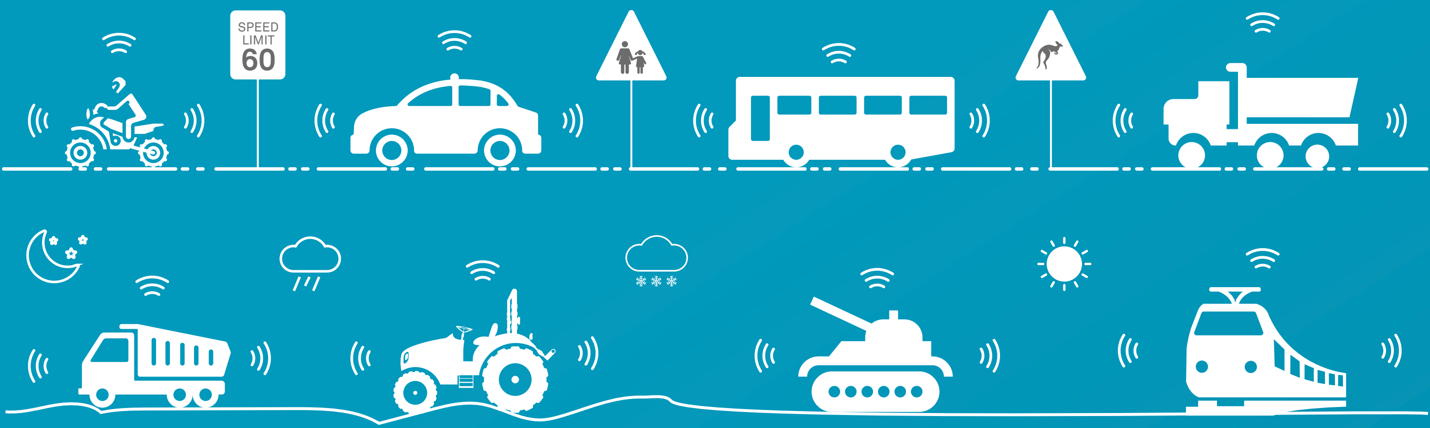
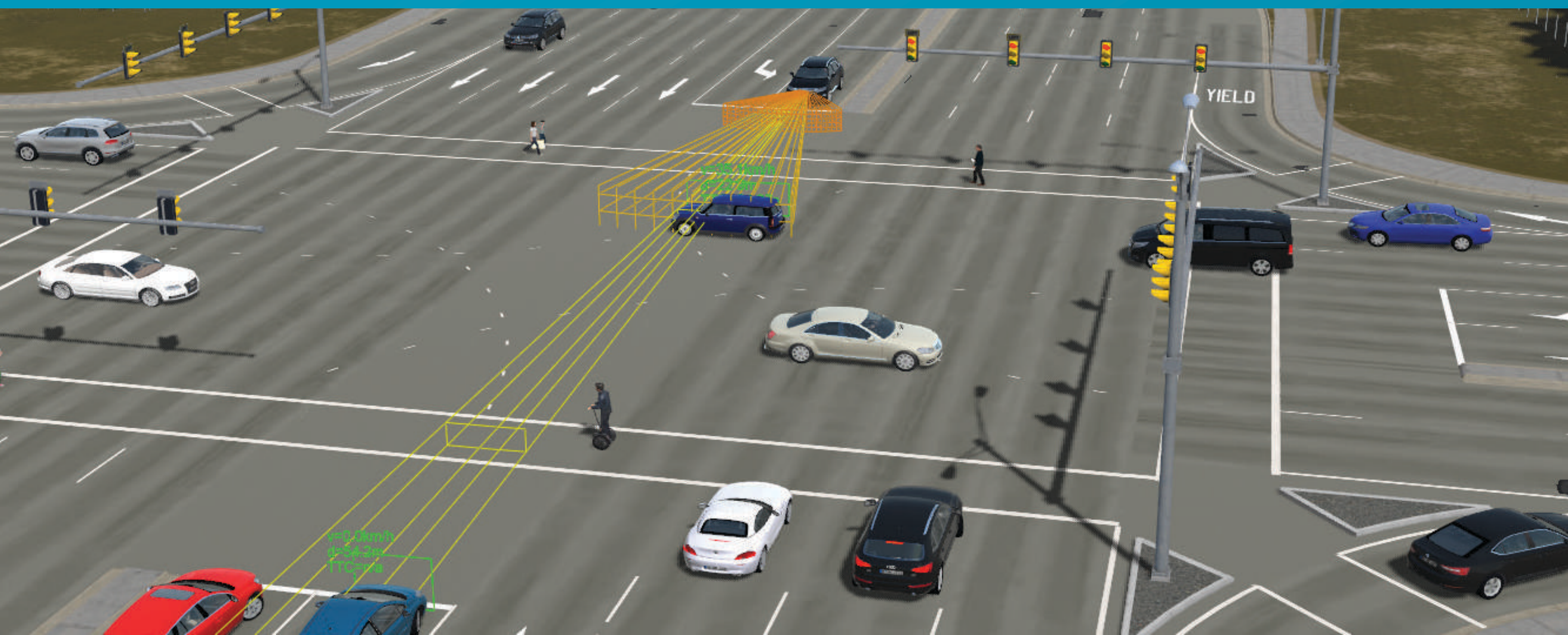
Virtual Test Drive 2020

Ensuring Safer ADAS and Autonomous
Vehicle Design using Simulation





Authors:
Marius Dupuis
Dr. Luca Castignani
Dr. Keith Hanna

HEXAGON | MSC Software


Smart Autonomous Mobility
Enable. Accelerate. Deploy.
Enabling customers of all types and sizes to accelerate and deploy a bold autonomous mobility vision.
mscsoftware.com/autonomous
HEXAGON
empowering an autonomous future

Table of contents

Foreword

The Autonomous Vehicle Industry has come a long way in the past decade or so. Truly futuristic progress has taken place where self-driving vehicles are concerned. A lot of resources and testing are being spent on road testing which has been deemed as a very important part of the process. However, road testing alone is simply not adequate and not easible when it comes to ensuring the safety of humans and vehicles on road. It would take us about a century to complete the testing of one self-driving vehicle model if we only rely on physical testing.

Every year, 1.24 million people die in traffic accidents and 50 million are injured worldwide (WHO data, 2013), and over 90% of these collisions are due to human error. The deployment of Level 5 autonomous vehicles can potentially save hundreds of thousands of lives every year. Simulation has a big role to play in accelerating the development of this sector. Industry leaders across the globe including companies like General Motors, BMW, Audi, Volkswagen are leveraging virtual testing to validate and to verify Advanced Driver Assistant Systems (ADAS) and autonomous driving systems.

This is where MSC Software wants to make a significant contribution through solutions like VTD where we experiment every relevant driving condition, including system faults and errors. Companies like Waymo is running a fleet of 25,000 virtual cars 24/7, simulating 13 million kilometers per day. Simulation is critical to us for achieving billions of miles of testing for automated driving development.

With our e-book on autonomous driving, we hope the readers will gain valuable insights on recent Research and Development in the self-driving space. The book also endeavors to shed some light on why autonomous driving is important and what is realistically achievable in the next 5 to 10 years.

Dr. Luca Castignani
Head of Autonomous Mobility Strategy, MSC Software

Volkswagen Group: Leveraging VIRES VTD to Design a Cooperative Driver Assistance System

By **Dr. Kai Franke, Development Online Driver Assistance Systems, Volkswagen AG**

“A combination of ADTF, VTD, and OMNet++ allows us to do a host of experiments to test and validate cooperative driver assistance systems.”

Figure 1. VIRES VTD is an open platform for developing Advanced Driver Assistance Systems

‘Cooperative Driving’ has attracted significant attention in recent years within automobiles. In order to increase the quality of signals, the availability and the perception range as well as to decrease the latency and the probability of total failure, advanced perception systems consisting of camera, radar and lidar systems with vehicle-to-vehicle (V2V) communication are required. Moreover, V2V communication enables advancements from individual to cooperative decision making. Advanced driver assistant systems (ADAS), which determine their behavior in due consideration of the definition of “cooperative behavior”, are capable of increasing the total utility of a group of cooperative vehicles. However, several technical issues have to be resolved on the way to deploying the first cooperative driver assistance systems (CDAS) on our public streets. For example, handling the misuse of the communication channel and the consideration of unequipped vehicle are some of the key challenges for cooperative driving.

This article focuses on a test framework for CDAS, which can be leveraged to master the complexity of distributed driver assistance systems (DAS) during the development process. A combination of ADTF (the application prototyping framework within the Volkswagen group), VTD (Figure 1, a simulation tool-chain from VIRES GmbH) and OMNet++ (an open-source component-based network simulator) allows us to do a host of experiments to test and validate cooperative driver assistance systems.

Simulation Framework

Since the development of CDAS requires at least two interacting vehicles, the implementation and the validation of the system necessitates a flexible test framework for connected vehicles. Figure 2 gives an overview of the proposed architecture used within this project (reference 1). The detailed description of interfaces and functionality follows hereinafter.

A. Application

A simplified illustration of the CDAS is composed of a planner implemented in ADTF (Automotive Data and Time triggered Framework) and a controller for each involved vehicle. There are three relevant interfaces of the application. The first interface represents the environmental model and the current vehicle state provided by the simulation gateway. The second interface to the network enables the communication

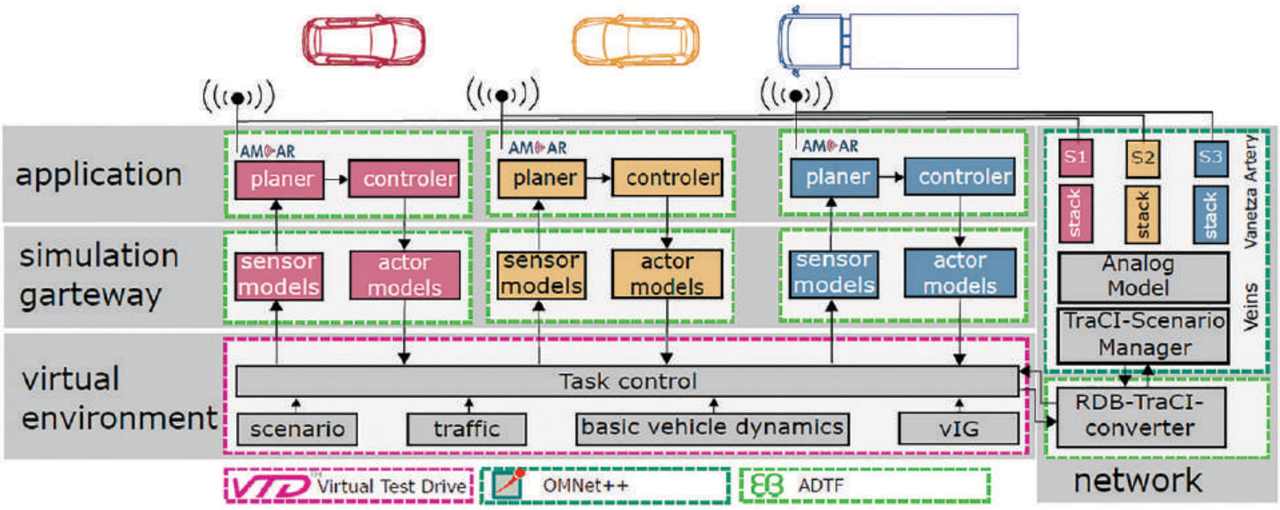


Figure 2. Overview of the VW simulation framework

between vehicles. The last interface to the simulation gateway realizes the controllability of the vehicle.

B. Simulation Gateway

For each vehicle (here an example is shown for three vehicles in Figure 3), the simulation gateway fulfills among others two tasks: the modeling of the perception (environment and vehicle state), and the reaction to controller outputs. The interface for the vehicle state includes, but is not limited to, the velocity, the longitudinal and lateral acceleration, and the steering wheel angle.

C. Virtual Environment

The software Virtual Test Drive (VTD) developed by VIRES provides the virtual environment we used.

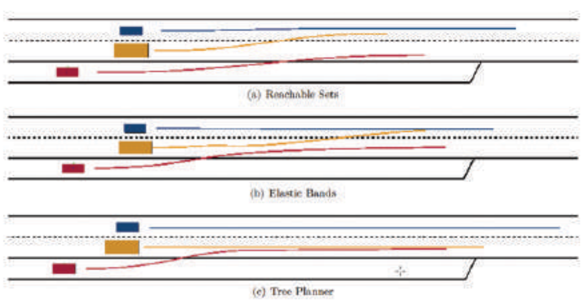


Figure 3. Results of the planning methods for the merging scenario of three vehicles

The central component is the task control coordinating additional modules with the help of the module manager. Additional modules are the scenario with roads and vehicle information, the traffic, the basic vehicle dynamics for internally controlled vehicles and the Image Generator (IG). The virtual environment transmits its information via Ethernet on the Real Time Data Bus (RDB) interface. Furthermore, the Simulation Control Protocol (SCP) interface provides a mechanism for operating the simulation.

D. Network

The network simulation can emulate the communication of the application via for example, ETSI ITS G5. In order to simulate the signal damping, the analog model uses information about line of sight and distances between the communicating vehicles. The RDB interface and the map of VTD (*.xodr format) contain the required information.

Simulation Results

A. Decentralized Decision Making

An example of a merging scenario on a highway is chosen to demonstrate the usability of the decentralized decision making (see Figure 3). The red vehicle wants to merge onto the highway, while the two lanes are blocked by a truck (yellow) and another vehicle (blue). The lane width amounts to three meters each.

Figure 4. Scenario for collision avoidance on oncoming traffic

> **“The proposed simulation framework allows a flexible modular combination of software components.”**

Three different planning algorithms generate offers for the merging scenario. It can be seen that the planning methods (a) and (b) recommend a lane change for the truck, while method (c) makes the truck stay in its lane. Planner (b) starts the lane change later than planner (a). Planner (c) solves the conflict situation by accelerating the truck and merging maneuver of the red vehicle behind the truck. The diversity of the offers results from different discretizations and different evaluation criteria. In order to demonstrate the decentralized decision making process, a cost function based on a fuzzy logic is applied, which enables a continuous prioritization between comfort, driving enjoyment, efficiency, and safety. Table I illustrates the different preferences of each vehicle. The truck focuses on efficiency, the red vehicle prefers driving enjoyment, and the blue vehicle prioritizes comfort.

Each vehicle comes to a different evaluation or rating of the offers, because of the varying preferences. The varying preferences can be caused by different brands, different vehicle models (sedan, van or SUV), or by an online driver monitoring system. Table II shows the results of the evaluation of each plan by each vehicle and the result of the two proposed selection criteria. The selected solution (bold) represents the compromise of the solution options. Plan (c) is selected by the sum criterion and plan (a) is selected by the squared sum criterion.

