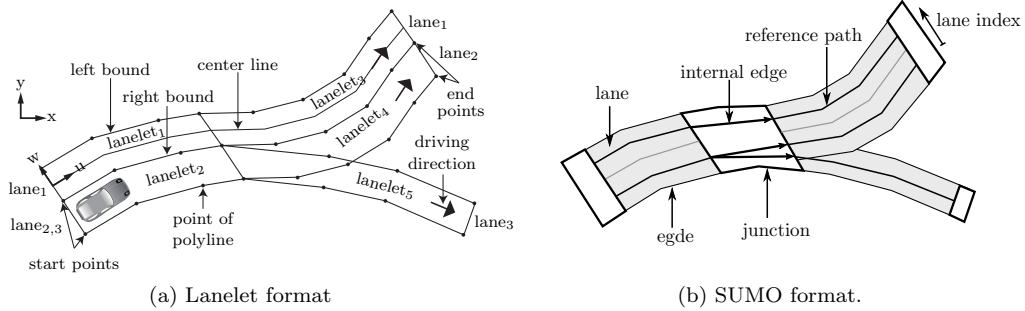


Figure 1: Comparison of map formats.

To represent the same road network in the SUMO and CommonRoad formats, a conversion from SUMO to CommonRoad is performed by first converting SUMO to the similar OpenDRIVE [19] format using the tool NETCONVERT from SUMO. From that format, a map in CommonRoad format is created using our converter from OpenDRIVE to lanelets [3]. Alternatively, OpenDRIVE maps can be used as a common basis for SUMO and CommonRoad maps. An example for a map conversion is shown in Fig. 2.

2.3 Vehicle Interface

The interface between the simulations in SUMO and CommonRoad is shown in Fig. 3. During the co-simulation, the states of traffic participants and the trajectories $x_H(t) \in \mathbb{R}^4$ of host vehicles are exchanged via commands from the TraCI API of SUMO. While the trajectories of host vehicles are controlled by one or several motion planners, all other vehicles are simulated in SUMO. In SUMO, the states of traffic participants $x_T(t) \in \mathbb{R}^4$ are defined by x- and y-position, velocity, and orientation. These states are read before every planning cycle of the motion planner with duration $\Delta t_C \in \mathbb{R}^+$ using TraCI and converted to a CommonRoad scenario object. Note that SUMO can be simulated with smaller time increments $\Delta t_S = \frac{\Delta t_C}{d}$, $d \in \mathbb{N}$ to enhance the accuracy of the simulation.

2.4 Synchronization of the Host Vehicles with SUMO simulation

In order to expose the host vehicles to other traffic participants, their planned trajectories for the subsequent time increment Δt_C are transferred through the interface. The interface interpolates the states of the host vehicles for every time increment Δt_S and sends them to SUMO. Since SUMO cannot include host vehicles, we override states of vehicles at each time increment Δt_S using the TraCI commands `moveToXY` for the position and orientation and `setSpeed` for velocity. Before activating the interface to the motion planner, the interface can initiate a simulation in which the host vehicles

