

Editors

Andras Kemeny

Jean-Rémy Chardonnet

Florent Colombet

DSC 2026 EUROPE^{XR}

Driving Simulation Conference & Exhibition

Product Solutions Book

Palais des Congrès

Antibes, France

September 16 – 18, 2026

ISBN 979-10-988022-0-1

Direction

Andras Kemeny
Driving Simulation Association

Editors

Andras Kemeny	<i>Driving Simulation Association</i>
Florent Colombet	<i>Renault</i>
Jean-Rémy Chardonnet	<i>Arts et Métiers Institute of Technology</i>

Technical committee

Chairwoman:

Luz Amanda Garcia Galeano (EBUSCO, Netherlands)

Technical committee members:

Omar Ahmad	NADS (United States)
Florent Colombet	Renault (France)
David Defianas	Stellantis (France)
Marius Dupuis	ASAM (Germany)
Vincent Honnet	SystemX (France)
Andras Kemeny	Driving Simulation Association (France)
Martin Peller	BMW (Germany)
Stéphane Régner	Renault (France)
Martin Sevenich	Aumovio (Germany)
Georg Stettinger	Infineon (Germany)
Sebastian Wagner	BMW (Germany)

ISBN 979-10-988022-0-1

© Driving Simulation Association

Preface

It is with great pleasure that the Driving Simulation Association is editing this DSC Europe 2026 XR Product Solutions Book, a collection of industrial solution papers and posters presented at the 2026 edition of the DSC Conference.

They are a well-known and significant sample of the most recent works carried out in driving simulation and XR and technology, driving simulator design, motion cueing algorithms and visual rendering. They show the recent revolution of engineers' experimenting and testing in the use of Driving Simulation. These new trends are the direct consequence of the extraordinarily fast development of driving simulation as well as the development of autonomous and connected vehicles. There is an increasing place for the development of scenario standards and this Product Solution Book witnesses again this trend.

Scenario modelling and control are an essential part of the presented papers, showcasing the increasing role of AI, machine learning in driving and massive simulation of ADAS and CAV (Connected and Autonomous Vehicles) Verification and Validation (V&V) and Mixed Reality in driving simulation, both in Motion Cueing & Control and Scenario Modelling and Generation. Trustworthiness is a crucial part of AI generated features as well as all the necessary control of the used technics, as we have seen recently with OpenAI AI tools going "rogue" and escaping the service provider's control.

Industrial keynotes, showing the weight of the industrial simulation chains, simulator centres and products, as well as ADAS and connected and autonomous vehicles are not integrated in the proceedings, but the reader may access through the conference website, <https://dsc2026.org/keynote-speakers/>, the presentations given by OEM and supplier representatives. The same way, industrial pitches are directly accessible for all, see: <https://dsc2026.org/industrial-pitches/>.

We would like to thank all the reviewers and members of the DSC Technical Committee; without their work it would not have been possible to guarantee the selection and quality of the published papers. They are too many to cite all here, but you can find them on our conference site. See: <http://dsc2026.org/committees/>.

We wish you a great pleasure in reading these papers bringing for you an insight in the latest trends in immersive simulation science and technology.

