A SIMULATION BASED INTERACTION ANALYSIS OF AUTOMATED VEHICLES

Lennart Luttkus
Lars Mikelsons

Department of Mechatronics
University of Augsburg
Eichleitnerstr. 30
Augsburg, Germany
{lennart.luttkus, lars.mikelsons}
@informatik.uni-augsburg.de

Peter Baumann
Oliver Kotte

Robert Bosch GmbH

Robert-Bosch-Campus 1
Renningen, Germany
{peter.baumann5, oliver.kotte}
@de.bosch.com

ABSTRACT

Autonomous vehicles will increasingly shape the streetscape in the future. At the beginning of the dissemination, interactions between autonomous vehicles will occur without a predefined communication interface. In this work, a simulation framework is set up that allows the investigation of the interaction of autonomous vehicles. In particular, an EGO centric vehicle simulation is converted into a vehicle simulation with multiple detailed modeled vehicles (MultiEGO) using co-simulation methods. To identify the risk in an early stage by using simulation methods, a criticality assessment of traffic situations is defined. In a case study, critical traffic situations are analyzed using a detailed vehicle model and a search method based on a generalized traffic scenario. This investigation shows how demanding it is to find all the driving malfunctions and gives an insight into the challenges of developing and releasing autonomous vehicles.

Keywords: MultiEGO vehicle simulation, vehicle interaction, criticality assessment, release of autonomous vehicles

1 INTRODUCTION

Mobility is at the threshold of introducing autonomous systems that perform the driving task at least temporarily without driver monitoring. In the next few years, autonomy will gradually increase. At least at the beginning, interactions could occur between autonomous vehicles without a communication interface between them.

The fact that an introduction of autonomous driving function carries risks show several accidents, some with fatal consequences (Berboucha 2018) of autonomous vehicles in the USA. Early virtual investigations of critical situations potentially leading to an accident can help to increase the safety of autonomous systems in road traffic.

Testing and validating autonomous vehicles is a big challenge, thus different approaches are used in recent research efforts. A framework was introduced by (O'Kelly et al. 2018), which learns traffic behavior based on a database. Thereafter, the focus of the simulated scenarios is shifted to accident-prone situations. Their EGO vehicle is based on deep-learning algorithms and does not include advanced vehicle dynamic control functions. Advanced methods to find critical situations are presented in (Mullins, Stankiewicz, and Gupta 2017): Here a surrogate model is used to accelerate the search and the boundaries of performance mode switches are analyzed in a higher resolution. The sensitivity of a traffic situation could be presented by these performance mode switches, whereas the present paper analyses the sensitivity with respect to the starting position manually.

A rather abstract scenario description is given in (Menzel, Bagschik, and Maurer 2018). The simulation results of this contribution support their approaches and the contributed framework proposes a possible solution for their required conversion of concrete scenarios into simulations. Nevertheless, neither Menzel nor the paper at hand answers the question of how to find useful scenario parameters quickly. Further abstraction levels of the scenario generation for the vehicle test are given in (Bach, Otten, and Sax 2016), (Pütz et al. 2017) and (Bagschik, Menzel, and Maurer 2018).

This work focuses on the range of critical situations that cannot be derived from accident databases, namely critical situations between autonomous vehicles. Therefore, a self-centered (EGO) vehicle simulation is expanded to a vehicle simulation with multiple detailed modeled vehicles (MultiEGO) using co-simulation methods. A criticality assessment of traffic situations as well as the automation of a generalized traffic simulation enables the search for unknown, critical vehicle interactions and thus helps to reduce the number of them. The sensitivity analysis of the simulation results shows how challenging it is to find all vehicle malfunctions and provides insight into the difficulties of developing and releasing autonomous vehicles.

2 METHODS

The three methods used in this paper are: A MultiEGO simulation through a co-simulation middleware, the identification of critical traffic situations with an automated criticality assessment for the EGO vehicles and the automatic generation of new situations through a random configuration of a generalized traffic situation. In the following, each is described in a more detailed manner.

2.1 MultiEGO Co-Simulation

To the authors' knowledge, there is no solution to concurrently simulate many vehicles with a high degree of physical detail and near-series vehicle software functions including a virtual ECU in a virtual environment. Existing simulation tools, such as VTD VIRES (VIRES Simulationstechnologie 2018), CARLA (CARLA et al. 2017) and AirSim (Shah et al. 2017) already support multi-agent simulation but are far behind the level of detail in comparison to self-centric vehicle simulations in which only one vehicle is modeled in detail. Vehicles are very complex systems and oversimplified models could lead to insufficient simulation results. In order to simulate multiple complex modeled vehicles in a virtual environment, two different approaches can be followed: Either a multi-agent vehicle simulation is used and the individual agents are provided with a better vehicle model, or several EGO simulations are coupled.