B. Closed Loop Simulation

The closed loop or hardware-in-the-loop simulation enables a study to evaluate the control error considering communication and calculation latencies of planner, controller, and vehicle dynamics. An application for collision avoidance exemplarily demonstrates the closed loop performance (Figure 4).

As an initial scenario, a driver starts an overtaking maneuver on a rural road. The driver misjudges the situation and the danger of a collision with the oncoming traffic arises. An active safety system detects the danger and starts/triggers the cooperative maneuver planning. The detection criterion could also be the time to collision (TTC). The TTC is calculated as the quotient of distance and relative velocity. The calculated cooperative maneuver plan targets the completion of the overtaking maneuver of the red vehicle and a deceleration of the truck and the blue vehicle.

Figure 5 shows a flagging characteristic. The longitudinal controller has a linear increasing controller error. This is caused by a constant velocity error. A possible reason is that the longitudinal controller does not model the decelerating influence of a steering maneuver. In this case the vehicle decelerates stronger than planned. The lateral controller shows an overshooting. The vehicle stays with 50 *cm* maximum

Table I. Preferences of the cooperative vehicle to parametrize the cost functions

	red vehicle	yellow truck	blue vehicle
safety	1	1	1
comfort	0.2	0.5	1
efficiency	0.1	1	0.4
driving enjoyment	1	0	0.1

Table II. Individual costs and combined costs for a cooperative merging scenario

	plan (a)	plan (b)	plan (c)
red vehicle	0.59	0.85	0.80
yellow truck	0.20	0.41	0.13
blue vehicle	0.44	0.17	0.17
sum $k_{i,v}$	1.23	1.44	**1.11**
sum $k^2_{i,f}$	**0.44**	0.93	0.69

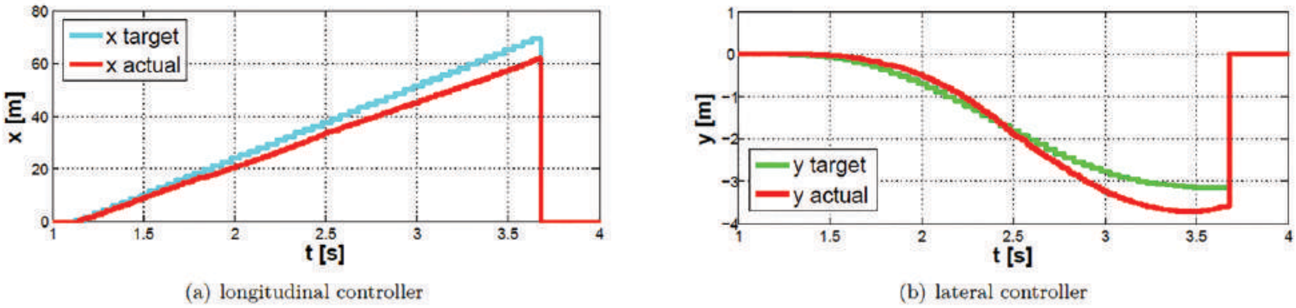


Figure 5. Results of closed loop simulation

controller error in a safe condition (stays on road, no collisions with obstacles). This error is caused by latencies and systematic errors in the feed forward controller. However, systematic errors, difference between vehicle dynamics model and inverted model in the feed forward controller, are made on purpose. A perfect vehicle dynamics model in the feed forward controller is impossible in reality, because of for example changing loads, changing wheel characteristics, and changing surface etc. Further controller adaption will be done with the help of real field test.

Conclusions

A new CDAS (Cooperative Driver Assistance System) imposes new requirements on simulation methods. The high degree of connectivity and interaction of the applications disable a development and later validation without considering the multi-directional influence. The proposed simulation framework allows a flexible modular combination of software components and considers modeling of perception, communication, and controlling of several vehicles in a virtual environment.

Reference

1. “A Cooperative Driver Assistance System: Decentralization Process and Test Framework” by Kai Franke, Reza Balaghiasefi, Michael Düring, Hendrik-Jörn Günther, Proc. 7th Tagung Fahrerassistenzsysteme Conf., 2015.
2. Source: https://pdfs.semanticscholar.rg/6361/393b-8c4067f857bf68f8ea7b79588eb19aba.pdf

NATO: Leveraging Adams and Luciad to Assess Mobility Characteristics of a Military Ground Vehicle

By **Hemanth Kolera-Gokula,**
Product Marketing Manager, MSC Software

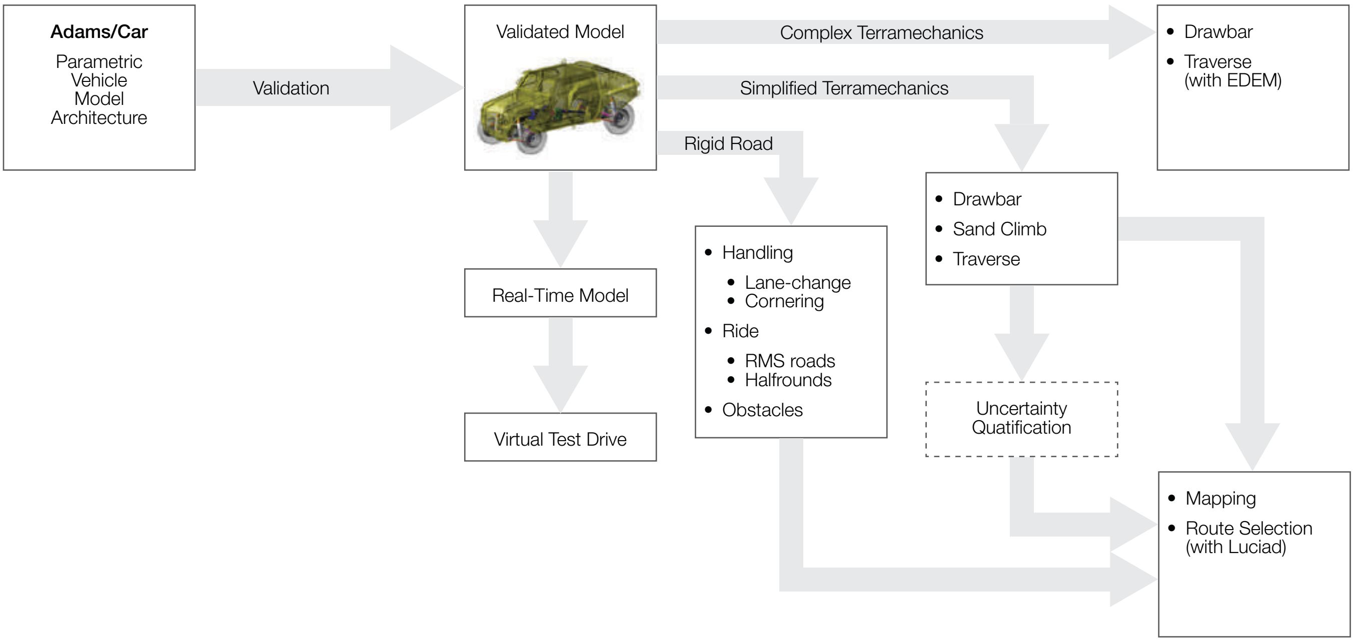


Figure 1 Mobility modeling for the NG-NRMM program

The mobility of a ground vehicle can be the difference between mission success and mission failure on the battlefield. In today's defense environment, there is a need to create rapidly deployable, highly mobile vehicle platforms that operate reliably across various terrain and road types. Vehicle simulation capabilities for assessing performance for different environmental conditions and operational scenarios have increased significantly in recent years.

In support of the Next Generation NATO reference mobility model (NG-NRMM) project for assessing existent CAE mobility analysis capability different facets of the Hexagon product portfolio were used to asses and visualize the mobility of the FED-Alpha, a Fuel Efficient Demonstrator vehicle (Figure 1). Adams models was created and validated against real-world calibration data by a team comprising of Eric Pesheck, Venkatesan Jeganathan, Tony Bromwell, Aniruddh Matange and Paspuleti Rahul Naidu to support this effort. These validated models were then used to accurately predict vehicle performance under a variety of on- and off-road operational scenarios. Select results from these investigations were integrated into Luciad, part of the Hexagon Geospatial portfolio, via a customized application for visualization and mobility mapping. Additionally, real-time compliance of the Adams model to support various autonomous and "Hardware-in-the-Loop" scenarios was demonstrated.

Creating and Validating the Adams Model

Adams Car, a solution vertical in the Adams portfolio focused on the modeling and simulation of vehicle assemblies and sub-systems was used to create a full-vehicle model of the FED-Alpha. Adams Car uses a template-based approach to model building; Reusable parametric templates of sub-systems such as chassis, tires, powertrain etc. can be populated with vehicle data and integrated to create a full vehicle assembly as shown in Figure 2. Typical model data includes design hard points, part mass properties, and component compliance characteristics. Adams Car allows detailed component representations, such as flexibility, friction, or frequency-dependent behavior where warranted. The level of fidelity and detail employed in the model was based on the simulation intent and available design data.

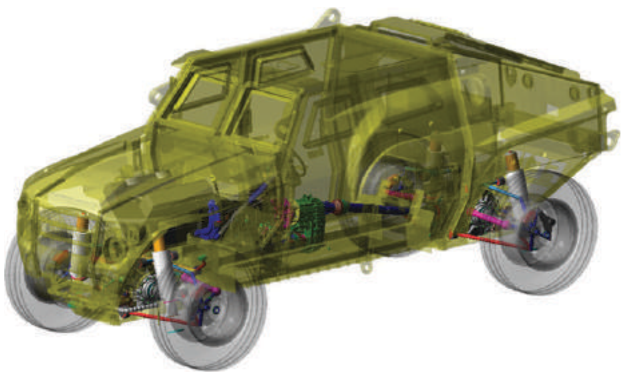


Figure 2 FED-Alpha Vehicle Assemble in Adams

The accuracy of the model was validated by comparison against data gathered from various vehicle test events. Metrics related to vehicle behavior, dynamics and ride quality were compared for model validation.

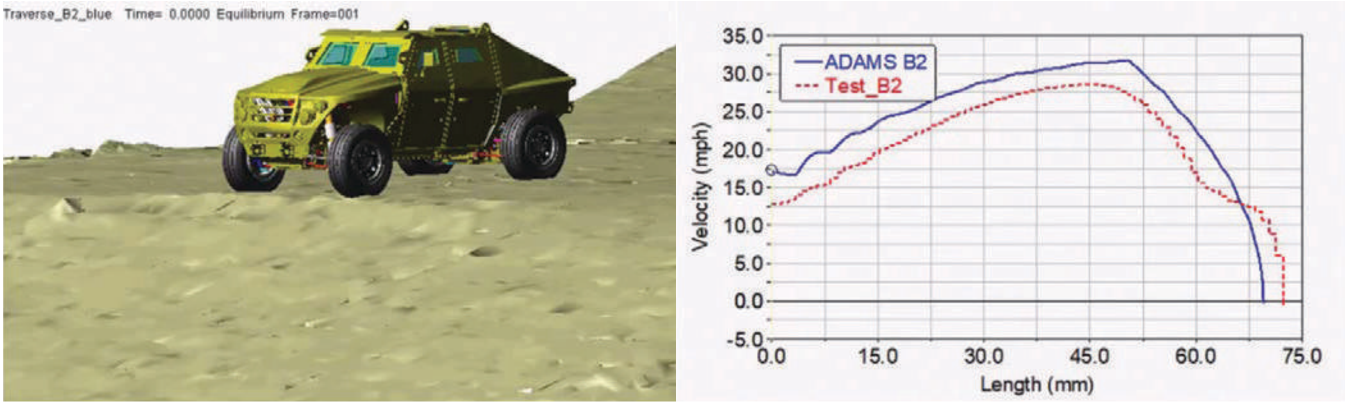


Figure 3 Adams Model Validation against Test Data

Predicting Vehicle Performance Using the Adams Model

The validated Adams model was then used to simulate various vehicle events to evaluate vehicle performance and mobility. These events consisted of both on and off road usage to mimic real battlefield scenarios. Typical military on-road evaluation events such as double lane change, indicating limit handling performance, half rounds, indicating ride quality, and step climbs, indicating obstacle navigation ability, were stimulated, with good agreement to test.

Evaluation of off-road performance is critical since achievement of certain mission objectives could require operation over unprepared terrain. Of crucial importance in off road modeling are the representation of the terramechanics; the soil properties and the interaction between the tire and the soil surface. Simple and detailed models for the description of the terramechanics were utilized in this initiative. Simple terramechanics models use empirical relationships, based on experimental measurements, to predict the response of deformable terrain to vehicle operation. These methods are computationally efficient, and were used to assess vehicle performance for well-defined draw-bar and hill-climb analyses. In addition, these methods were applied to scanned terrain geometry for more generalized off-road performance analyses.

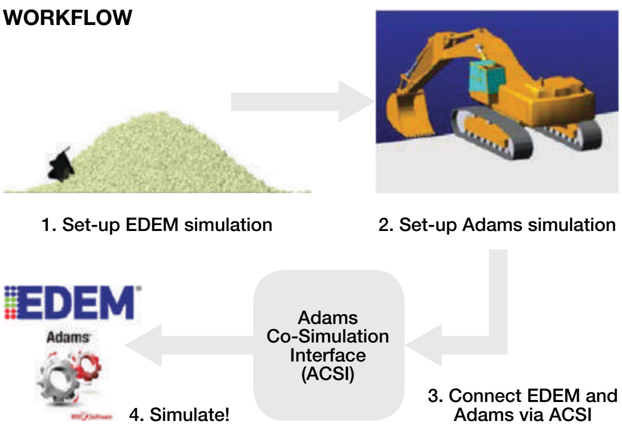


Figure 4 Adams EDEM Co-Simulation workflow

In addition, the computational efficiency of this method facilitated the support of stochastic analysis approaches, where uncertainties due to variations in model and terrain inputs were also accounted for, statistically. These stochastic simulations represented hundreds of potential soil characteristics, and allowed prediction of vehicle performance over a statistical range of soil and terrain properties and resultant development of confidence intervals for vehicle performance.

Higher fidelity approaches, where the soil properties emerged from simulated particle interactions were also employed. This was accomplished using a co-simulation between Adams and EDEM, a Discrete Element Method (DEM) based simulation offering from DEM solutions. In the DEM method, the material is represented by a collection of interacting particles with simple shapes (typically based on circles and spheres). The typical co-simulation workflow between Adams and EDEM is as shown in Figure 3. Potential EDEM contact is defined for designated vehicle parts. The displacement of these parts is determined by Adams and provided to EDEM. EDEM then determines the resultant reaction forces, which are passed back to Adams.

Using these approaches, tests such as a drawbar pulls, and sand-bed acceleration were simulated to gauge tractive behavior of the FED under various off-road scenarios. Though computationally intensive, these simulations were proved to add significant fidelity and result in more accurate correlation to test results.