The Driving Simulation Association Executive Board

Andras Kemeny, Jean-Rémy Chardonnet, Florent Colombet, Stéphane Espié

Table of Contents

DIL SIMULATION	7
From On-Road Testing to Ride Replay: Validation of Simulator-Based Ride Comfort Assessment using a Specialized Driving Simulator <i>Stefan Jeindl, Philipp Hofmaninger, Paul Kovacevic, Leo Nolan, Gavin Farmer, Zoltan Ferenc Magosi, Ioana Victoria Koglbauer, Arno Eichberger</i>	9
Engineering the Full Spectrum: The Evolution of VI-grade Driving Simulation <i>Stefano Bernini, Francesco Ambrogio, Fabio Formaggia</i>	15
An ISO Initiative Towards Standardizing Driving Simulators <i>Tomoya Chida, Martin Fischer, Masaki Fujimoto, Chantal Himmels, Kazuki Kuwabara, Hitoshi Kyogoku, Johannes Lindner, Hideyuki Muramatsu, Fabian Ruß, Marcus Schmitz, Martin Sevenich, Atsushi Tsubouchi, Stefan Vorderobermeier</i>	17
ADAS	21
Scenario-Based Approach in System Engineering: Toward an Assessment of Coverage <i>Christophe Bohn, Erwan Revert, Faouzi Adjed</i>	23
ASAM QSQ: Quantifying Simulation Quality for Credible Virtual Validation <i>Yash Shah, Diego Sánchez Lázaro</i>	29
From Concept to Validation: Mastering Scene and Scenario Fidelity with the RoadRunner Product Family <i>Simone Hämmerle, Philipp Hafemann</i>	33
Developing ADAS with High-Fidelity Vehicle Models and Integrated Controllers <i>Alyssa Scheske, Ahmet Keles, Bharath Kaimal</i>	39
Test Management Process within SCANeR and the 3DEXPERIENCE Platform <i>Reine Gros, Sandrine Loembe, Franck Corbier</i>	45
HIL DIL SIMULATION DESIGN	49
Extending a Modular Simulation Framework to Support Immersive Virtual Reality for Human-in-the-Loop Driving Studies <i>Peter Baker, Joseph Mitchell, Emil Chodowiec, Xizhe Zhang, Siddhartha Khastgir, Paul Jennings</i>	51
Development Environment for the Validation of Steer-by-Wire Systems <i>Alexander Ahlert, Martin zur Heiden, Philipp Schmidt, Henning Kemper, Thomas Karasiewicz, Guillaume Miller</i>	57
A Distributed Driving Simulation Framework utilizing Data Distribution Service <i>Christian Bauer, Chris Schwarz</i>	65
A HiL-Driven Brake Pedal Feedback Architecture for Driver-in-the-Loop Simulation <i>Raphael Groß, Anton Tworek, Bernhard Riegert, Peter E. Pfeffer</i>	71
SENSORS IN DRIVING SIMULATION	77
Active Infrared with SE-RAY-IR – SCANeR Studio <i>Florian Faucher, Antoine Boudet, Thomas Nguyen</i>	79

A Unified Phenomenological Sensor Model for Real-Time Raw Data Generation in Autonomous Driving Simulation	85
<i>Ahmed Yousif, Mohamed Elsobky</i>	

Radar sensor simulation validation	93
<i>Florian Faucher, Alain Servel</i>	

SIMULATION DESIGN 99

Creating OpenDRIVE maps for 3D models using the ODRoNeS plugin for Blender	101
<i>Albert Solernou, Jorge Garcia de Pedro, Hsuan Chao</i>	

A Low-Cost Driving Simulator Platform for Integrated Vehicle Comparison, HIL testing and Public Engagement	105
<i>Stefan Haehnlein, Judith Frueh, Mirko Ronecker, Christian Kloeffer</i>	

AI & MACHINE LEARNING FOR SCENARIO GENERATION 109

An AI-Based Validation Intelligence Layer for Regulation-Ready Autonomous Driving	111
<i>Yves Peirsman, Stephen Lernout</i>	

POSTERS 115

Virtual Reality Driving Simulator for Education – Form Factor, Cybersickness and Perception	117
<i>Helen Loeb, Ayagoz Smagulova, Matan Rosenberg, Eduardo De La Pena, Oskars Sniegs</i>	

Transforming Simulation with Reality-Based Neural Rendering & World Models	119
<i>Tolga Kart, Piotr Sokolski</i>	

An Empirical UDP Network Assessment of Real-Time Latency and Data Integrity for a Distributed Heterogeneous Simulation Setup	123
<i>Machiel van der Stelt, Thomas Bräunl, Paul Roberts</i>	