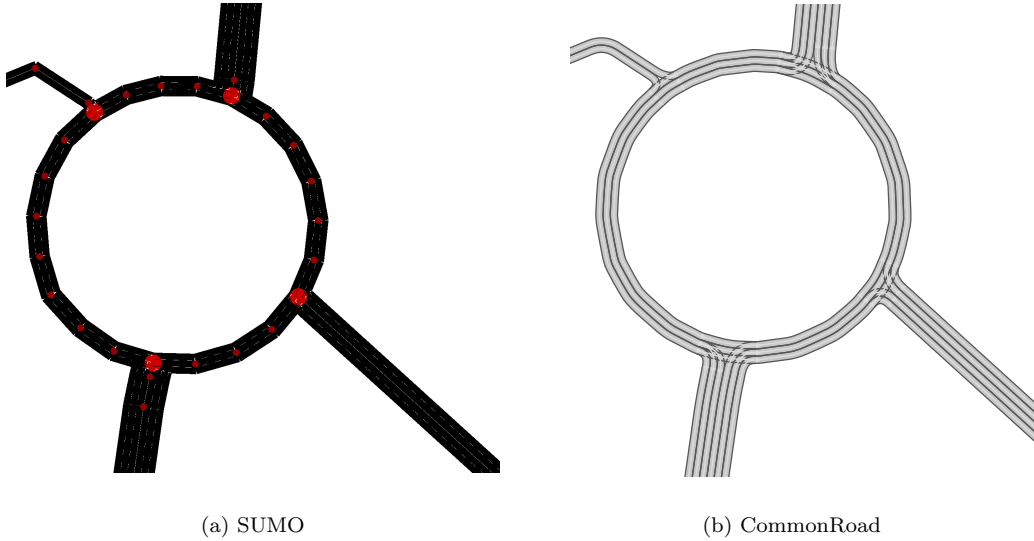


Figure 2: Roundabout in SUMO format converted to CommonRoad map format.

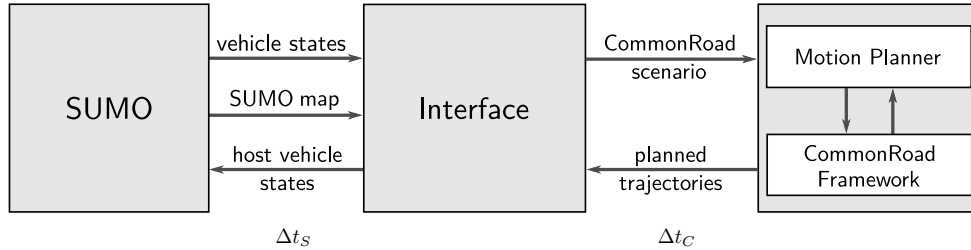


Figure 3: Synchronization between SUMO and Motion Planning Framework.

are controlled by SUMO for the time interval $t \in [0, t_{start}]$, so that the motion planner can focus on interesting situations.

2.4.1 Lane-Change Synchronization

One strength of SUMO is the integration of vehicle models implementing the reactions of traffic participants to their surrounding vehicles. When performing a lane change, vehicles in the target lane slow down to maintain a safe distance once they are notified about the lane change intent according to [11]. However, this implementation does not work in conjunction with the command `moveToXY`, because then other vehicles are not notified about the lane change and thus only react once the host vehicle has entered their lane.

In order to trigger the reactions of vehicles in the target lane immediately when the host vehicle starts a lane change, we use the TraCI command `changeLane` instead of `moveToXY` to forward the host vehicle to SUMO for the duration of the lane change. Through `changeLane`, other vehicles are notified about the lane change. As a consequence, the host vehicle is controlled by SUMO for the duration of the lane change, resulting in a trajectory $x_{H,s}(t) \in \mathbb{R}^4$. The approach presented below does not guarantee that the trajectory $x_{H,s}(t)$ conforms to the driven trajectory $x_H(t)$ planned by the motion

planner. However, in return for a more realistic simulation of traffic behaviors and a simple integration with existing vehicle models in SUMO, we accept a small error between both trajectories.

To ensure that the error $e(t) = \|x_{H,s}(t) - x_H(t)\|_2$ does not exceed a defined threshold $e_{max} \in \mathbb{R}^+$, the interface switches back to `moveToXY` once the condition $e(t) < e_{max}$ is violated (see line 12 in Algorithm 1). This condition is monitored throughout the lane change. To reduce the error of both trajectories, we extend `changeLane` and the lane changing models to additionally accept the lane-change duration t_l as an input, instead of using a global default duration. The duration t_l is computed by the function `DetectLaneChange($x_H([t_k, t_k + t_h])$)` that checks whether the planned trajectory intersects with a lane that is laterally adjacent to the currently occupied lane at any time during the planning interval $[t, t + t_h]$. By $t_h \in \mathbb{R}^+$ we denote the planning horizon of the motion planner.