Mobility Mapping

Leveraging the broader Hexagon portfolio (Figure 5), the Luciad Lightspeed technology from the Geospatial Business Unit was used to project the FED Alpha mobility characteristics predicted by Adams onto the test terrain at the Keewenaw Research Center (KRC). The integration of Adams predictions with geospatial mapping technology demonstrates the capability to visualize vehicle speed throughout a mapped domain based upon a combination of soil, grade and predicted vehicle performance data. Additionally, optimized routes can be computed based on selected route endpoints.

Additional operational data such as side-slope predictions and obstacle information can be incorporated into the above framework, thus creating a platform for comprehensive mobility assessment on an actual terrain using simulated vehicle performance data.

Real-Time Virtual Model Performance

To demonstrate the applicability of the full fidelity Adams models used for mobility assessment, to adjacent Hardware in the loop (HIL) and ADAS applications, a reduced order, real-time compliant variant of the full-fidelity model was created. The ability to derive vehicle dynamics modeling variants of varying fidelity, to support a specific simulation intent allows users to deploy a single modeling solution without costly, error-prone model translations between various tools. Furthermore, with the Adams Real-Time approach, the user has additional freedom to retain model features of interest. Typically, real-time vehicle performance may be achieved with a few simplifications of select component and connection representations, depending on the analysis and integration requirements. In this case, only the anti-roll bar model was simplified. The real-time model was tested in the VTD (Virtual Test Drive) analysis environment to demonstrate capability. In addition, the numerical accuracy and efficiency of this model was assessed relative to the baseline full-vehicle performance.

Adams has had a long standing presence in the area of on-road analysis. This effort demonstrates how these models can be extended using the broader Hexagon portfolio and reused for off-road analysis in the context of road terrain representation, real time analysis and operational mapping.

Learn more about Adams:
www.mscsoftware.com/adams

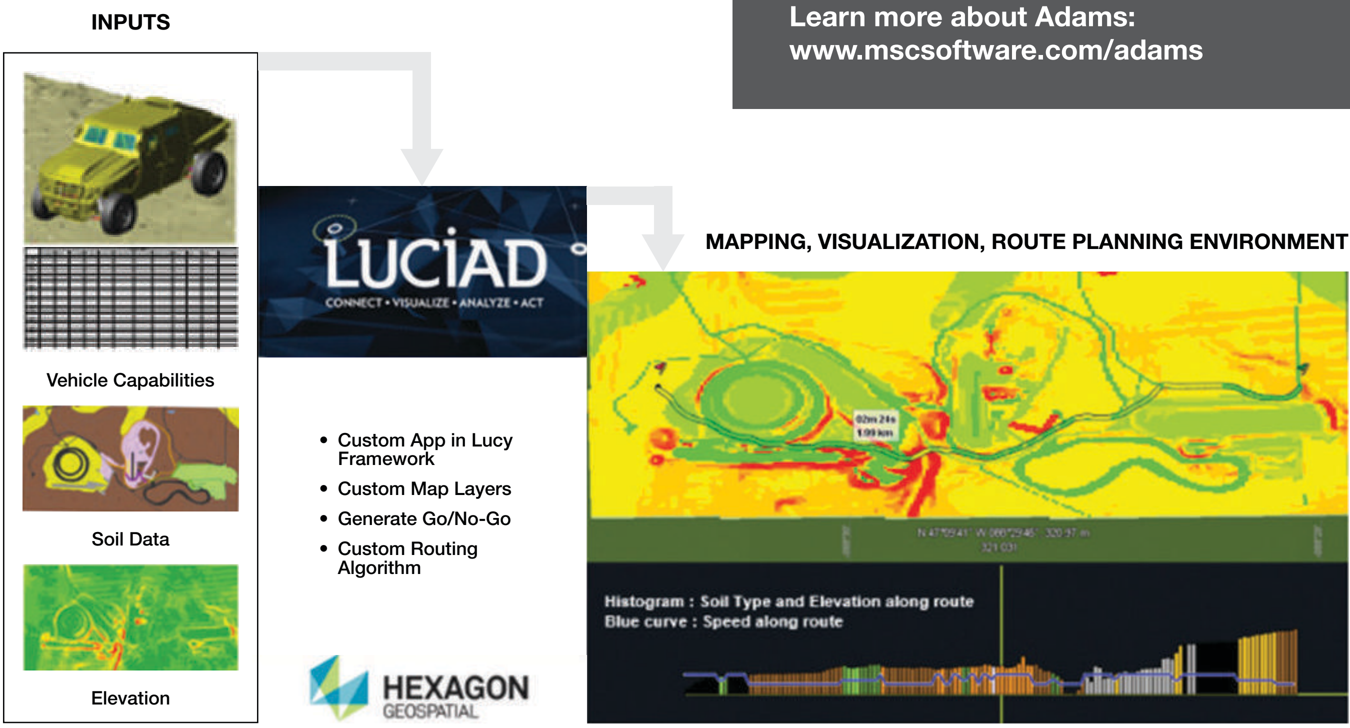


Figure 5 Mapping workflow, showing speed made good and route prediction

Autonomous Vehicle Testing

With **Christopher Kinser, General Motors, Milford, Michigan, USA**

Engineering Reality Magazine recently interviewed Chris Kinser from General Motors, the Director of their Global Autonomous Driving Center in Michigan and an industry expert in rapidly emerging sector of vehicles (AVs). He has a long history of working on electrification and autonomous driving systems, and his team in Milford is responsible for vehicle integration of several General Motors' advanced technology programs, including self-driving vehicles, as well as automated driving and active safety technologies. Chris's expertise in software, controls systems and vehicle performance integration have been recognized with three Boss Kettering Awards. Chris holds a Bachelor's in Electrical Engineering from Kettering University and a Master's of Engineering from Rensselaer Polytechnic Institute, USA.

What does your role in General Motors involve and what is GM's overall approach to the fast-emerging Self-Driving opportunity?

I am the Director of the Global Autonomous Driving Center at our Milford, Michigan proving ground where I manage a large engineering team. We believe that autonomous technology will play a key role in our vision of a world of zero crashes, zero emissions and zero congestion through the enormous potential benefits it holds for society in the form of increased safety and access to transportation.

General Motors is in a unique leadership position when it comes to developing and deploying self-driving vehicles in that we are the only company to have everything from design, engineering, validation, and testing all under one roof. My team is works closely with teams all around the country on developing autonomous driving solutions.

What do you see as the big challenges to Autonomous Mobility going mainstream in the next 10 years?

We are in the middle of a fundamental shift in how people and goods move through the world. Autonomous mobility will certainly play a huge part in that and at GM, we will be guided by the needs of our customers. It is also one of the most difficult challenges for automotive engineering. The biggest challenge I see to Autonomous Mobility going mainstream is getting all the systems necessary for self-driving vehicles to work together seamlessly. Next time you're behind the wheel, take a moment to reflect on all the tasks you are performing to drive the vehicle. Working on developing a system that can perform those same tasks is the engineering challenge of our lifetime. That's why at GM, we believe that a safe self-driving vehicle should be built from the ground up with seamless integration of the self-driving system.

Will all autonomous cars be electric vehicles?

At GM, we believe that all autonomous vehicles will be electric vehicles. Not only are electric vehicles better for the environment and quieter for city traffic, but they allow for simpler integration of the advanced technologies required for the cleanest and safest operation of autonomous vehicles. For example, an all-electric vehicle has a more stable power source and a faster responding propulsion system that provide it inherent advantages over its internal combustion counterparts.

Why did GM choose Hexagon/MSC technology for its Autonomous Driving strategy?

We see Hexagon as a company totally devoted to the autonomous sector in its business focus. Hexagon's combination of sensor and scanning technologies like Leica cameras, and its simulation software suite like MSC's VTD (Virtual Test Drive) software, fill many of the needs of the market. VTD is in the center of a comprehensive GM simulation environment that we have developed with Hardware-in-the-Loop. We use VTD in conjunction with software products like CarSim and Simulink (for control systems) in our real time virtual automated driving vehicle testing environment.

What is your vision for GM in the autonomous mobility space in say 5 years from now?

It is still the early days of autonomous mobility and we are excited by the opportunities for this technology to improve the world. In terms of engineering and development, we will continue to listen to our customers and deliver advanced mobility solutions that meet their needs.

Which country or countries do you think will go fully autonomous with cars first in your opinion?

I can't speak to the specifics of timing, but we have focused our shared autonomous development on San Francisco and the United States.

General Motors Advances **Virtual Autonomous Driving & Active Safety**

By **Chris Kinser, General Motors**

Figure 1: **Cruise Autonomous Cars**

General Motors operates a total vehicle performance center at the Milford Proving Ground in Michigan (Figure 2). The Global Autonomous Driving Center is a subset of this work focused on developing active safety features like advance park assist, lane keep assist, full-speed range adaptive cruise, and Super Cruise. This work is guided by GM's vision of a future with zero crashes, zero emissions, and zero congestion. The mission of our team is to provide smooth, capable driver assist systems that delight our customers.

GM's Approach to Automated Driving

The industry standard scale for levels of autonomy (SAE) is helpful from an academic perspective when discussing vehicles and their capabilities. However, when we begin development of a new vehicle or system, we don't start with a level in mind, but rather with the use case and a set of features that we believe we can safely implement. It is this focus on safety that guides us through the process.

Figure 2: **General Motors operates a total vehicle performance center at the Milford Proving Ground in Michigan**

General Motors is the only company that has everything from design, engineering validation, and testing all under one roof. This is more than just designing and building the vehicle. It also includes everything from in-house security and connectivity systems to software development and high-resolution mapping. Having everything under one roof puts us in a unique position to safely develop and deploy autonomous vehicle technology.

Super Cruise

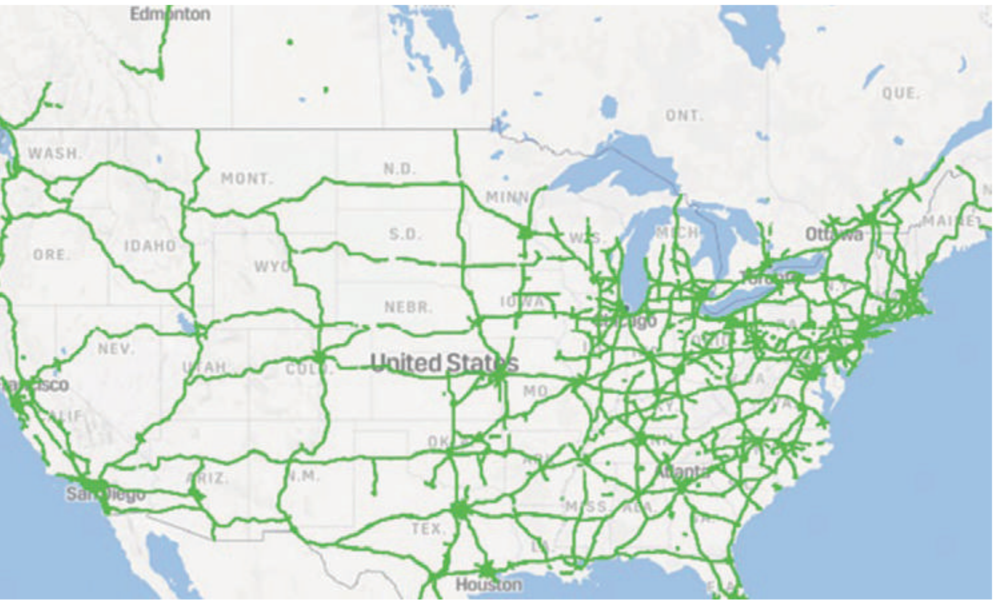
Super Cruise is an advanced driver assistance feature that enables hands-free driving on supported roads. It combines adaptive cruise control and lane-centering control with a driver attention system (Figure 3) to allow you to drive with your hands off the wheel and eyes on the road. Super Cruise is aimed at providing comfort and convenience in long-distance travel and daily commutes. Customers receive updated maps on a regular basis (Figure 4).

Figure 3: **General Motor's Super Cruise Cadillac Driver Attention System**

Safety is engineered into every step in Cruise's self-driving vehicles including design, development, manufacturing, testing and validation.

Cruise Autonomous Vehicle (AV) Program

In May 2016, GM completed the acquisition of Cruise Automation a Silicon Valley startup with considerable self-driving software development expertise. Combined with our expertise in engineering and developing vehicles, our teams began testing self-driving vehicles in San Francisco, CA, Scottsdale, AZ and Warren, MI. By September 2017, we revealed our first self-driving test vehicle built from the start to operate on their own with no driver [1]. Safety is engineered into every step in Cruise's self-driving vehicles including design, development, manufacturing, testing and validation. On a typical day, Cruise autonomous test vehicles safely execute 1,400 left turns and our teams analyze all that data and apply learnings. Based on Cruise's experience of testing self-driving vehicles, every minute of testing in San Francisco is about as valuable as an hour of testing in the suburbs because of the complex decisions being made.