In this work, several EGO vehicle models are coupled using co-simulation methods (Gomes et al. 2017). Therefore, the middleware Model.CONNECT (AVL 2019) is used to orchestrate the coupled vehicle models and to enable data exchange between them. The coupling of multiple EGO vehicles is described below as a MultiEGO vehicle simulation. The physics of the EGO vehicles used in this work is modeled in CarMaker (IPG Automotive 2019) and the control functions are either modeled or imported in Simulink (MathWorks 2019). To allow the simulation of autonomous driving on multilane highways under consideration of traffic, the EGO vehicles are equipped with the near-series control functions of an electronic stability program, an adaptive cruise control function, a lane-keeping system, as well as an early development state of a lane changing function. In addition, the EGO vehicle dynamic model was extensively validated against real-world test drives and measurements, whereas the correctness of the MultiEGO co-simulation has so far only been validated against experienced situations.

Figure 1 shows a simple exemplary driving scenario to be investigated with MultiEGO simulation. Three vehicles drive on a one-lane road and drive from left to right. The two white vehicles represent the EGO vehicles described above. The blue vehicle is a simple traffic object, which moves only in a longitudinal direction and follows a predetermined speed profile. A schematic radar cone of vehicle one and two symbolizes the autonomous driving ability.

Figure 1: Two EGO vehicles equipped with an adaptive cruise control following a traffic object

In order to couple self-centered simulation environments, it is necessary to not only exchange the relevant signals (position, velocity, etc.) of the corresponding ego vehicles but to ensure that all vehicles act as they would move in the same environment. Thus the relevant signals of all dynamic objects have to be exchanged, i.e. the simulations have to be synchronized. The environment of both instances is identical. Each instance calculates the behavior of one EGO vehicle, whereas the second instance (CM2) calculates the vehicle behavior of the non-EGO (traffic) vehicle and the behavior is imitated within the first instance (CM1) by copying the position and rotation of the traffic object from CM2 to CM1.

Figure 2 shows schematically the synchronization of the two EGO vehicle simulation environments. The environment of the first CarMaker instance (CM1) is shown above, the second CarMaker instance (CM2) can be seen below the red dashed line. Both instances have a white EGO vehicle and two blue traffic objects. CM1 only calculates the position of the own EGO vehicle and receives the position and orientation of the vehicles two and three through the middleware Model.CONNECT. CM2 calculates the position of its EGO vehicle as well as of the traffic object (T03) driving ahead. The position of vehicle one is received via Model.CONNECT. It corresponds to the position of the EGO vehicle of CM1.

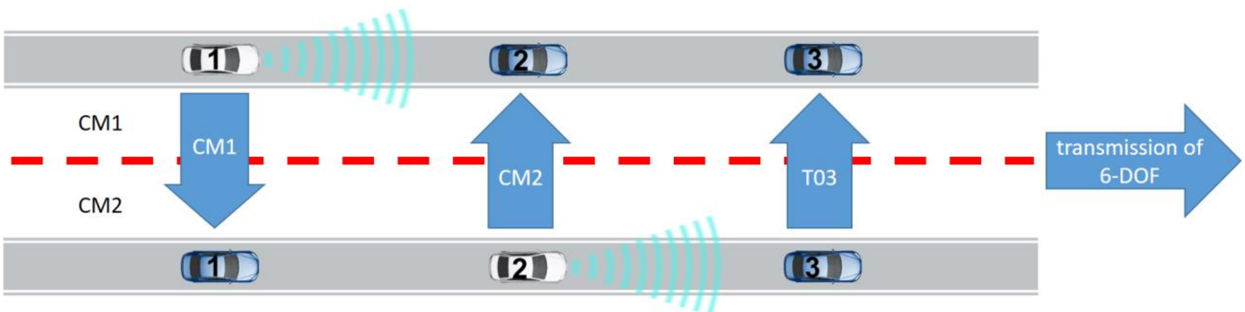

Figure 2: Synchronization of the CarMaker instances

The signal exchange between the CarMaker instances does not take place directly via Model.CONNECT, but CarMaker has an interface to Simulink, which is then connected to Model.CONNECT. Figure 3 shows this coupling between the three software tools.