Algorithm 1 Host Vehicle Synchronization with SUMO simulation

Require: MotionPlanner, TraCI instance, simulation duration t_{end} , planning horizon t_h , time increment of motion planner Δt_C

```

1:  $t \leftarrow 0$ 
2: performing_lane_change  $\leftarrow false$ 
3: while  $t < t_{end}$  do
4:    $x_H([t, t + t_h]) \leftarrow \text{MOTIONPLANNER.GETPLANNEDTRAJECTORY}(t_k)$ 
5:    $t_l, \text{lane\_change\_planned} \leftarrow \text{DETECTLANECHANGE}(x_H([t_k, t_k + t_h]))$ 
6:   if lane_change_planned = true then
7:     if performing_lane_change  $\neq true$  then
8:       TRACI.CHANGELANE( $t_l$ )
9:       performing_lane_change  $\leftarrow true$ 
10:    else
11:      if  $e(t) > e_{max}$  then
12:        TRACI.MOVETOXY( $x_H(t_k)$ )
13:        performing_lane_change  $\leftarrow false$ 
14:      end if
15:    end if
16:  else
17:    TRACI.MOVETOXY( $x_H(t_k)$ )
18:  end if
19:  TRACI.SIMULATIONSTEP()
20:   $t \leftarrow t + \Delta t_C$ 
21: end while

```

3 Continuous Lane Change in SUMO

In this section, we suggest an improvement for lane-change handling of vehicles in SUMO. The previous implementation [11] uses either instantaneous lane changes or constant velocity for lateral motion as illustrated in Fig. 4. We propose lane changes with a more realistic acceleration.

To be compatible with previously-developed lane-changing models, we extend the existing functionality of SUMO. Let us briefly recall the current implementation first. Lane-change behavior is defined in the class `MSAbstractLaneChangeModel`. Specific lane-changing models inherit from it and implement different behaviors regarding decision making related to lane changes. In the function `computeSpeedLat`, constant-velocity lane changes are implemented where at every time step $t_k = \Delta t_S \cdot k$, $k \in [0, n]$ of the lane change, the lateral speed $v(t_k) \in [-v_{max}, v_{max}]$ bounded by the maximal lateral velocity v_{max} is

computed. Using $v(t_k)$, the lateral distance is computed:

$$d(t_{k+1}) = d(t_k) + \Delta t_S \cdot v(t_{k+1}).$$

To integrate a lane change with continuous acceleration, we compute the lateral velocity $v_c(t)$ which considers the boundary conditions

$$v(0) = 0, \quad v(t_n) = 0, \quad \left. \frac{d}{dt} v(t) \right|_{t=0} = 0, \quad \left. \frac{d}{dt} v(t) \right|_{t=t_n} = 0$$

for enforcing a smooth transition at the beginning and end of the lane change. This is achieved by using the polynomial

$$v_c(t, d) = d c t^2 (t_n - t)^2, \quad (1)$$

where the constant c is chosen in order to travel the lateral distance $d \in \mathbb{R}^+$ during the lane change, i.e.,

$$\int_{t_0}^{t_n} v_c(t, d) dt = d.$$

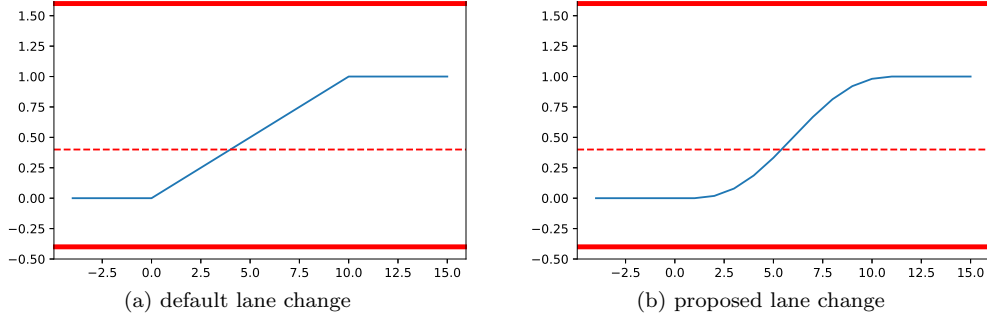


Figure 4: Continuous lane change compared to the previous implementation in SUMO.