Reference

'How we built the first real self-driving car (really)', Kyle Vogt, Cruise, September 11, 2017
Blog Post: https://medium.com/cruise/how-we-built-the-first-real-self-driving-car-really-bd17b0dbda55

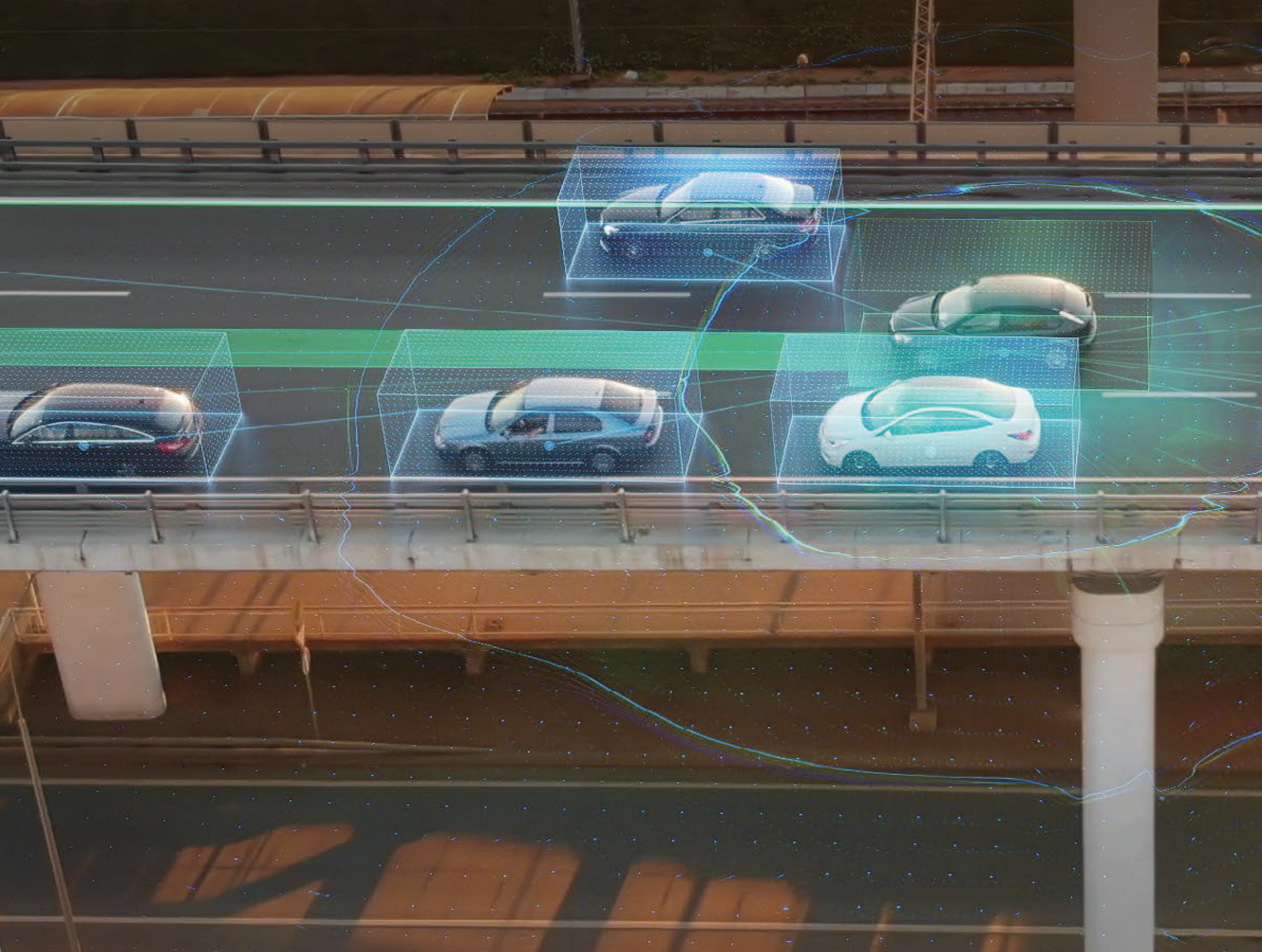
Figure 4: **GM Super Cruise, before going to production, required mapping every major road in the U.S. and Canada**

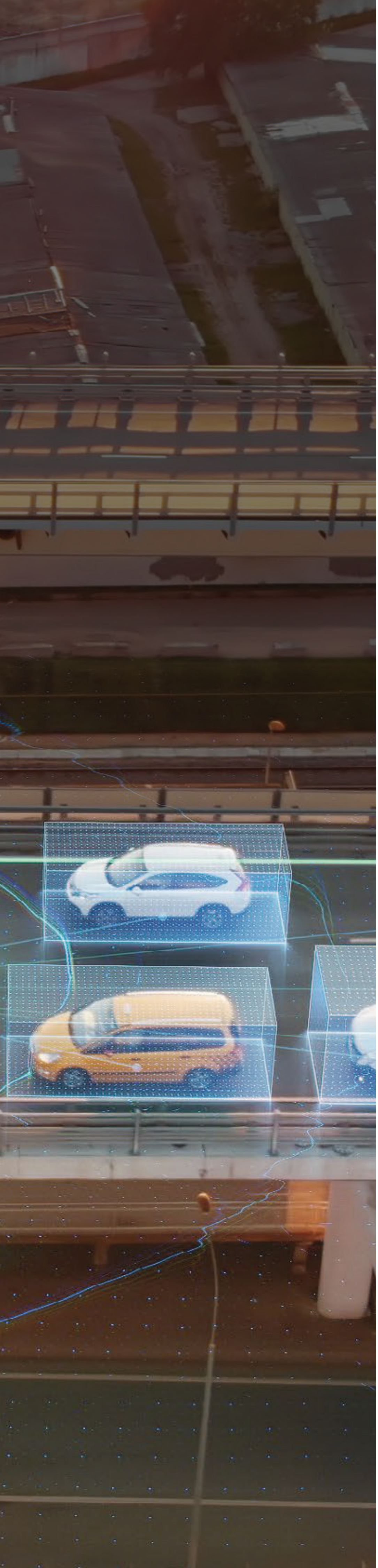

cruise
JOIN THE DRIVERLESS REVOLUTION

Multi-Resolution Traffic Simulation

for Connected Car Applications using VIRES VTD

By **AUDI AG:** Andreas Kern
Technical University of Munich: Manuel Schiller
Institute of Transportation Systems: Daniel Krajzewicz
VIRES, part of Hexagon: Marius Dupuis





Vehicular ad hoc networks (VANETs) have attracted a lot of research attention over the last few years because they have the potential to improve traffic safety, efficiency and driver comfort. In fact, several ADAS (Advanced Driver Assistance Systems) applications, such as cooperative driving and subsequently automated driving, can only be achieved through wireless communication between the vehicles on the road.

Since those systems often exhibit safety-critical features, rigorous testing and validation must be completed before their mass adoption. Although real road tests using physical prototype vehicles offer the highest degree of realism, the large amount of resources needed to perform large-scale and extensive testing of vehicular networks renders their use impossible. Simulations are essential to validate the performance of such solutions in large-scale virtual environments. Furthermore, simulation-based evaluation techniques are invaluable for testing those complex systems in a wide variety of dangerous and critical scenarios without putting humans at risk.

In the automotive industry, the use of simulation (Figure 1) is well established in the development process of traditional driver assistance and active safety systems, which primarily focus on the simulation of individual vehicles with a very high level of detail. When investigating and evaluating the performance of ADAS based on vehicular communication, this isolated view of a single vehicle alone or a small number of vehicles in the simulation is not sufficient anymore. Potentially, every vehicle equipped with wireless communication technology could be coupled in a feedback loop with the other road users participating in the vehicular network, and therefore, the number of influencers that need to be considered is drastically increased.

These considerations lead to a trade-off between accuracy in terms of the simulation details for each vehicle and scalability in terms of the number of vehicles that can be simulated with the available computing resources. This article presents an approach to solve this trade-off by coupling multiple resolutions of traffic simulation to get highly accurate simulation results where they are needed, and simultaneously achieve an efficient simulation of large-scale scenarios of the surrounding environment.

The Developed Multi-Resolution Traffic Simulation

Microscopic Traffic Simulator: SUMO

We chose to use Simulation of Urban MObility (SUMO) as the traffic simulator responsible for the simulation of the low-resolution area (LRA). SUMO is a microscopic, space-continuous, and time-discrete simulator. While it is employed in a wide range of research domains, its most notable use is shown in a high number of research papers regarding VANET simulations. SUMO is well known for its high execution speed, as well as for its extensibility. SUMO is ideally suited to simulate a high number of vehicles residing in the LRA due to its efficiency, which is partly achieved through its simplified driver model (which determines the path a vehicle will take).

Nanoscopic Traffic and Vehicle Simulator: VIRES Virtual Test Drive

We employ the nanoscopic traffic and vehicle simulator VIRES Virtual Test Drive (VTD) for the simulation of the high-resolution vehicles. VTD was developed

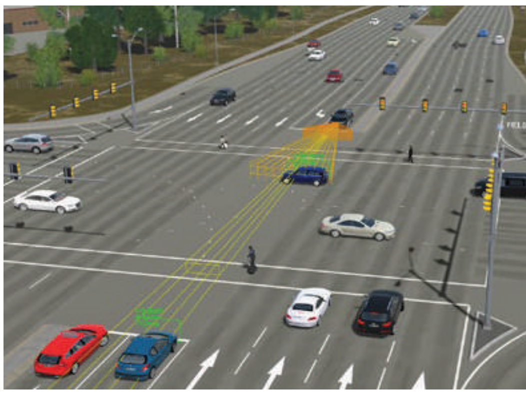


Figure 1: **ADAS simulation**

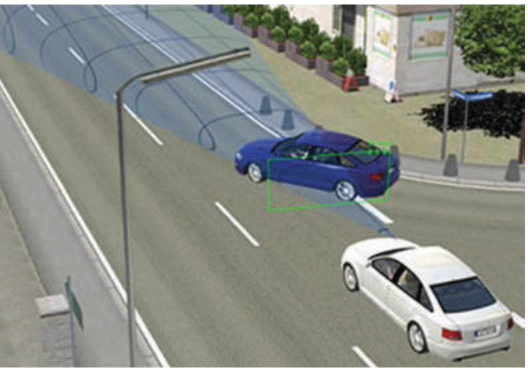


Figure 2: **3D visualization of a simulated RADAR sensor in VIRES VTD**

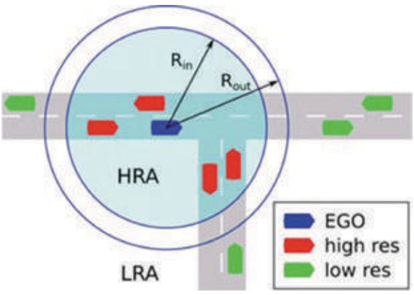


Figure 3: **Dynamic partitioning of the simulation area**

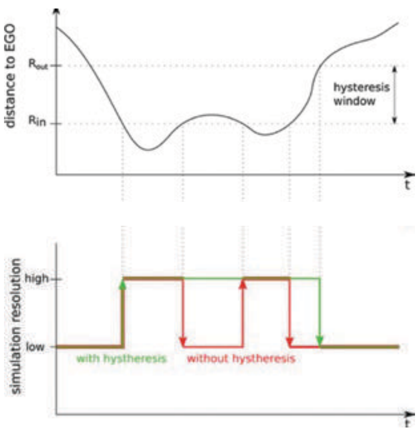


Figure 4: **Comparison of simulation resolution switching**

for the automotive industry as a virtual test environment used for the development of ADAS and Autonomous Vehicles. Its focus lies on the interactive high-realism simulation of driver behaviour, vehicle dynamics, and sensors. VTD is highly modular, so any standard component may be exchanged by a custom and potentially more detailed implementation. Its standard driver model is based on the intelligent driver model; however, an external driver model may be applied if necessary. The same concept applies to the vehicle dynamics simulation, where the standard single-track model can be substituted by an arbitrarily complex vehicle dynamics model adapted for specific vehicles. Each simulated vehicle can be equipped with arbitrary simulated sensors, for example a RADAR sensor, which is shown in Figure 2.

Offline Pre-processing To Enable Coupling

The two simulators rely on different data formats representing the modelled road network. In order to be able to run a co-simulation of both simulators, the underlying data basis must match. VTD uses the OpenDRIVE format to specify the road network. This specification models the road geometry as realistically as possible by using analytical definitions. SUMO on the other hand approximates the road network geometry by line segments. There are also differences in the modelling of intersections and lane geometries. To achieve a matching database, we convert the road network in an offline pre-processing step from OpenDRIVE to the file format SUMO supports.

Online Coupling and Synchronization

The coupling of the simulators at simulation runtime is based on the master-slave principle. Figure 4 shows this sequence of operations during a single simulation step, in which VTD and SUMO can operate with different temporal resolutions without losing synchronization.

S_{VTD} is the length of a time step for the high-resolution area (HRA), whereas S_{SUMO} is the length of a time step for the low-resolution area (LRA). Typically, the nanoscopic simulation is run at a higher frequency than the microscopic one. T_{VTD} and T_{SUMO} respectively denote the local simulation time in each simulator. At the beginning of each simulation step, a new timestep is simulated in VTD. If the next timestep has been reached for SUMO and the condition $T_{VTD} \geq T_{VTD} + S_{SUMO}$ is therefore fulfilled, the state of the high-resolution vehicles is sent to SUMO through a gateway. This triggers the simulation of the next timestep in the low-resolution model, and as a result, the positions of the low-resolution vehicles are passed back. These vehicles are now classified, and, if applicable, the change of resolution is performed for individual vehicles. When an exchange of a vehicle between the simulators happens, the previously mentioned inherent difference in the underlying road network may cause problems if a vehicle cannot be mapped based on its position in a specific lane due to differences in accuracy. This is especially true for complex intersections which are modelled quite differently.

After all the resolution changes have been successfully completed, the simulation is unblocked again and the next timestep can be simulated. This synchronization is very important to ensure reproducible simulation results across multiple simulation runs.

Dynamic Spatial Partitioning of The Simulated Area

Our approach aims to couple traffic simulation models of different resolutions at dynamic regions of interest. Contrary to conventional traffic simulation, we are not interested in investigating a large number of vehicles from a bird's eye perspective, but the focus is rather on a single vehicle (or a limited number of vehicles) which are used to conduct a test drive in the virtual environment. This vehicle of interest has the ADAS system under investigation onboard, and is referred to as the EGO car. The simulated measurements and sensor values are fed into the ADAS, and depending on its type and its use case, the respective ADAS directly or indirectly influences the vehicle's state and behaviour.

Based on this distance criterion, an area of interest is defined that centres around the EGO car, and in which the defined simulative high-fidelity requirements must be fulfilled. Since the EGO car is driving continuously through the virtual environment, this area of interest is likewise being moved along. We therefore partition the global area of the simulation dynamically into a high-resolution area (HRA) and a low-resolution area (LRA). Figure 3 shows a schematic view of the dynamic spatial partitioning. There, the HRA is defined as the area of a circle which is centred around the EGO vehicle. Red vehicles are within that circle and are therefore simulated in high resolution by the involved nanoscopic simulator, whereas the green vehicles are outside of the circle and are consequently simulated in low resolution by a microscopic simulation. All vehicles exist in the microscopic simulation, but in the nanoscopic simulation contains only the high-resolution vehicles, and their movements are applied to their proxy counterparts in the microscopic simulator.

Due to the dynamic nature of road traffic, the EGO car, the high-resolution vehicles as well as the low-resolution vehicles are permitted to move continuously. The classification of the assigned resolution mode is therefore performed after each time step of the simulation. Vehicles for which the classification has led to a change in resolution are transferred to the appropriate simulator. This change of resolution is possible in both directions at every time step. However, since the HRA is defined to be centred around the EGO car, it is always simulated in high resolution.

In order to prevent vehicles which are close to the boundary between HRA and LRA from oscillating very frequently between the two resolution areas, a hysteresis controller as depicted in Figure 5 is applied in the classification process. As shown in Figure 3, the two thresholds R_{in} and R_{out} are defined. A vehicle is transferred into the high-resolution simulation only if its distance to the EGO car falls below the value of R_{in}. The exchange back to the low-resolution simulation is carried not out until the threshold R_{out} is exceeded.

VTD is highly modular, so any standard component may be exchanged by a custom and potentially more detailed implementation.

Simulation & Evaluation

Scenario and Simulation Setup

A synthetic scenario was created for testing the coupling concept and evaluating its performance. It consists of a single straight road running west to east with a length of 50 km and two lanes, one for each direction. Each lane is configured to have a constant inlet of 1,000 vehicles per hour heading either east or west. The EGO car is located near the start of the road. It drives from west to east and is followed by a traffic flow, heading towards the oncoming traffic flow. This artificial road was first modelled in the OpenDRIVE format and was then converted to the SUMO road network format.