Figure 3: Coupling between CarMaker, Simulink and Model.CONNECT

CarMaker simulates the EGO vehicle physics, the vehicle environment and the behavior of the traffic objects. Simulink is a signal flow based tool that simulates the control of the vehicle including drive train, brakes, and autonomous driving functions. Model.CONNECT is a tool-independent middleware, which typically allows the co-simulation of different simulation tools. In this case Model.CONNECT is used to connect two instances of the same simulation toolchain.

The presented co-simulation framework allows reusing the existing, highly detailed, self-centered vehicle simulation models while still expanding the system boundaries. In comparison, a multi-agent simulation (VIRES Simulationstechnologie 2018), (CARLA et al. 2017) is better in terms of scalability but does (currently) not allow the reuse of existing models. Note that even with the support of corresponding interfaces (e.g. FMI) integration of models with a high level of detail in such a way that the same results are produced is not a straight forward task because of a strong coupling between environment and vehicle (e.g. wheel-road contact). Moreover, a traffic simulation as presented in (Krajzewicz et al. 2002) focuses on the traffic flow and therefore, the level of detail of the traffic agents is insufficient for virtual validation.

The vehicles used in this context do not communicate explicitly and only recognize the behavior of other vehicles through their own idealized sensors. In addition, the effects of weather and road conditions as well as vehicle faults were not taken into account for the time being.

2.2 Criticality Assessment

The presented framework is supported by an automated criticality assessment, which enables the efficient identification of unknown hazard potentials. The assessment is based on (Hankey et al. 2016) and (Hoel, Wolff, and Laine 2018).

The criticality function is evaluated at each simulation time step and the resulting numerical value is accumulated over time.

$$\sum crit(t) = \sum crit(t-1) + crit(t)$$

In order to evaluate the criticality of a situation, three aspects are considered (Table 1): Safety violations with the highest criticality gradient followed by comfort limits and soft goals.

Table 1: Criticality assessment

Criterion	**Criticality gradient**
Safety violations, like collisions and leaving the roadway	+10
Acceleration exceeding comfort limits (longitudinal 3.5 m/s², lateral 2 m/s²)	+1.5
Secondary goals not achieved, like desired vehicle speed	+0.05

2.3 Generalized Highway Scenario

Discovering unknown malfunctions in highly complex systems, like autonomous vehicles, is a big challenge. One approach is the extensive virtual testing of the systems. In the following chapter, an approach for a randomized search of critical situations in a simulation is presented.

In order to detect unknown accident situations, a generalized initial situation is constructed, which generates different traffic situations by a random variation of the starting position. The starting position intervals are limited; therefore, the initial situation is always plausible. From situation start, the vehicles interact during a simulation period of 60 seconds and a potentially critical situation evolves.

This initial situation takes place on a three-lane, straight highway. The generalization of the traffic scenario is based on (Menzel, Bagschik, and Maurer 2018). Figure 4 shows an overview of the scenario. Seven vehicles are involved, including five traffic objects (blue) and two EGO vehicles (white). The traffic objects follow different longitudinal velocity trajectories and brake for vehicles driving in front on the respective lane (traffic vehicle behavior is defined in CarMaker). They do not move in a lateral direction and therefore do not change lanes. The starting position of all vehicles is variable within certain sections, whereby different starting situations are configured.

The behavior of the EGO vehicles is calculated in Simulink. The sensors in CarMaker provide the location of the road markings and the position and speed of all road users to the vehicle control functions in Simulink. Based on this sensor information, the EGO vehicles try to reach their target speed of 120 km/h and, if necessary, overtake slow-moving vehicles without endangering other road users. Thus the EGO vehicle behavior resembles a rudimentary highway pilot.

For each vehicle group, an explanation of the starting position and the vehicle behavior is shown in Figure 4. Within this description, the first line describes the driving task. The position-space along the route is then defined using the x coordinate. The y-coordinate defines in which lane the vehicle is initialized. The "randi([from, to])" function generates random integer values in the given parameter space to select the starting position between the lower limit "from" and the upper limit "to".