3.1 Implementation

If the lane-change duration t_l is globally fixed as currently implemented in SUMO, the term $c t^2 (t_n - t)^2$ in (1) can even be computed offline for all t_k since it is independent of d . However, our implementation can also be applied to individually set lane-change durations, as mentioned in Section 2.4.1. In the base case where no lateral speed constraint is given, calling the function `computeSpeedLat` returns $v(t_k)$, and the current step $k \rightarrow k + 1$ is incremented accordingly as shown in Algorithm 2. Lane-changing models in SUMO can also constrain the lateral speed limit v_{max} . To comply with the implementations of these models, the velocity for the computation of $d(t_k)$ is limited (c.f. line 9) and a second iterator $m \in \mathbb{N}$ is introduced. The iterator m is only incremented after the traveled distance $d(t_{k+1})$ exceeds the distance without consideration of v_{max} at t_m (c.f. line 11 of Algorithm 2).

4 Numerical Example

We demonstrate the presented combination of our motion planning framework with SUMO through a highway scenario with multiple vehicles interacting with the host vehicle. We use a sampling-based motion planner [26], which utilizes the CommonRoad collision checker³ and a constant-velocity prediction for surrounding vehicles. Time increments of $\Delta t_C = \Delta t_S = 0.1$ s are used and the planning horizon t_h of the motion planner is 2 s.

³<http://commonroad.in.tum.de>

Algorithm 2 Smooth Lane Change**Require:** total lateral distance d , lane-change duration t_l , maximal velocity v_{max}

```

1:  $k = 0$ 
2:  $m = 0$ 
3: while  $m \cdot \Delta t_s < t_l$  do
4:   if  $v_{max} = \emptyset$  then
5:      $d(t_{k+1}) \leftarrow d(t_k) + \Delta t_s v_c(t_k, d)$ 
6:      $k \leftarrow k + 1$ 
7:      $m \leftarrow m + 1$ 
8:   else
9:      $d(t_{k+1}) \leftarrow d(t_k) + \Delta t_s \min(v_{max}, v_c(t_m, d))$ 
10:     $k \leftarrow k + 1$ 
11:    if  $d(t_{k+1}) \geq \sum_{i=1}^l v_c(t_m, d) \Delta t_s$  then
12:       $m \leftarrow m + 1$ 
13:    end if
14:  end if
15: end while

```

The trajectory planner of the host vehicle drives with constant velocity when starting a lane change at $t = 5.2$ s. To demonstrate the synchronization using `changeLane` presented in Algorithm 1, we compare the results of our algorithm to a synchronization with SUMO through `MoveToXY` only. As shown in Fig. 5 for using `MoveToXY`, the vehicle in the rightmost lane starts an overtaking maneuver, even after the host vehicle clearly started a lane change but is still in its lane. As a result, the host vehicle is forced to abort its lane change at $t = 6.1$ s in order to avoid a collision and only performs the lane change after letting the other vehicle pass. In contrast, the other vehicle does not perform a lane change when synchronizing the host vehicle with SUMO through using `changeLane`, because it is notified earlier.

5 Conclusions

We have shown how to couple SUMO with our motion-planning framework for testing automated vehicles. The proposed algorithm for synchronizing lane changes with SUMO yields a more realistic behavior of other traffic participants, and our extension of lane-change models yields more realistic motions of vehicles. The presented interface is published on <http://commonroad.in.tum.de>.

In the future, we plan to integrate the presented framework into our CommonRoad benchmark suite [1]. Our suite currently consists of scenarios with predefined trajectories for all traffic participants, enabling benchmarking of motion planners. Scenarios with interactive traffic participants will extend the scope of the benchmarks to further aspects of decision making.

Acknowledgements

The authors gratefully acknowledge financial support by the Central Innovation Programme of the German Federal Government under grant ZF4086007BZ8, by the BMW Group within the CAR@TUM project, and the project interACT within the EU Horizon 2020 programme under grant agreement No 723395.