We performed two series of experiments. In the first series, the nanoscopic traffic simulator VTD was applied to the whole simulated area. In the second series, we used the described multi-resolution concept to partition the simulation area between VTD and SUMO. We chose a timestep of S_{VTD} 20 ms for the high-resolution area in VTD and a timestep of S_{SUMO} 1 s for the low-resolution area in SUMO. The hysteresis thresholds which define the dynamic area of interest were set to R_{in} 500 m and R_{out} 550 m.

Performance Evaluation

We measured the duration it takes to perform each simulation step over the simulation period of 1,800 s, while the number of vehicles was constantly being increased. Each series consists of five separate simulation runs to account for fluctuations in the measured execution times. To illustrate the trends of the measurements more clearly, the moving average is also displayed in the following figures.

Figure 6 shows the performance development of the nanoscopic simulation while increasing the simulated vehicle count over the simulation period. The duration of each simulation step is almost constant up to a count of 70 vehicles. Until then,

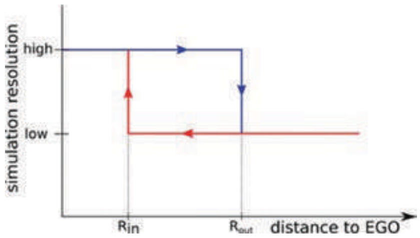


Figure 5: **Hysteresis control of the simulation resolution**

the duration is around 12 ms, which is less than the timestep length of 20 ms and therefore yet fulfils the real-time constraint. At around 150 vehicles, the duration is beyond these 20 ms and real-time simulation is not possible anymore. With increasing vehicle count, the duration for each timestep also considerably increases and reaches 180 ms at the end of the simulation period. This results in a factor 15 computation time increase compared to the amount of at the beginning of the simulation. The overall simulation took over 120 min to complete, which is four times more than the simulated time.

Figure 7 shows the performance development of the multi-resolution simulation in the same simulation scenario over the same simulation period. While the total vehicle count is increased the same way as in the pure nanoscopic simulation, the separately plotted nanoscopic vehicle count illustrates the number of cars

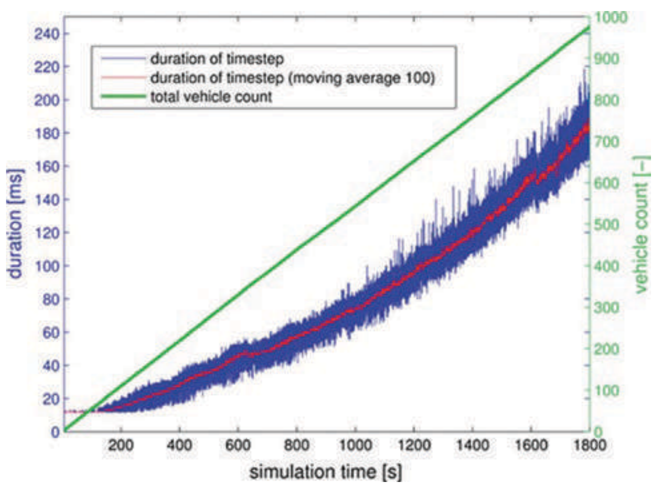


Figure 6: **Simulation performance—nanoscopic simulation only**

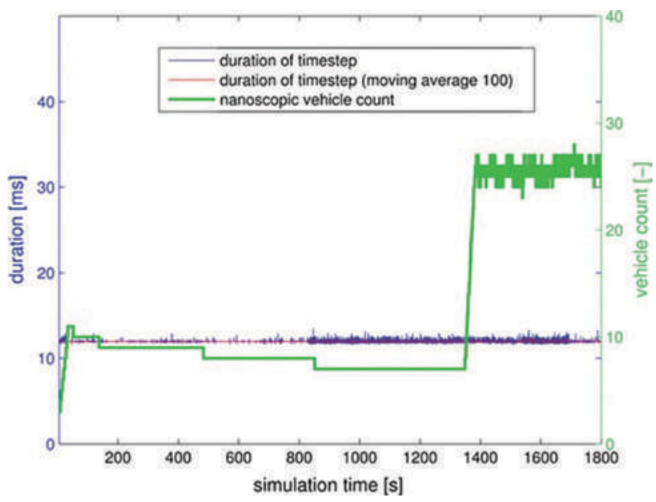


Figure 7: **Simulation performance: multi-resolution simulation**

which are within the high-resolution area. It shows that reducing the nanoscopic model's area of interest fulfils the aim of reducing the overall simulation time. After a local maximum of 11 nanoscopic cars is reached, this count decreases slowly since slower vehicles are left behind the faster moving EGO car. At around simulation time 1,350 s, the two traffic flows from each end of the road meet in the middle of the road, which then increases the nanoscopic vehicle count. However, because the extent of the HRA is limited, the nanoscopic vehicle count does not exceed a certain limit, which for the given configuration is at around 27 vehicles. The duration for the timesteps stays on average constant around 12 ms, so the overhead resulting from the coupling of the two simulators is negligible. The execution time of the microscopic simulator is also shown to be negligible due to its less detailed, yet much more efficient, simulation model. The overall simulation took less than 18 min to complete, so the simulation was faster than real time by factor 1.66 and the real-time constraint was fulfilled throughout the whole simulation period.

Conclusion

In this article, we proposed a concept for coupling traffic simulators of different simulation resolutions to achieve a multi-resolution traffic simulation which focuses on a dynamically-determined area of interest. The presented methodology partitions the simulation area into a variable, highly detailed region of interest represented by a nanoscopic model, with VIRES Virtual Test Drive (VTD), and the surrounding area simulated at low resolution by a microscopic model. The evaluation shows a dramatic reduction of computation time in comparison with a pure nanoscopic simulation of the same simulation dimensions, which even makes real-time simulation possible. This divide-and-conquer strategy enables accurate, realistic, and large-scale testing and validation of real implementations of driver assistance systems based on vehicular networks in a virtual environment. As the next step, we are investigating the application of the multi-resolution simulation methodology for the other domains relevant for the simulation of vehicular networks, namely network simulation and application emulation, to model the whole system across all domains efficiently at high fidelity.

Reference

"Multi-resolution Traffic Simulation for Large-Scale High-Fidelity Evaluation of VANET Applications", Manuel Schiller, Marius Dupuis, Daniel Krajzewicz, Andreas Kern and Alois Knoll, © 3rd SUMO Conference 2015 Berlin, Germany

This divide-and-conquer strategy enables accurate, realistic, and large-scale testing and validation of real implementations of driver assistance systems based on vehicular networks in a virtual environment.

Achieving Autonomous Driving **with Simulation & Testing**

By **Dr. Luca Castignani, Chief Autonomous Driving Strategist, MSC Software**

Self-driving is becoming more and more realistic. Every day, thousands of autonomous vehicles (Figure 1) are being tested on the roads by companies like Waymo, Cruise, Uber, Tesla, and some of those companies have accumulated millions of miles of road testing data, enhancing and validating their autonomous "brain", with the hope that in the near future, full automation can be achieved.

When the Pumpkins Take a Stroll

Today, everyone understands the importance of road testing for self-driving vehicles, and the industry is spending a fortune on it. On an average, a fully equipped autonomous vehicle can cost more than half a million dollars, so a small fleet of 20 vehicles would mean a 10-12 million dollars investment in the hardware itself, to perform the road testing for autonomous driving. However, is road testing really enough to help us reach level 5 autonomy in the foreseeable future?

To answer that question, first we need to understand: how many miles of testing is required to develop an autonomous driving system? The commonly accepted number among the industry is "one billion miles". So how many miles of road testing have we done so far? Waymo, the world's leading autonomous driving company in road testing, has accumulated an impressive 9 million miles in the past 9 years. However, even if we increase that effort by 10 fold, it would still take about 100 years for us to complete the validation of one self-driving system, if we solely rely on road testing.

As long as you only have to check a few use cases (in the range of tens), you can easily test them on real roads. However, in order to assure safety for Autonomous Vehicles, the number of conditions to

Figure 1. Autonomous Vehicle Testing Platform Developed by AutonomouStuff, Part of Hexagon

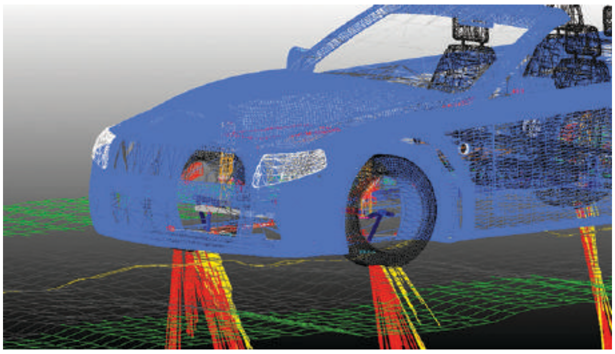


Figure 2. High-fidelity Vehicle Dynamics Model in MSC Adams, Part of Hexagon

be evaluated scales quickly to millions and there is no way to tackle it without simulations. For example: What if you want to know how the car will behave when the city decides to paint all the road signs in yellow instead of white? Or what happens when the trees planted today grow to a size that prevents the driver from seeing the pedestrians?

With simulation, it's also possible to create outlier scenarios for testing. Think of workers carrying a large mirror while crossing the street. Think of children dressed up as pumpkins, out for a walk on Halloween. Not many of these scenarios have been taken into consideration, but those are the realities.

How is Autonomous Driving Simulation Different than the Traditional Vehicle Simulation?

Computer-aided engineering (CAE) simulation has been a trusted tool leveraged by the automotive industry for dozens of years now. From vehicle handling & steering, ride & comfort (Figure 2), NVH, durability, aerodynamics, controls validation, all the way to manufacturing process simulation and advanced materials (composites) analyses, CAE companies like MSC Software (acquired by Hexagon AB in 2017) have been providing industry leading simulation solutions to help auto OEMs and suppliers refine the attributes of every newly developed vehicle model.

Engineers have been using CAE to improve the vehicle performance for a long time, so how is autonomous driving simulation (Figure 3) different than the traditional CAE simulation?

First of all, in an autonomous simulation environment, we need to capture more than the vehicle under design (the so called "Ego Vehicle"). Different types of participants need to be included in the scenario, for example, other vehicles, pedestrians, cyclists, animals (moose, deer, kangaroos) and so on.

Secondly, a realistic perception is crucial to accurate simulation. Unlike the vehicle models in a traditional CAE environment, the "Ego Vehicle" in an autonomous testing model doesn't always have a perfect understanding of its surroundings. Instead, it only knows what its sensors perceive, therefore it is important to accurately simulate those different types of sensors (cameras, RADAR, LiDARs…) and also how they function especially in adverse atmospheric conditions (sun glare, fog, snow, rain, evening light…).

These are just some examples that highlight how autonomous driving simulation is very different than a traditional CAE car simulation, and for those same reasons, not every traditional CAE solution provider is going to be a natural fit as an autonomous driving simulation partner.

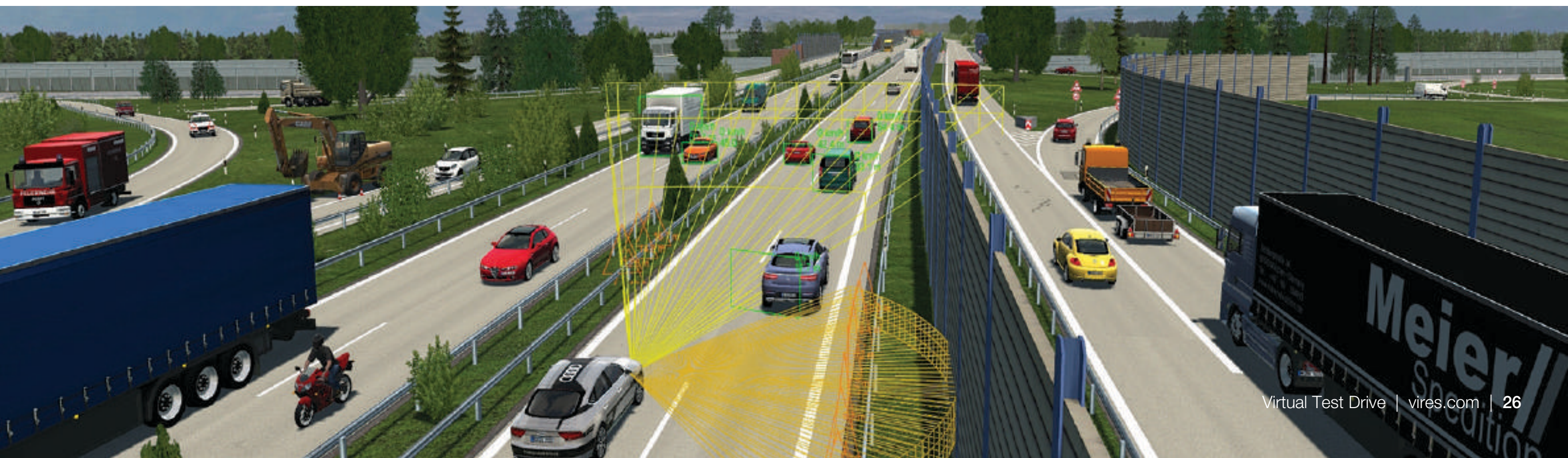


Figure 3. Autonomous Driving Sensor Simulation Environment by Vires VTD (Virtual Test Drive), Part of Hexagon

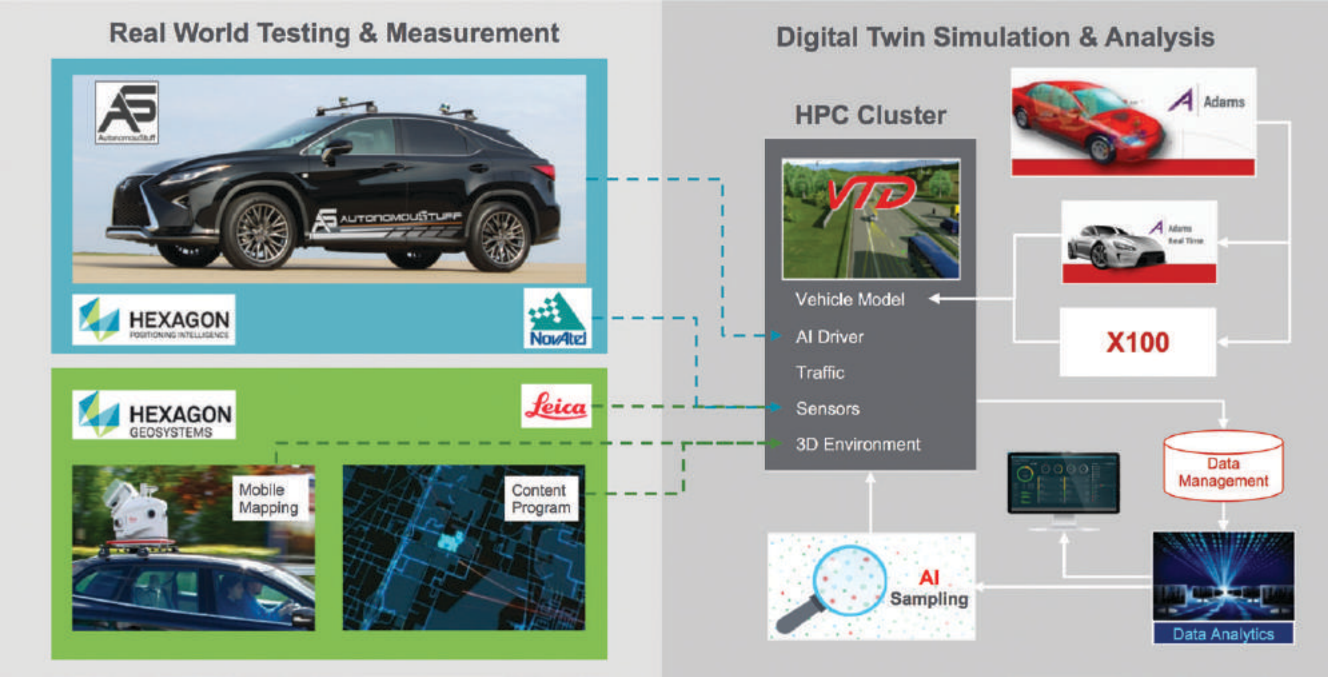


Figure 4: Hexagon's Complete Autonomous Driving Simulation & Testing Portfolio

A Comprehensive Strategy for Autonomous Driving Simulation and Testing

Through a series of acquisitions with MSC Software and VIRES VTD in 2017, AutonomouStuff in 2018, along with indigenous domain expertise in sensors, smart city and positioning intelligence solutions, Hexagon holds the leading edge in autonomous driving validation (Figure 4). This includes solutions for these domains: Vehicle CAE Modeling, Sensor Measurements and Modeling, 3D Environment modeling, Scenario Testing, Data Management, AI Driver and above all an open platform on which to integrate these.