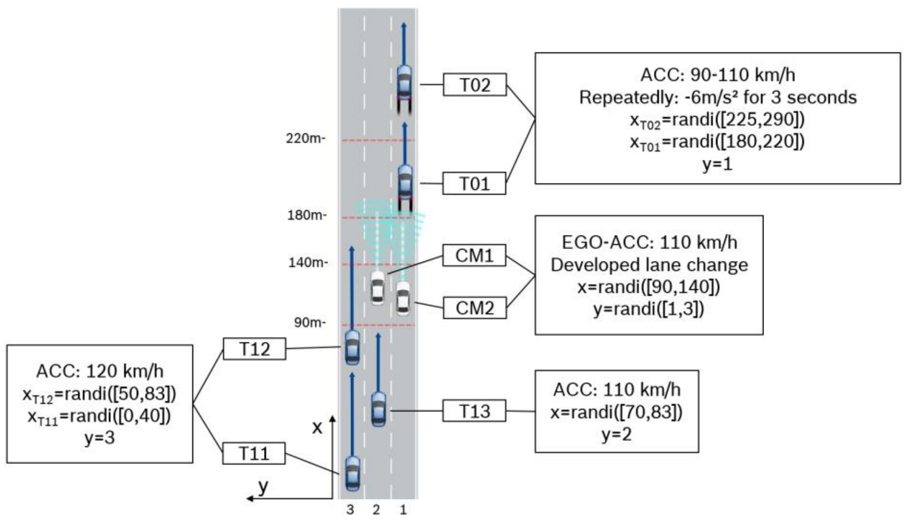

Figure 4: Basic situation of the randomized search for critical scenarios

This search strategy is rather straight forward and leads to enormous computation times. However, the approach followed here allows to validate the co-simulation based MultiEGO simulation and underlines the demand for more sophisticated search strategies as will be seen later. A potential candidate for such a strategy was recently presented in (O'Kelly et al. 2018).

3 RESULTS

The results of the MultiEGO simulation are presented below. First, a relatively simple scenario is analyzed with two EGO vehicles and the effects of the system boundary extension on the performance of the adaptive cruise control. This analysis shows the validity of the proposed method. Thereafter, a double lane change with an EGO vehicle interaction is analyzed. Subsequently, the results of the criticality assessment and the random scenario generation are presented. Due to the complexity of the traffic situations, supporting videos are linked below the results (Luttkus 2019).

3.1 Simple Scenario – Over Braking

Using the MultiEGO simulation method from the previous chapter, a simple scenario on the highway is realized in order to show the validity of the approach. The sequential braking behavior is examined when several EGO vehicles drive in succession with adaptive cruise control.

Figure 5 shows a snapshot of the simulation visualization of IPGMovie (visualization of CarMaker environment). The snapshot focuses on the second CarMaker instance. Thus, the middle vehicle is the EGO vehicle. The last vehicle in driving direction gets its position from the EGO vehicle of the first CarMaker instance, as can be seen in Figure 2. At the very front is the blue traffic object with the given speed trajectory. The traffic object accelerates to 100 km/h and brakes with -2 m/s² repeatedly. The two EGO vehicles try to achieve their target speed of 120km/h while maintaining an appropriate safety distance to the vehicle ahead. This vehicle functionionality is generally known as Adaptive Cruise Control (ACC).

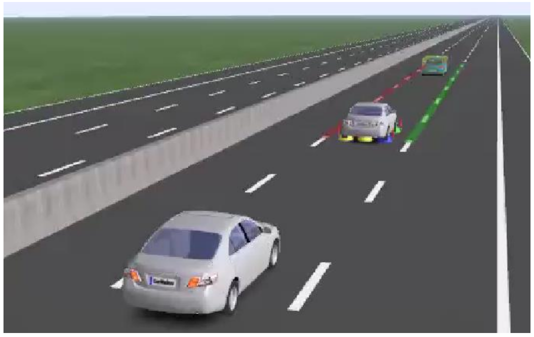

Figure 5: MultiEGO simulation within the CarMaker visualization

For a simulation with a duration of 60 seconds, figure 6 shows the position, the speed and the acceleration of the three vehicles. Traffic object T03 is shown in blue, the middle EGO vehicle CM2 in black and the third vehicle CM1 in red. The decisive finding is hidden in the acceleration.

From the results a wave behavior in the maximum deceleration depending on the position of the cars can be seen for example at a time of 25-27 s: For a given deceleration of -2 m/s² of the traffic object, the following car CM2 has to brake with a deceleration of -3.5 m/s². The last car in the series CM1 then already brakes with -4 m/s² exceeding the limit of -3.5 m/s² felt as comfortable. This behavior repeats at ~50 s.