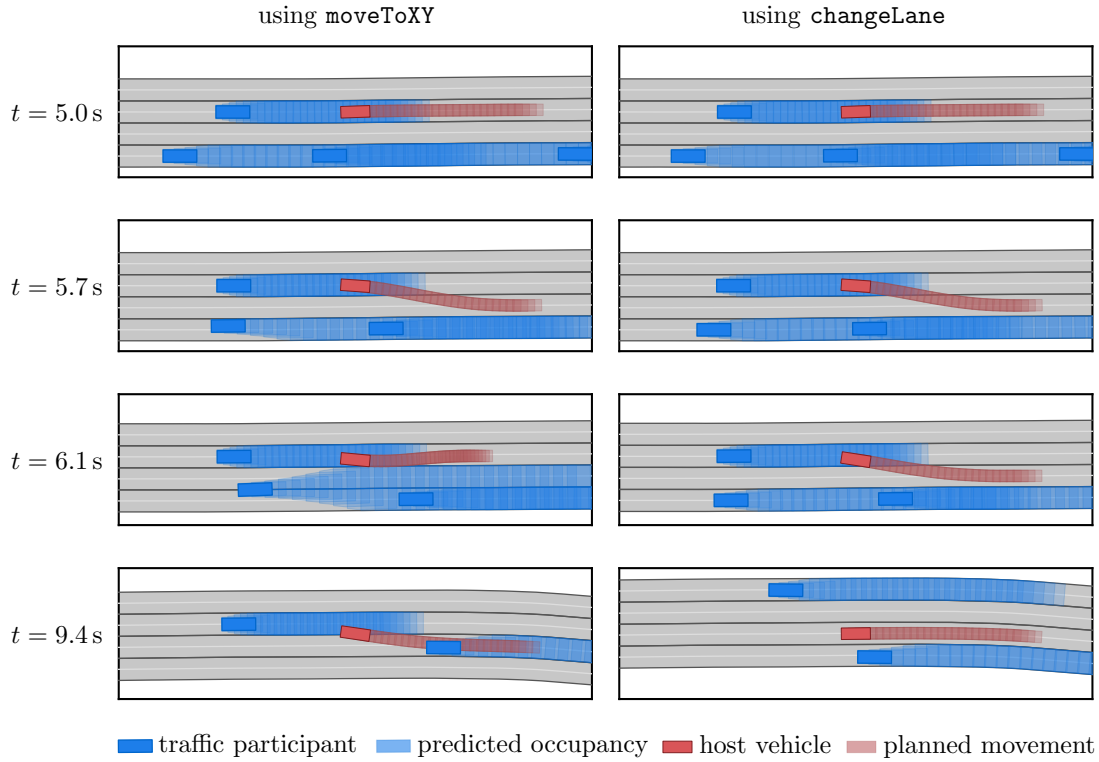


Figure 5: Comparison of lane changes realized by the functions `changeLane` and `moveToXY` in a highway scenario at different times t .

References

- [1] M. Althoff, M. Koschi, and S. Manzinger. CommonRoad: Composable Benchmarks for Motion Planning on Roads. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 719–726, 2017.
- [2] M. Althoff and S. Magdici. Set-based prediction of traffic participants on arbitrary road networks. *IEEE Transactions on Intelligent Vehicles*, 1(2):187–202, 2016.
- [3] M. Althoff, S. Urban, and M. Koschi. Automatic Conversion of Road Networks from OpenDRIVE to Lanelets. In *Proc. of the 2018 IEEE Int. Conf on Service Operations and Logistics, and Informatics*, pages 157–162, 2018.
- [4] M. Aramrattana, T. Larsson, J. Jansson, and A. N abo. Extended Driving Simulator for Evaluation of Cooperative Intelligent Transport Systems. In *Proceedings of the 2016 annual ACM Conference on SIGSIM Principles of Advanced Discrete Simulation*, pages 255–258, 2016.
- [5] Baidu Apollo Team. Apollo: Open Source Autonomous Driving, 2017.
- [6] M. Barthauer and A. Hafner. Coupling traffic and driving simulation: Taking advantage of SUMO and SILAB together. In *EPiC Series in Engineering*, volume 2, pages 56–44. EasyChair, 2018.
- [7] P. Bender, J. Ziegler, and C. Stiller. Lanelets: Efficient Map Representation for Autonomous Driving. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 420–425, 2014.
- [8] A. Cottignies, M. Daley, E. Newton, and H. Parker. rFpro & SUMO: The Road To A Complete Real-Time Simulation of Urban Environments for DIL, ADAS and Autonomous Testing. In *SUMO*