A. Virtual Test Drive (VTD)

VTD is an open platform for creation, configuration, and animation of virtual environments for the testing and validation of Autonomous Vehicles. It acts as the coordinator for the domain segments mentioned above. It receives vehicle position and motions, rebuilds the 3D environment in real time (including traffic and pedestrians), computes the sensor perception, calculates the movements of all surrounding vehicles and so on. This stream of data can then be used to train the AI Driver at all levels (sensor fusion, object detection, path planning) or to assess its performance in terms of safety, comfort and efficiency.

B. Vehicle CAE Model

Depending on the scenario that the simulation needs to address, having vehicle models with different level of complexity can be handy. For example, for a common scenario such as emergency braking on a highway, a simplified model is preferred so a higher number of scenario permutations can be verified in a given amount of time. For a more dynamic scenario that perhaps involves a swift lane change to avoid a crash, a higher fidelity Adams Car model with a well-correlated suspension system is going to be essential. Not to mention that within an autonomous vehicle the riding comfort will become even more critical to the passengers such as to not suffer motion sickness while reading your favorite book or working with the laptop.

C. Sensors and Sensor Models

VTD has a complete set of sensors to replicate the physical sensors used in an autonomous vehicle: cameras (included infrared), LiDAR, RADAR and ultrasonic sensor. Each sensor can be represented

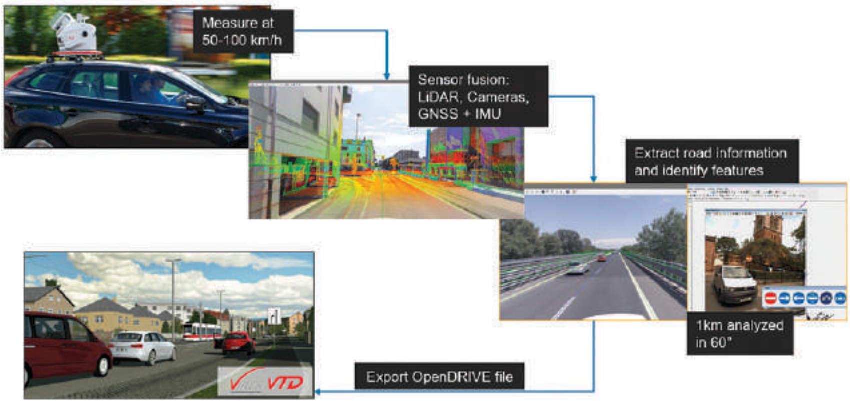


Figure 5. Creating Virtual 3D Environment in VTD from Metrology Road Measurements

with different levels of fidelity, from reproducing the intricacies of a laser beam reflecting over rough surfaces to simply capturing the basic sensor characteristics (in order to achieve the maximum simulation speed). To further enhance the fidelity and the variety of the sensor models, team VTD is also working with the world-leading sensor manufacturers like Leica and NovAtel (all part of Hexagon).

D. 3D Driving Environment

A 3D virtual environment can be generated either from inside VTD, or from scanning the actual roads. Creating the environment inside VTD gives you maximum control over all the details, while generating the 3D environment from measurements (LiDAR/camera) is more realistic and much faster. With Hexagon's new Leica Pegasus:2 mapping platform and its connection with VTD (Figure 5), engineers are expected to speed up the "Road Digitization" by a factor of 20 in the near future.

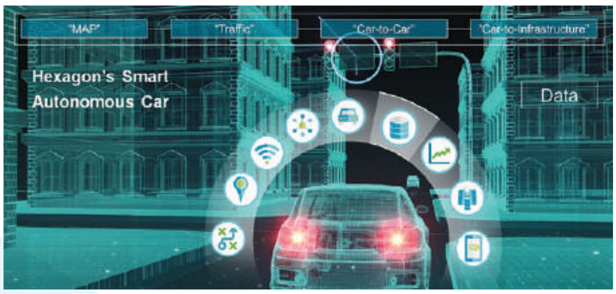
E. Scenarios and Data Management

With millions of scenarios to be evaluated at each step of the autonomous vehicle development, there is simply no way to manage everything manually. Indeed, Intel calculates that 1 Petabytes of data will be generated each day by an autonomous vehicle. That is where SimManager, the simulation management platform of MSC, comes into play to store the generated data and appropriately label them for easy access at any stage. With such a well-organized collection of data, to find out the "needle in the haystack" (such as "extract all simulations with rain") is like child's play and to compute meaningful performance indexes (such as "average time to collision") becomes a no-brainer.

F. Artificial Intelligence (AI) Driver

The AI Driver is the core of every autonomous system, and users can easily connect VTD to their own AI Driver to carefully validate them under all conditions, including sensor failure or misbehavior such as mud sputters covering a portion of a LiDAR. MSC Software is also working with its sister company, AutonomouStuff (both part of Hexagon), to connect AutonomouStuff's AI Driver to VTD so partners of AutonomouStuff can run their physical road tests and virtual tests with exactly the same AI brain.

In summary, today Hexagon owns many of the simulation and testing assets necessary for autonomous car projects: sensors and technology to manage smart intermittent sampling, HD maps from Hexagon Geosystems, and a turnkey platform for autonomous vehicle development from AutonomouStuff. Add in MSC Adams vehicle modeling, VTD to recreate the external environment and traffic, data management via MSC SimManager, and there is a very compelling turnkey autonomous vehicle solution toolset for both simulation and testing awaiting both OEMs and Start-ups around the world.



BMW Group: Generation and Validation of Sensor Models **for Automated Driving Systems Using VIRES VTD**

By **Alexander Schaermann, Data Engineer, BMW Group**
Timo Hanke, Data Engineer, BMW Group

INTRODUCTION

Due to advancements in sensor technology and data processing algorithms over recent years, great progress has been made to enable automated driving systems to improve safety and comfort for the vehicle driver and occupants. Yet, due to the complexity of self-driving, one of the main challenges remains in ensuring and validating the safe conduct of the automated driving systems for public use.

Virtual worlds provide a suitable, safe and controlled environment to handle an important part of the required testing and validation efforts. A proper choice of scenarios as well as the generation of virtual sensor data that closely matches reality are among the central requirements for the success of the virtual development approach. Virtual sensor data is generated by means of sensor models that form a central component of the virtual environmental perception (Figure 1). This perception data constitutes one of the main input streams for the decision making algorithms of an automated driving system. Hence, the fidelity of the sensor model is a deciding factor for the viability and validity of virtual development and testing.

Generally speaking, there are two types of sensor models:

Sensor error models aim to reproduce the statistical characteristics of errors, i.e. deviations between the perceived and true values, of the measurement and perception performed by vehicle sensors.

Sensor measurement models, on the other hand, are based on a physical description of the measurement process, and they generate low-level measurement data based on the virtual scene. Models of this type are commonly used for a variety of sensors in robotics research, while the measurement models for automotive sensors are only emerging.

In this article, we introduce a sensor measurement model for an automotive LiDAR sensor. The model is based on a ray tracing approach for the simulation of the measurement process. This enables the real-time generation of a LiDAR Point Cloud within the framework of an automotive driving simulator. By directly comparing data from the real-world test drive to virtual data generated by the sensor model in a virtual environment, we are able to quantify the accuracy and validity of the sensor model using appropriate metrics.

SENSOR MEASUREMENT MODEL

A. Real-time Ray Tracing in a Driving Simulator

We consider the scanning type of LiDAR sensor, which is typically used in the automotive industry. This type of sensor determines distance by measuring the travel time of a laser pulse reflected by a target surface. Its angular resolution is achieved by means of scanning, i.e. by moving the transmitted laser beam as well as the selective field of view of the optical detector array successively over the sensor's complete field of view. Most commercially available systems at this time employ a mechanically rotating mirror for the scanning task. The operating principle of this type of sensor lends itself to a modeling approach using ray tracing techniques. The virtual environment for the proposed sensor model is provided by the Vires VTD driving simulation software (Figure 2), which offers a ray tracing framework based on the Nvidia OptiX ray tracing engine.

Figure 1. Virtual Sensor Models in VIRES VTD Environment

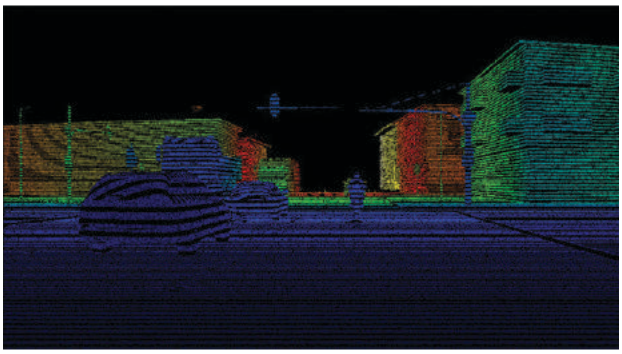


Figure 2. LiDAR Model Simulation in VIRES VTD

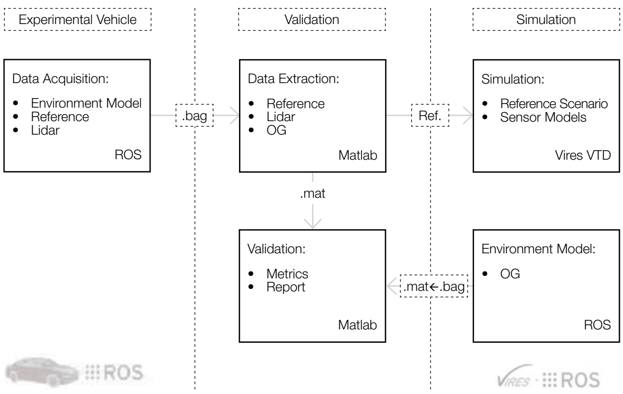


Figure 3. Tool chain for sensor model validation

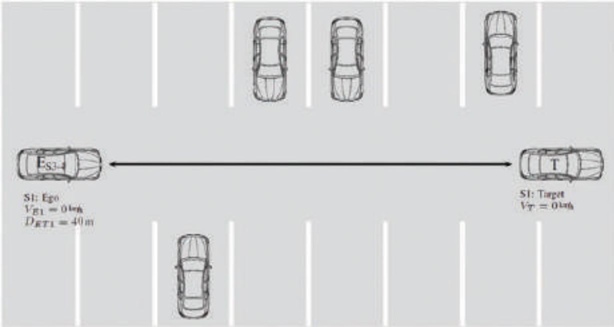


Figure 4. Static Validation scenarios with 40m distance between ego and target

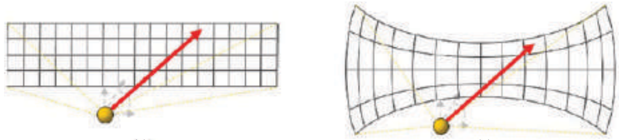
B. Virtual Point Cloud Generation

To model the beam transmission, reflection, and detection of the LiDAR sensor, the camera program of the sensor measurement model generates a ray for each set of azimuth and elevation angles. That results in a Point Cloud, if a valid distance measurement is obtained (see reference 1 for more details).

SENSOR MODEL VALIDATION

A. Methodology for Validation

For the validation of the LiDAR sensor model described in section II, we propose the procedure shown in Figure 3. This method is based on the comparison of real and synthetic data. In the first step, real data is captured with an experimental vehicle equipped with LiDAR sensors, a differential global positioning system (DGPS) and environment model algorithms running in ROS (middleware) and including an occupancy grid implementation. Then, synthetic data is generated using the LiDAR sensor model described in section II and exactly the same occupancy grid implementation as used in the experimental vehicle, but provided with simulated data from the sensor model in VIRES VTD. For data exchange between the model and ROS, the Open Simulation Interface (OSI) is used. As soon as real and synthetic data are captured, we evaluate the data in the validation framework using Matlab in a two-step procedure. In the first step, the direct comparison of real sensor data and model output is taking place. In the second step, we compare occupancy grids generated with real and synthetic LiDAR data representing the static environment of the test vehicle.