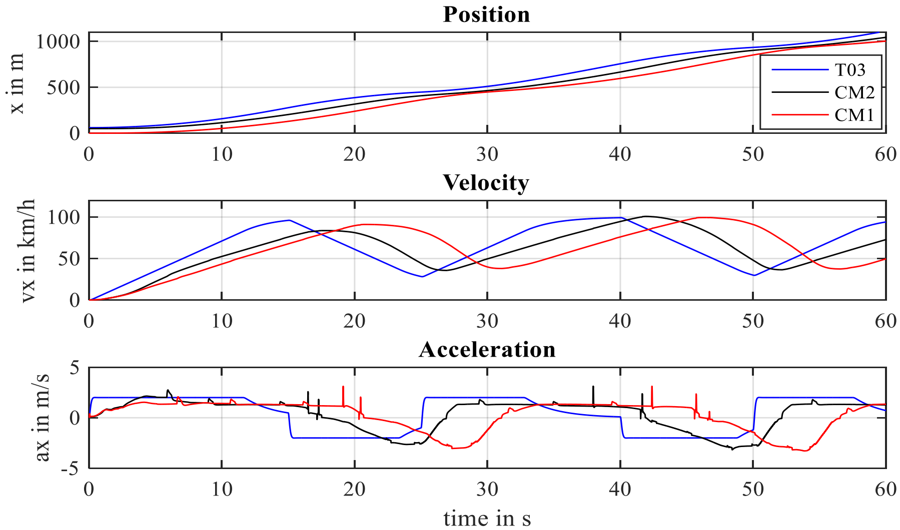

Figure 6: Trajectories of the simple MultiEGO simulation

With the presented framework such effects like over braking can be virtually investigated and the results can be reintegrated to the development process of these functions. Although over braking can be detected in an EGO vehicle simulation by the reaction of the first EGO vehicle to the preceding traffic object, the MultiEGO simulation significantly enhances it. Furthermore, the used method makes it possible to observe an actual interaction between two or more EGO vehicles. A video of the adaptive cruise control fleet simulation can be found here: https://youtu.be/VMh-sns6sD8

There is also a video of three EGO vehicles following a traffic object, where the last ego vehicle has to break even harder: https://youtu.be/FKHByT0CMmM

3.2 Simple Scenario – Double Lane Change Abortion

The previous chapter showed the validity of the MultiEGO co-simulation approach by reproducing an effect from real world, but the EGO vehicle interaction was unidirectional from the preceding EGO vehicle CM2 to the following vehicle CM1. In this section, a bidirectional interaction is presented by deploying a traffic situation of a simultaneous lane change abortion of two EGO vehicles.

Figure 7 presents the schematic lane change abortion in three succeeding scenes of a three-lane, straight highway with the direction of travel from left to right. In the following, the traffic situations are described from a perspective in direction of travel. As before, traffic objects which follow a given velocity trajectory along the roadway and do not change lanes are shown in blue and the EGO vehicles in white. The arrows in front of the vehicles indicate the direction of travel and the planned trajectory. The length of the arrows represents the speed of the vehicles qualitatively. In the situation I there are four vehicles on the three-lane highway. The two EGO vehicles are located in the right and left lane, which drive faster than the traffic object in the middle lane. The second traffic object is located in the right lane and drives slower than the EGO vehicles. This results in situation II, where the middle lane is not occupied anymore and appears to be free for both EGO vehicles. The right EGO vehicle begins the overtaking maneuver, while the left EGO vehicle decides to change its lane to the right. Both vehicles try to get to the same lane at the same time. Thus, they react with a lane change abortion, which requires the right EGO vehicle to brake hard for the slower traffic object in front (situation III).

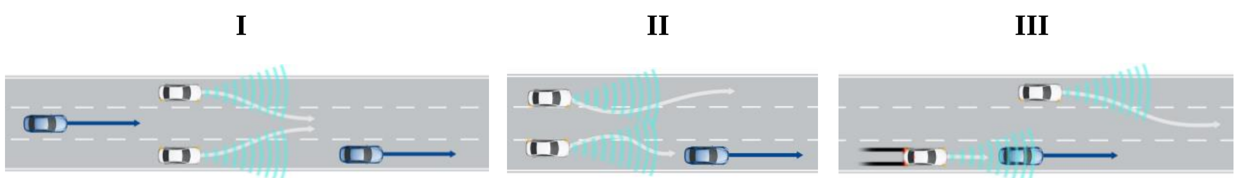

Figure 7: Three snapshots from the double lane change abortion

The visualization of this situation is shown in Figure 8.