- 2017: *Towards Simulation of Autonomous Mobility*, 2017.
- [9] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An Open Urban Driving Simulator. In *Proc. of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.
 - [10] M. F. E. Rohmer S. P. N. Singh. V-REP: a Versatile and Scalable Robot Simulation Framework. In *Proc. of the Int. Conference on Intelligent Robots and Systems (IROS)*, 2013.
 - [11] J. Erdmann. Lane-changing model in SUMO. In *Proc. of the SUMO2014 Modeling Mobility with Open Data*, volume 24.
 - [12] M. Garzon and A. Spalanzani. An hybrid simulation tool for autonomous cars in very high traffic scenarios. In *2018 15th International Conference on Control, Automation, Robotics and Vision, ICARCV 2018*, pages 803–808, 2018.
 - [13] N. Koenig and A. Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2149–2154, 2004.
 - [14] M. Koschi and M. Althoff. SPOT: A Tool for Set-Based Prediction of Traffic Participants. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 1679–1686, 2017.
 - [15] D. Krajzewicz. Kombination von taktischen und strategischen Einflüssen in einer mikroskopischen Verkehrsflusssimulation. *Fahrermodellierung in Wissenschaft und Wirtschaft, 2. Berliner Fachtagung für Fahrermodellierung*, pages 104–115, 2009.
 - [16] S. Krauß. Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics. 1998.
 - [17] S. Li, Y. Wu, Z. Xu, and X. Lin. Improved lane-changing model for VANETS in sumo. *The 7th IEEE/Int. Conf. on Advanced Infocomm Technology*, pages 260–266, 2014.
 - [18] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. P. Flotterod, R. Hilbrich, L. Lucken, J. Rummel, P. Wagner, and E. Wiebner. Microscopic Traffic Simulation using SUMO. In *Proc. of IEEE Conf. on Intelligent Transportation Systems*, pages 2575–2582, 2018.
 - [19] H. G. M. Dupuis, M. Strobl. OpenDRIVE 2010 and beyond — status and future of the de facto standard for the description of road networks. In *Proc. of the Driving Simulation Conference Europe*, pages 231–242, 2010.
 - [20] T. Nguen That and J. Casas. An integrated framework combining a traffic simulator and a driving simulator. *Procedia - Social and Behavioral Sciences*, 20:648–655, 2011.
 - [21] J. L. F. Pereira and R. J. F. Rossetti. An integrated architecture for autonomous vehicles simulation. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing - SAC '12*, pages 286–292, 2012.
 - [22] M. Schiller, A. Knoll, M. Dupius, D. Krajzewicz, and A. Kern. Multi-resolution Traffic Simulation for Large-Scale High-Fidelity Evaluation of VANET Applications. In *SUMO User Conference 2015 - Intermodal Simulation for Intermodal Transport*, pages 17–36.
 - [23] M. Semrau, J. Erdmann, B. Friedrich, and R. Waldmann. Simulation framework for testing ADAS in Chinese traffic situations. In *SUMO 2016 Conference Proceedings*, pages 103–113, 2016.
 - [24] S. Shah, D. Dey, C. Lovett, and A. Kapoor. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In *Field and Service Robotics*, 2017.
 - [25] S. Söntges and M. Althoff. Computing the Drivable Area of Autonomous Road Vehicles in Dynamic Road Scenes. *IEEE Transactions on Intelligent Transportation Systems*, 19(6):1855–1866, 2018.
 - [26] M. Werling, S. Kammel, J. Ziegler, and L. Gröll. Optimal Trajectories for Time-Critical Street Scenarios using Discretized Terminal Manifolds. *International Journal of Robotic Research*, 31(3):346–359, 2012.
 - [27] M. R. Zofka, S. Klemm, F. Kuhnt, T. Schamm, and J. M. Zöllner. Testing and validating high level components for automated driving: simulation framework for traffic scenarios. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 144–150, 2016.