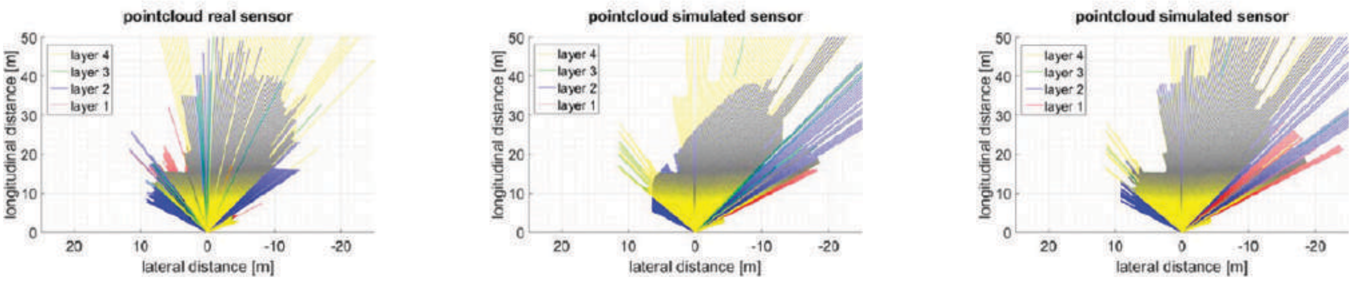


(a) (b)

Figure 5. Sampling grids for ray tracer: (a) Cartesian (b) Spheric

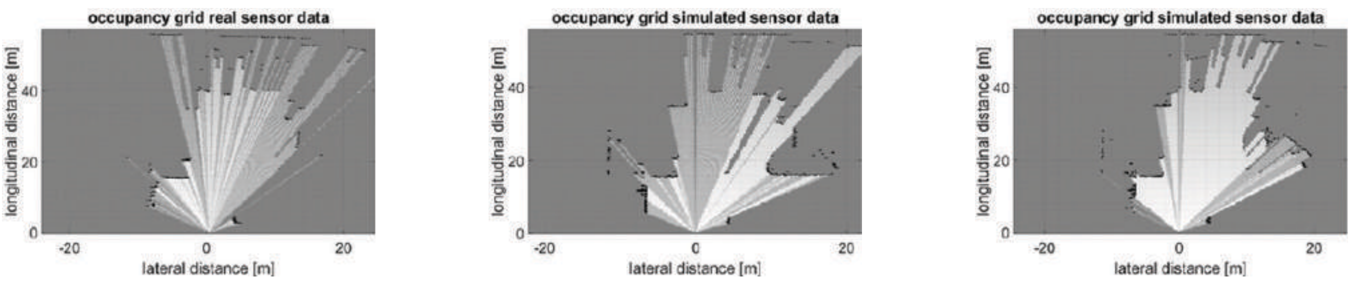
B. Validation Premises

For the validation of the sensor model, a static scenario is evaluated (see Figure 4). The two vehicles, Ego (E) and Target



(a) (b) (c)

Figure 6. Visualization of Point Cloud: (a) real Point Cloud, (b) synthetic Point Cloud from SC1, (c) synthetic Point Cloud from SC2



(a) (b) (c)

Figure 7. Visualization of occupancy grids: (a) real occupancy grid, (b) synthetic occupancy grid from SC1 (c) synthetic occupancy grid from SC2.

(T), shown in the schematic have an approximate distance of D1 = 40 m to each other. This area was modeled as a virtual 3D model for a simulator with particularly high fidelity requirements with respect to geometric dimensions and positions. Using this scenario, we show how the modeling of the LiDAR sensor's geometric configuration can be tested and how different configurations influence the generated Point Cloud and occupancy grids.

Two different sensor configurations SC1 and SC2 of the model approach are applied in this study. SC1 uses a Cartesian sampling grid for ray generation visualized in Figure 5(a), and SC2 uses a spherical grid shown in Figure 5(b). Usually for image generation, a linear distribution of rays generated with ray tracing is needed, so the SC1 approach would be the right choice.

Model State	Validation Level	Overall Error	Barons correlation	Pearson correlation
SC1	EDM PC	8729.2	0.733	0.824
	SG	1.0816×10^6	0.637	0.679
	OG	2.3668×10^6	0.602	0.677
SC2	EDM PC	8566.4	0.743	0.832
	SG	9.385×10^5	0.721	0.764
	OG	2.117×10^6	0.634	0.703

TABLE I: Calculated results for different validation metrics: overall error, Barons and Pearson correlations at different validation levels: Point Cloud, scan grid and occupancy grid.

However, since the geometry of the beam deflection of a LiDAR sensor leads to a conic shape of the point cloud, the spherical sampling grid is a more suitable choice for this purpose.

C. Data Evaluation

a. We start the investigation with a qualitative inspection of the captured Point Cloud shown in Figure 6. Visually observing the Point Cloud, it is obvious that the real Point Cloud is more similar to the Point Cloud generated from Sensor Configuration 2 (SC2) compared to Sensor Configuration 1 (SC1).
b. As mentioned before, occupancy grids are used as an abstraction level for sensor model validation. Here, we additionally use scan grids (SG) as a further abstraction level. Scan grids are single shot recordings of occupancy grids generated from a Point Cloud, whereas the occupancy grids are over time accumulated scan grids.

For evaluation of the environment model output, the real world scenario is re-simulated, and scan grids as well as occupancy grids are computed using generic Point Clouds from the two sensor configurations. The scan grid results are shown in Figure 7. Visually comparing the scan grid representations of the two sensor configurations with the real data, we can see a higher alignment between the real scan grids and the scan grids from the SC2. To quantify this observation, three metrics are applied and summarized in Table I. Similar to the quantitative results from the Point Cloud evaluation, these values show lower overall error and higher correlations between real scan grids and scan grids from SC2 compared to SC1.

SUMMARY AND FUTURE WORK

In this article, we propose a physically motivated sensor measurement model based on a ray tracing approach for an automotive LiDAR sensor. The model was employed to faithfully recreate the full sensor processing chain in a virtual environment with the help of VIRES VTD. Furthermore, a full processing chain in the virtual environment starting from low-level sensor data and ending with the first fusion stage of the whole automated driving system was reproduced in a virtual environment. With the presented setup, it is possible to evaluate real driving situations and reconstruct them in the simulation from high-fidelity data for static and dynamic scenarios. As sample use cases, we showed a static situation on a parking lot. We could quantify how closely the internal environment representation, i.e. the input to the automated driving function, matches between real world scenario and the simulation using a raw data LiDAR sensor model and appropriate validation metrics. The results represented in this paper show a higher correlation between real and synthetic data using the sensor model with a spherical ray tracer sampling grid.

Reference

1. "Generation and Validation of Virtual Point Cloud Data for Automated Driving Systems", T. Hanke, A. Schaermann, M. Geiger, K. Weiler, N. Hirsenkorn, A. Rauch, S-A. Schneider, and E. Biebl, IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), October 2017, Yokohama, Japan

Shaping Smarter Simulation **with Artificial Intelligence**

By **Dr. Horen Kuecuekyan, Director of Product Development & Artificial Intelligence, MSC Software**

Simulation provides key insights into system behavior and performance, especially with design optimization and validation. However, there are many instances where simulation or design exploration is not applicable because of limited computational resources.

Artificial Intelligence (AI) is a promising approach to help reduce the less important simulation scenarios by studying the existing simulation data. Many different machine learning algorithms are applied to train an AI-model, such as decision trees, random forests, fuzzy logic, Markov decision-based artificial neural networks DQN (Deep Queue Networks), and various other refinements beyond DQNs. In many instances, an AI-model is not required to have the same fidelity as an actual simulation model, since most engineers simply expect the trained AI-model to be better and more consistent than the engineering judgement or simplified (reduced order) models.

Figure 1. A typical scenario to test an autonomous driving system using VTD

When there is not a sufficient amount of physical data available, simulation generates the simulation data to train a reliable AI-model.

Our Artificial Intelligence team at MSC Software focuses on applying AI technologies both on the existing simulation data and also on strategies to generate simulation data in the AI process.

Here are Some of the Key Application Areas:

Development of AI Sampling: AI Sampling will generatively create the individual simulations to be used for multiple purposes, including training the AI-models. One application of the AI Sampling is our initiative to create a Smart Testing Environment for

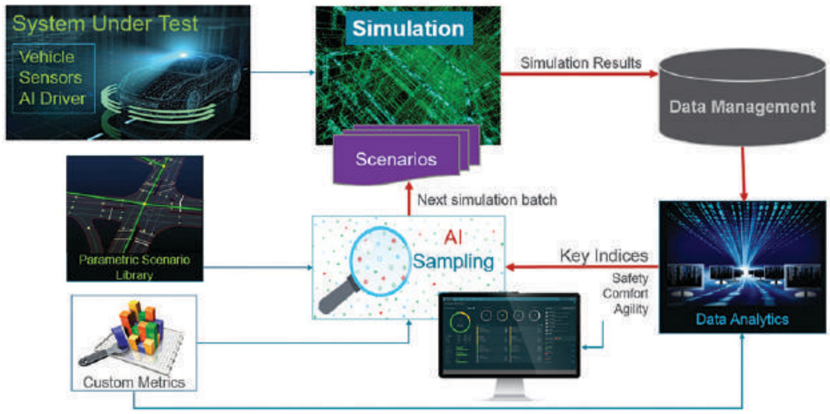


Figure 2. End-to-end cycle for the variation, simulation, result evaluation and training of the AI for the autonomous system

Autonomous Systems (STEAS). The AI Sampling will generate the test plan to perform the relevant and important scenario simulations (Figure 1) to validate either ADAS (advances driver assistant system) or fully autonomous systems.

One of the main questions people ask around autonomous systems is, "how can we generate and test the millions of scenarios that are needed to virtually validate an AI Driver?"

With AI-sampling, we are developing a solution to handle this huge "event space" (Figure 2). The basis for AI-sampling is a parametric and modular scenario library, which allows the creation of a broad set of individual test cases. The simulation results are analyzed and classified on their relevance and diversity, and then used in the AI Sampler as inputs to create a more refined test case set with each iteration. This test case set represents all the different behavior patterns to be tested, verified, and applied to analyze the simulation databases on their relevance. By applying this iterative process, AI Sampling can learn how to optimize the test set and create a feasible and relevant set of test cases, which covers all the different situations that can occur.

Creating the Predictive Models: We refer to the predictive models as "AI Twins" (Figure 3). AI Twins can predict the outcome of simulation studies and can be used in the product development lifecycle when performing the traditional simulation is too costly or takes too much time.

Our AI team at MSC Software is working with the goal to train AI to learn from simulations, to extend the knowledge over time, and dramatically increase the performance and efficiency in the modeling process.

Applying AI and machine learning tools in the technological applications can enhance simulation efficiency, improve product quality and reduce production costs. AI Sampling will automate the simulation generation, sample the constantly growing design space, and help autonomous driving developers to capture the millions of scenarios that are needed to achieve level 5 autonomy.

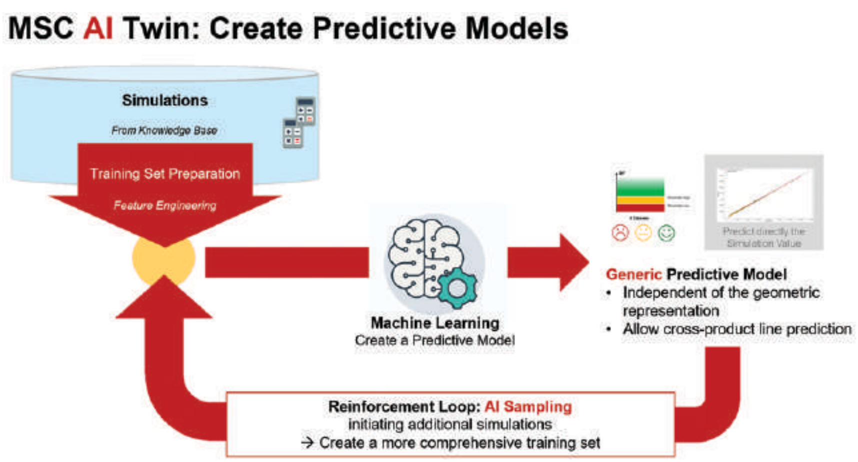


Figure 3. MSC AI Twin

Road Testing or Simulation? – The Billion-Mile Question for Autonomous Driving Development

By **Dr. Luca Castignani, Chief Autonomous Strategist, MSC Software**

Figure 1. Autonomous Driving is a Fast Growing Industry

Just two years ago, the future looked so bright for autonomous vehicles and everything was coming at a fast pace. Most auto OEMs, Tier 1 suppliers and hundreds of start-ups around the world presented their aggressive plans to bring self-driving vehicles on our roads. Traditional OEMs were a bit more cautious, but new players were very bold in their announcements, which is understandable considering that they had to convince their investors that the future of Mobility-as-a-Service (MaaS) was coming very soon.

And then, something tragic happened early in 2018. Something that had not been foreseen by many people, blinded by the hype that was burning millions of dollars on a daily base. A vehicle used by Uber to perform self-driving test (it would be wrong to call it "a self-driving vehicle", since nothing like this exist as of today, and what we see on our streets are just "test platforms") hit and killed a woman crossing the street in Arizona.

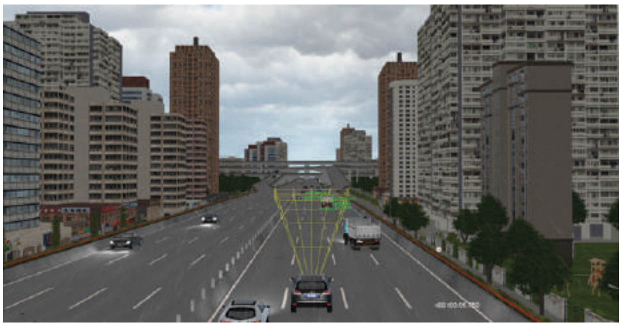


Figure 2. Simulating Autonomous Vehicle driving on a highway in Beijing, China. Simulation done in VIRES VTD.

The reaction from the public was immediate and strong, which put into question not just Uber but also the whole self-driving effort (NVIDIA lost 10% of its market value in the following 2 weeks). The Washington Post titled "Fatal Uber crash spurs debate about regulation of driverless vehicles" [1] and the Guardian "Uber crash shows catastrophic failure of self-driving technology" [2].

What should be the lesson learned from this case? A LinkedIn user expressed it in the best way: "It is completely unacceptable that undesirable beta software is being tested on the roads. This is not an online game where you have several lives".

Achieving autonomous vehicle functionality and safety requires millions of tests to cover all driving scenarios, and there is no way to get even close to that if not extensively using (to an extent never dared in the history of engineering) simulation in the virtual world. Trying to achieve level 5 autonomy only (or mostly) with road testing is as useless as boiling the ocean.

We are engineers, so let's do some basic mathematics. According to the US traffic accident reports [3], statistically, there is one person killed on the road for every 140 million kilometers driven. Therefore, to statistically prove (with 95% confidence) that an Autonomous Vehicle is as good as a human driver, it has to be tested for 415 million kilometers without causing any death [4].