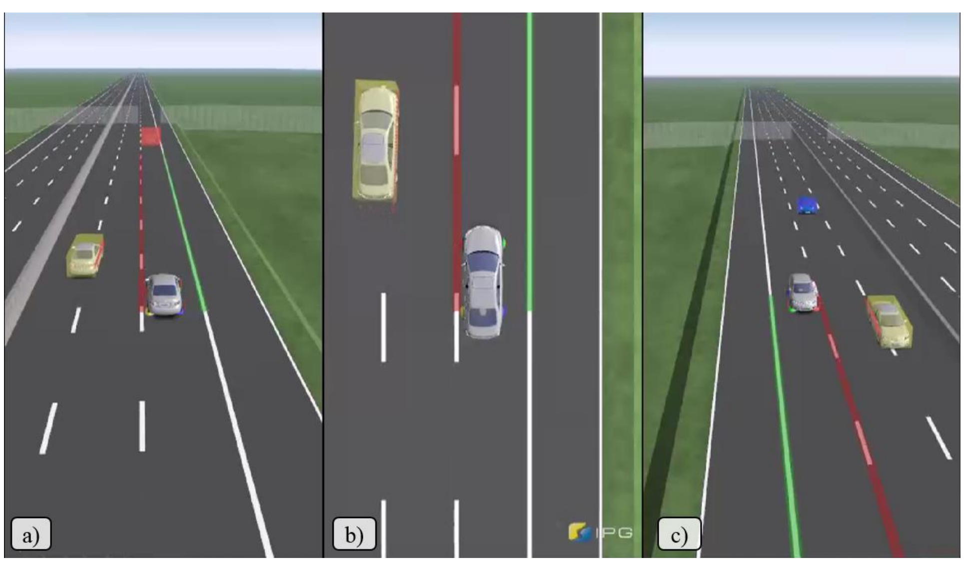

Figure 8: Three different perspectives of snapshot II from the simulation visualization. Section a) aligns the view in driving direction, section b) shows a bird's-eye view and section c) shows the rear view.

Note that such an EGO vehicle interaction cannot be observed within a self-centered vehicle simulation environment. Also, real testing of such a situation is a complex task. Two autonomous vehicles, two conventional vehicles, a three-lane motorway and a lot of time to find the best starting positions and velocity trajectories would be required to recreate it.

3.3 Generalized Highway Situation – Dependency between Initial Starting Position and Criticality

After ensuring the validity of the MultiEGO simulation, the generalized traffic scenario from chapter 2.3 was used to simulate a variety (600) of slightly modified highway scenarios. The criticality assessment identified four percent of the simulated scenarios as prone to accidents, due to the immature lane change algorithm. Disregarding the reason for the accident, the dependency between the starting position modification and the occurrence of an accident was further investigated.

It is obvious that there is a connection between the time elapsed since the initial situation until an accident and the distance by which the starting position can be varied in order to continue to observe the same accident. However, the gradient of the situation size is bigger than expected. The elapsed time and starting position correlation are shown in Table 2 for three different scenarios. Each scenario is a variation of the generalized highway situation and therefore results in different durations until the accident occurs. The situation size describes the possible starting position variation of the accident causing EGO vehicle, to ensure, that the same observed accident, reoccurs. If more time passes before an accident, then the development of the scenario is much more sensitive to changes in the starting position.

Table 2: Correlation between time of crash and possible starting position variation

Time of crash in seconds	Situation size in meter
10	2.5
35	0.003
45	<0.0002

In one scenario configuration, a collision occurred after 45 seconds. If the starting position of the colliding EGO vehicle was changed by more than 0.1 mm in any direction, it was a collision-free scenario. The criticality, time and starting position diagram in Figure 9 shows the thinness of this critical situation with respect to the starting position. It was found by examining another situation with a greater position resolution.

The situation is so narrow, because the used lane change algorithm makes a discrete decision, whether to change a lane or not. In this specific situation, a very small difference in the starting position enlarges quickly over simulation time due to the interaction of the EGO vehicle with other vehicles. When it finally comes to the decision if the lane is changed or not this difference leads to different decisions and thus to the critical situation and the adjacent, collision-free situations.

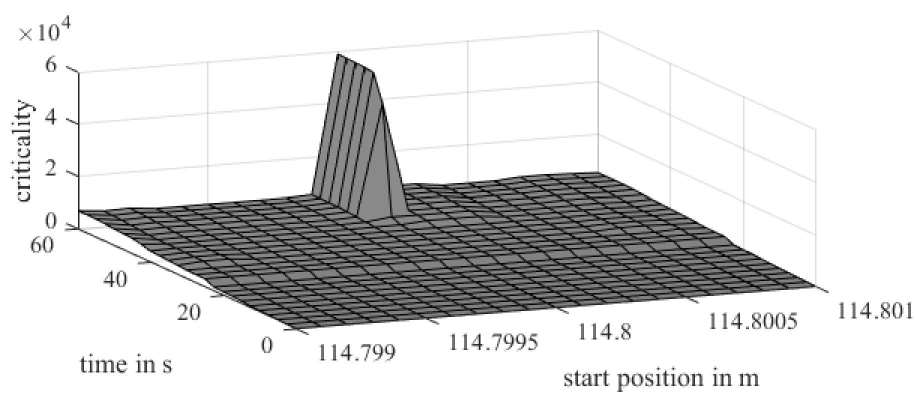

Figure 9: Thinness of the accident after 45 seconds with respect to the starting position

A comparison of the adjacent accident-free situation can be seen in the following videos:

Comparison between 114.7999 m and 114.8 m: https://youtu.be/6YuIpaBGX2o

Comparison between 114.8001 m and 114.8 m: https://youtu.be/XFv1oiTj6_0

The videos show an accident, which was not anticipated by the lane change algorithm designer and presents a real danger. This situation will be an important test case for future lane change developments.

It is challenging to find the right compromise. On the one hand, it is desirable that the traffic situations have as much time as possible to develop into vehicle configurations that were previously unknown and unpredicted to be critical. On the other hand, it is convenient if the accident situation depends as much as possible on the starting position so that the accident situation can be analyzed precisely by means of starting position variation.

Nevertheless, the simulation results show that real-world testing cannot be sufficient and the same holds for simulating generalized traffic situations. The presented generalized highway scenario has approximately 1.2 trillion different configurations when starting positions are discretized with a step size of one meter. While simulating 60 virtual seconds lasts 720 seconds, it would take 28 million years to simulate every different scenario configuration on our i7-3770 CPU due to the complexity of the simulation. Note, that the road, vehicle type, desired speeds, etc. are not even varied. Therefore, advanced methods like rare-event detection, surrogate modeling, etc. are required.

4 CONCLUSION

This paper shows that it is possible to reuse existing EGO vehicle simulation models in a MultiEGO context, by applying co-simulation methods. In addition, it has been exposed that the over braking vehicle behavior, which is known from real-world driving experiments, is also found in the simulation. Furthermore, it has shown that such a framework can assist in finding critical situations. The presented framework, including an automated criticality assessment and the random situation configuration, supports finding new, previously unknown and critical traffic situations.

Future work includes improving the criticality assessment and its smoothness to enable an optimizer based approach. For example, the time-to-collision could be included (Lee 1976). Furthermore, the generalized traffic scenario should include a greater variety of roads, vehicles, trajectories, etc. In addition, the randomized search could be improved by using an optimization-based approach or a rare event simulation. Moreover, a reasonable approach could be to use an adaptive level of detail for the simulation. Thus, interesting situations could be found faster and thereafter examined in detail. Additionally, it would be desirable to unify an interface in vehicle environment simulations to integrate different EGO vehicles in a multi-agent simulation.

Autonomous vehicles move in an open, not exactly specified context. As a result, the system boundaries must be broader and the resulting overall system simulation involves ever greater challenges. This paper illustrates these challenges and shows initial approaches to an overall system simulation in an open context.

REFERENCES

AVL. 2019. "Model.CONNECT™ - IODP Portfolio - Avl.Com." https://www.avl.com/-/model-connect-. Accessed January 17, 2019.

Bach, Johannes, Stefan Otten, and Eric Sax. 2016. "Model Based Scenario Specification for Development and Test of Automated Driving Functions." *IEEE Intelligent Vehicles Symposium, Proceedings* 2016-Augus (IV): 1149–55.

Bagschik, Gerrit, Till Menzel, and Markus Maurer. 2018. "Ontology Based Scene Creation for the Development of Automated Vehicles." In *IEEE Intelligent Vehicles Symposium*, IV:1813–20.

Berboucha, Meriame. 2018. "Uber Self-Driving Car Crash: What Really Happened." https://www.forbes.com/sites/meriameberboucha/2018/05/28/uber-self-driving-car-crash-what-really-happened/#2d11dc404dc4. Accessed January 15, 2019.

CARLA, Alexey Dosovitskiy, German Ros, Felipe Codevilla, and L Antonio. 2017. "CARLA : An Open Urban Driving Simulator." In *Conference on Robot Learning (CoRL)*, 1:1–16.