Many self-driving enthusiasts claim that "autonomous vehicles will reduce the number of people killed on the road by a factor of 20" [5]. But this a very bold statement that needs to be proven before being accepted by the general public, regardless of how appealing it sounds. After all, we are scientists, and we believe in data. And the data that we need to prove that the

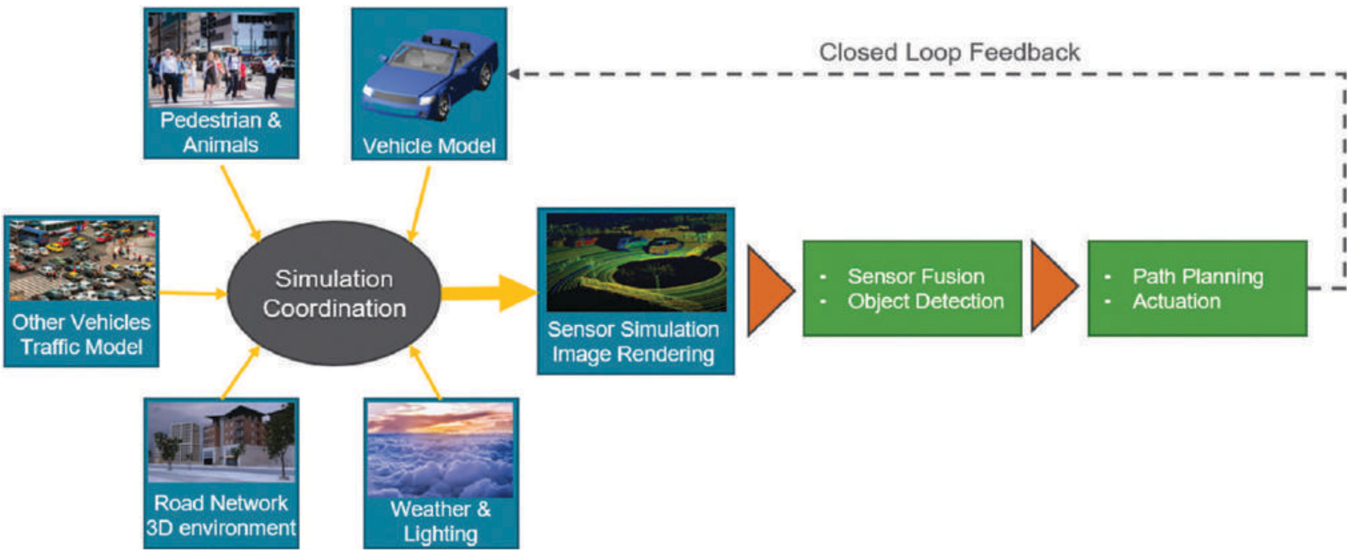


Figure 3. The end-to-end autonomous simulation workflow

autonomous vehicle is “as good as” a human driver, is 415 million kilometers. Do you claim your system is 20 times better? To statistically prove it, show me your results after 8 billion kilometers of testing!

By the way, every time someone modifies the position of a single sensor, the counter has to restart! Every time that some vehicle characteristic is changed (e.g. wheelbase, mass…), the counter has to restart. And every time a piece of software is updated… the counter has to restart!

Is there still someone that believes that road testing can bring you even close to full autonomy? Does anyone still believe that accumulating 1 million kilometers of road testing is a goal to be celebrated?

Take Waymo as an example (for the few of you who do not know them they are the “self-driving branch” of Google and they are considered the leaders in this space). Well, in 10 years they have accumulated 16 million kilometers of road data (really an outstanding outcome, even if most of these kilometers have been accumulated on sunny days in California and Arizona); at this rate, the richest company in the world would need over 200 years to prove they are “as good as human drivers”. And that’s why besides the road testing, Waymo is also running a fleet of 25 thousand virtual cars 24/7, simulating 13 million kilometers per day [6]. “Computer simulations are actually more valuable, as they allow manufacturers to test their software under far more conditions and stresses than could possibly be achieved on a test track.” said Ron Medford, Google’s safety director for the self-driving car program.

Everyone understands the necessity for road testing, but at the same time, we should notice that there are obvious drawbacks. Besides being slow and potentially dangerous if the testing is done prematurely, road testing is not repeatable or controllable, which are essential for autonomous system development.

To solve these issues, engineers tend to leverage the proving ground, which is much more repeatable. Moreover, real sensors can be evaluated on an actual vehicle. However, one of the disadvantages for the proving ground is the limited number of scenarios that engineers can test with. Each proving ground usually contains a set of scenarios and generally speaking it is slow and costly to build/construct new scenarios in the proving ground.

Now let’s take a look at simulation or virtual testing. In my opinion, there are a few key reasons why simulation is more applicable than road testing or proving ground for autonomous system development, especially in the initial phases of the project.

First, virtual testing is more scalable when it comes to cost. A fully equipped autonomous vehicle can cost up to half a million dollars, so a fleet of 200 vehicles would mean a 100 million dollars investment in the hardware itself (vehicles, sensors, data storage, wiring…). On the other hand, scaling virtual testing only requires you to have software licenses and CPU/GPUs to run the simulations, which is generally 100 times cheaper. Not to mention the operational cost to manage such a large fleet of vehicles (drivers, insurance, workshops, maintenance…). As an example of this scalability, BMW recently announced their new High-Performance-Cluster dedicated to the development of Autonomous Vehicles with more than 100,000 cores and more than 200 GPUs [7].

“Many self-driving enthusiasts claim that ‘autonomous vehicles will reduce the number of people killed on the road by a factor of 20.’ But this a very bold statement that needs to be proven before being accepted by the general public, regardless of how appealing it sounds.”

Secondly, the ground truth is always available in virtual testing. In the virtual environment, you always know if it is a pedestrian or if it is a car in front of you, and there is no need to hire service companies to do annotation/labeling for the road data that you collect from testing. When it comes to one billion miles of road data that is requested to validate the autonomous system, it is simply infeasible to annotate it all with human labor.

Thirdly, with simulation, engineers would be able to test the functions of the controller software in the early design stages. One would be able to test the different functions of the software separately with model-in-the-loop simulations without having to wait for the entire control system to be completed. Since you can replay the virtual scenarios as many times as you want, it’s much easier and cheaper to analyze, debug or iterate the core algorithms without having to consider the nuances of the actual production software.

Finally yet importantly, it is much more convenient to create permutations of a situation with virtual testing. Engineers can easily repeat the same test with a different set of parameters, like more pedestrians, higher speed, less sensor visibility, lower road friction, and many more. Permutations of a few basic scenarios with multiple parameters creates thousands of scenarios. And that's the key to ensure robustness and reliability of driving algorithms.

Autonomous Vehicle (AV) simulation is different from traditional vehicle simulations in a sense that apart from the vehicle itself, also the "environment" in which the vehicle operates is fundamental to assessing how it copes with all driving situations. The "environment" of an AV is quite rich (and sometimes crowded) as it includes all other vehicles, pedestrians, animals and of course the road, the sidewalks, buildings and even weather conditions. So, let's take a closer look into all these components.

To start with, the engineers need a vehicle model which represents the same dynamics characteristics as the actual vehicle. When you train the AI controller to drive the actual vehicle, the vehicle model needs to incorporate not only the correct mass and engine power, but also other correct behaviors like braking efficiency, or the load transfer during cornering events. All these performances are heavily influenced by the fundamental suspension designs (dampers, antiroll bars…) and the tire-road interactions.

Besides the vehicle model, the 3D environment also needs to be carefully constructed. 3D environments include the road network, which defines the space that the vehicle can occupy, and when and how the vehicle can occupy each lane. Besides the road itself, the immediate surroundings of the road is equally important. Trees and bushes can obstruct the view of the traffic signs, pedestrian from the sidewalks may suddenly decide to cross the street, buildings on the side of the street may cast shadows on to the road or reduce GPS accuracy. All these elements have to be realistically modeled to properly set the scene where the actions take place.

Of course, the autonomous vehicle shares the road with other vehicles, which can be bicycles, motorbikes, cars, buses, trucks with trailers, Segway, a police officer on a horse or anything else. Anything that is allowed to be driven on the road should be included in this case. And any of those participants might have their own way of interacting with the rest of the traffic. For example, a motorcycle splitting lanes during a traffic jam, while a large truck can easily get stuck in the traffic because of its slow acceleration, and a cyclist might decide to move from the sidewalk to the middle of the road to make a left turn. It is important that all those traffic participants be captured in their unique ways of maneuvering.

The pedestrian and their behaviors also need to be modeled, especially the way they interact with the oncoming traffic. The engineers need to reproduce the gestures of the pedestrians, for instance, whether or not they are watching the traffic, when they are distracted by texting on the phone while crossing the street. Animals' behaviors can be even more unpredictable, like jumping in front of the vehicle erratically, getting stuck in the middle of the road, or staring at the car when it's approaching.

The last important factor one needs to consider for the environment simulation is weather and lighting, which is critical since it impacts the way sensors perceive the scene. When it's raining outside, the vehicle needs to slow down because the driver's vision and the road friction have been changed. With the low-lying sun during sunset or sunrise, human driver needs to wear sunglasses because otherwise he/she couldn't really see the road clearly. Similarly, they also affect sensors like cameras, RADARs, or LiDARs. Fog reduces the visibility of a camera (and absorbs energy from a RADAR) and LiDARs are sensitive to rain drops since they scatter the laser beams.

Actually, the perceived sensor data is the most valuable piece of information that the AV simulation provides. With this data accurately available, engineers can focus on the following phases of the Autonomous Driving development.

Figure 4. Simulating pedestrian crossing the road while on cell phone. Simulation done in VIRES VTD.

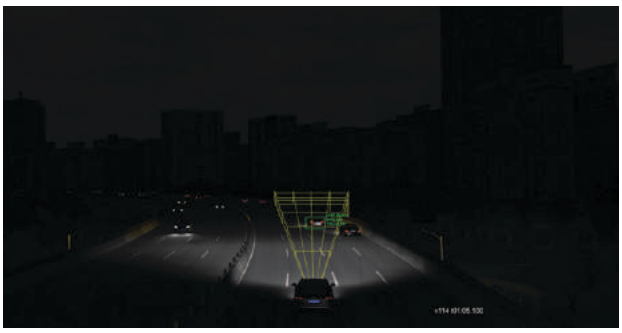


Figure 5. Simulating vehicle driving during evening. Simulation done in VIRES VTD.

The first step is the so-called “sensor fusion” phase, in which data from different sensors is combined to calculate accurate position and orientation information. From the camera, the object is recognized, and when you associate the laser point cloud with the object, the distance to the object can be measured with LiDAR. And RADAR can even provide the speed of the object.

Now that the engineers have a clear understanding of the surroundings of the Ego vehicle, they can move on to the next phase, which is typically called “path planning”. With the understanding of the pedestrians and the other vehicles in the traffic, the engineers need to predict what those other traffic participants will do in the next few tenths of second to the next few seconds. And essentially the vehicle needs to decide at that point what the safest thing is to do to cope with the situation.

Even with the sensor fusion, sometimes the situation is still not 100% understood by the AV (autonomous vehicle). If the vehicle is driving on the highway on a sunny day, all the sensors are giving the correct information and the vehicle has a clear view of a long distance ahead. But imagine if the autonomous vehicle is driving on a crowded street in New York during rush hour on a foggy day, you can’t always tell if there’re two pedestrians or three pedestrians in front of you. When the vehicle makes a decision as to which path to follow, it needs to consider not only the destination, but also what the safest route is to get there.

After the safest path is identified, it is time to decide on how to actuate the vehicle, which means how to apply the throttle, the brake, the steering wheel to follow that path, or how to adjust the damping in the suspension system to ensure a smooth ride. This is the so called “actuation phase” and is the playground of very specialized engineers that master control theory of ground vehicles.

In the virtual simulation workflow, all this information is being provided to the vehicle dynamics model as closed loop feedback. And based on those inputs with torques/forces, the vehicle model predicts its updated displacement, velocity and orientation to interact with the surroundings (including oncoming traffic or pedestrians that may be triggered to cross the street) and the simulation loop proceeds.

VIRES Virtual Test Drive (VTD) provides all the ingredients necessary for engineers to perform the autonomous driving simulation, and at the same time, VTD is compatible with not only MSC Software’s internal technologies, but also with a number of 3rd party software. As an example of this, while VTD offers 2 different embedded techniques to represent the vehicle dynamics (with varying speed of simulation and accuracy of results) it can also be combined with Adams Car (the de facto standard in vehicle dynamics simulation) or with any other vehicle dynamics software. Same applies to the traffic models: VTD offers the most comprehensive traffic simulation capabilities in the industry (driving style of every vehicle can be set according to a number of parameters, and thousands of vehicle can be simultaneously simulated if necessary); nonetheless other traffic models can be incorporate into VTD as well, such as SUMO [8] or PTV Vissim [9].

Autonomous Driving is one of the most exciting, yet daunting tasks in the next decade to come. Road testing alone will never get us anywhere close to the billion miles of validation needed to ensure the safety of an autonomous car. In order to develop an autonomous driving system that can truly save tens of thousands of lives, comprehensive simulation of the real world is the key to success.

References

1. Washington Post https://www.washingtonpost.com/local/trafficandcommuting/deadly-driverless-uber-crash-spurs-debate-on-role-of-regulation/2018/03/23/2574b49a-2ed6-11e8-8688-e053ba58f1e4_story.html?utm_term=.d8bf81d9de6d
2. The Guardian https://www.theguardian.com/technology/2018/mar/22/self-driving-car-uber-death-woman-failure-fatal-crash-arizona
3. NHTSA https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812603
4. Rand Corporation https://www.rand.org/content/dam/rand/pubs/research_reports/RR1400/RR1478/RAND_RR1478.pdf
5. Assuming that about 95% of road deaths are caused by human error
6. https://waymo.com/safety/
7. https://www.press.bmwgroup.com/global/article/detail/T0293764EN/the-new-bmw-group-high-performance-d3-platform-data-driven-development-for-autonomous-driving?language=en
8. SUMO is an open source traffic model originally create by DLR. For more information, please see https://sumo.dlr.de/index.html
9. This functionality will be added in VTD 2019.1 http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/

Marius Dupuis

Managing Director
VIRES, Simulationstechnologie GmbH an MSC Software Company

Marius Dupuis is the Managing Director, VIRES Simulationstechnologie GmbH, an MSC Software Company. Dupuis holds a degree as "Diplom" engineer in aerospace from Stuttgart University.

He began his professional career in 1995 at Eurocopter, Germany in the field of flight simulation for engineering purposes. Dupuis co-founded VIRES in late 1996 and began working full-time as General Manager of VIRES in 1998. He has been with VIRES ever-since and is fully dedicated to participating in the development of VIRES software.

Dr. Luca Castignani

Head of Autonomous Mobility Strategy, MSC Software

Dr. Luca Castignani is the Head of Autonomous Mobility Strategy, MSC Software. He has been designing customer-oriented solutions in NVH and MBD. In his previous role, Dr. Luca served in Ferrari Auto leading Vehicle Dynamics, NVH and later as the head of CAE. He was in charge of designing the new generation of Ferrari suspensions that fits the whole range of cars. Dr. Luca has a PhD in Vehicle Dynamics from Politecnico di Milano (Italy) in cooperation with Pirelli Tires.

Dr. Keith Hanna

Vice President, Marketing

Dr. Keith Hanna is the Vice President, Marketing of MSC Software. Dr. Hanna brings over 25 years of experience in the CFD, CAE, EDA and PLM industries, spanning a wide range of global technical and marketing roles inside Siemens PLM, Mentor Graphics Corp., ANSYS Inc. and Fluent Inc.

His career prior to engineering simulation included practical experience of the metallurgical and mining industry at Br. Steel and De Beers. He has both BSc and PhD engineering degrees from the University of Birmingham in England and is a respected commentator on the CFD/CAE industry, a pioneer of CFD in sport, and a former member of the Executive Committee of the International Sports Engineering Association.