Gomes, Cláudio, Casper Thule, David Broman, Peter Gorm Larsen, and Hans Vangheluwe. 2017. "Co-Simulation: State of the Art." *ArXiv Preprint ArXiv:1702.00686*.

Hankey, Jonathan M., Miguel A. Perez, Julie A. McClafferty, and Virginia Tech Transportation Institute. 2016. "Description of the SHRP2 Naturalistic Database and the Crash, Near-Crash, and Baseline Data Sets." *The Strategic Highway Research Program 2 - Transportation Research Board of The National Academies*.

Hoel, Carl-Johan Johan, Krister Wolff, and Leo Laine. 2018. "Automated Speed and Lane Change Decision Making Using Deep Reinforcement Learning." *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* 2018-Novem: 2148–55.

IPG Automotive. 2019. "CarMaker | IPG Automotive." https://ipg-automotive.com/products-services/simulation-software/carmaker/. Accessed January 17, 2019.

Krajzewicz, Daniel, Georg Hertkorn, Peter Wagner, and Christian Rössel. 2002. "SUMO (Simulation of Urban MObility) - an Open-Source Traffic Simulation." In *4th Middle East Symposium on Simulation and Modelling*, 4:183–87.

Lee, David N. 1976. "A Theory of Visual Control of Braking Based on Information about Time-to-Collision." *Perception* 5 (4): 437–59.

Luttkus, Lennart. 2019. "Chair of Mechatronics - University of Augsburg - YouTube." https://www.youtube.com/channel/UCWk1RO51V3ck72Ee5HvXPkQ/featured. Accessed February 5, 2019.

MathWorks. 2019. "Simulink - Simulation Und Model-Based Design - MATLAB." https://de.mathworks.com/products/simulink.html. Accessed January 17, 2019.

Menzel, Till, Gerrit Bagschik, and Markus Maurer. 2018. "Scenarios for Development, Test and Validation of Automated Vehicles." *IEEE Intelligent Vehicles Symposium, Proceedings* 2018-June: 1821–27.

Mullins, Galen E, Paul G Stankiewicz, and Satyandra K Gupta. 2017. "Automated Generation of Diverse and Challenging Scenarios for Test and Evaluation of Autonomous Vehicles." *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 1443–50.

O'Kelly, Matthew, Aman Sinha, Hongseok Namkoong, John Duchi, and Russ Tedrake. 2018. "Scalable End-to-End Autonomous Vehicle Testing via Rare-Event Simulation." *Conference on Neural Information Processing Systems (NeurIPS 2018)* 32 (NeurIPS).

Pütz, Andreas, Adrian Zlocki, Julian Bock, and Lutz Eckstein. 2017. "System Validation of Highly Automated Vehicles with a Database of Relevant Traffic Scenarios." In *12th ITS European Congress*.

Shah, Shital, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. 2017. "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles." *Field and Service Robotics Conference 2017 (FSR 2017)*, May, 1–14.

VIRES Simulationstechnologie. 2018. "VIRES Simulationstechnologie GmbH." https://vires.com/. Accessed January 17, 2019.

AUTHOR BIOGRAPHIES

Lennart Luttkus is a Ph.D. student at the University of Augsburg. He holds a Master of Science in Mechanical Engineering from the University of Duisburg-Essen. His research interests lie in the virtual validation of autonomous vehicles in an open context. His email address is lennart.luttkus@informatik.uni-augsburg.de.

Lars Mikelsons is professor for Mechatronics at the University of Augsburg. He holds a diploma in Mathematics and a Ph.D. in Engineering, both from the University of Duisburg. After his dissertation, he worked in industry in the field of virtual engineering in the automation and automotive domain. His current fields of activities include virtual validation, mixed real/virtual prototypes and hybrid (mixed data and physical) modeling. His email address is lars.mikelsons@informatik.uni-augsburg.de.

Peter Baumann is a Ph.D. Student at the University of Duisburg-Essen. He holds a Master of Science in Engineering Cybernetics from the University of Stuttgart. His research interests include spatially distributed co-simulation, and in particular, the numerical challenges that emerge from it. His email address is peter.baumann5@de.bosch.com.

Oliver Kotte works as a research engineer for Robert Bosch GmbH. He holds a Ph.D. degree from ETH Zürich, a Master of Science degree from The Georgia Institute of Technology, and a Dipl.-Ing degree from the University of Stuttgart in systems and control engineering disciplines. His expertise lies in both the modeling and the tooling of complex co-simulations. His email address is oliver.kotte@de.bosch.com.