COMPANIES' INDEX 125

DIL SIMULATION

From On-Road Testing to Ride Replay: Validation of Simulator-Based Ride Comfort Assessment using a Specialized Driving Simulator

Stefan Jeindl¹, Philipp Hofmaninger¹, Paul Kovacevic², Leo Nolan³
Gavin Farmer³, Zoltan Ferenc Magosi², Ioana Victoria Koglbauer², Arno Eichberger²

(1) Magna Steyr Fahrzeugtechnik, Graz, Austria

(2) Graz University of Technology, Graz, Austria

(3) Dynisma Ltd., Bristol, United Kingdom

Abstract - The Dynisma DMG-X is a high-bandwidth driving simulator featuring a system latency of 3-4 ms, a motion bandwidth exceeding 100 Hz, and full six-degree-of-freedom (DOF) capability, purpose-built for ride comfort and NVH evaluation. Integrated into Magna's engineering portfolio and installed at the Advanced Driving Simulation Center (ADSC) at Graz University of Technology, the DMG-X enables systematic ride comfort assessment through Ride Replay, a method that reproduces 6 DOF motion signals derived from real vehicle measurements and high-fidelity multi-body-simulation (MBS) without requiring real-time model execution. This paper presents the validation of this product solution through a three-stage approach: manufacturer acceptance tests against tightly specified motion fidelity requirements, objective input-output correlation analysis across both time and frequency domains, and a controlled within-subject study, comprising professional test drivers and amateurs rating ride comfort and realism across various road surface types and vehicle setups. The findings demonstrate that the DMG-X combined with Ride Replay is a viable tool for frontloading subjective ride comfort evaluation, supporting a more cost-efficient and time-effective development process and enabling informed decisions prior to the availability of physical prototypes.

Keywords: ride comfort, driving simulator, multi-body-simulation

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

The research in ride comfort of vehicles has traditionally relied on objective evaluation of virtual simulation results using detailed multi-body-simulation (MBS) models. These objective assessments are subsequently complemented by subjective fine-tuning during on-road testing, once the first physical prototypes become available. To enhance quality while reducing both cost and time-to-market, the development process is increasingly shifting towards virtual methods (Schwarzhuber, et al., 2026). The objective is to identify key aspects of the subjective evaluation prior to the availability of physical prototypes. Nevertheless, the limited real-time performance of detailed MBS models, alongside the constraints of state-of-the-art driving simulators, particularly in reproducing vibrations with frequencies up to 100 Hz, have so far hindered the partial replacement of physical on-road testing. The integration of the Dynisma DMG-X with Ride Replays derived from high-fidelity MBS now enables systematic virtual evaluation, allowing for early-stage assessment of vehicle ride comfort. The proposed solution facilitates subjective insights into vehicle dynamics and ride quality well before physical

prototypes become available, supporting a more efficient and cost-effective development process.

Product solution

Dynisma's Motion Generators (DMGs) are designed to deliver maximum motion information to the driver through high bandwidth across low and high frequencies, low latency, and excellent signal-to-noise ratio, while remaining free of undemanded jerk, cross-axis coupling, and exhibiting an isotropic response across all six degrees of freedom (DOF).

The performance requirements of the project partners for ride and NVH evaluation led to the development of a new motion generator, the DMG-X (shown in Figure 1), whose key specifications are summarized in Table 1. Of particular significance are the motion bandwidth exceeding 100 Hz and the system latency of 3-4 ms, both of which are reflected throughout all aspects of the system design, as detailed in the following.

Mechanical

The design principle prioritizes direct load paths. Mo-

Table 1: Basic specifications of DMG-X

Nominal payload	250-750 kg
System response	
Motion bandwidth	>100 Hz
Motion latency	3-4 ms
Maximum excursion	
Longitudinal	500 mm
Lateral	500 mm
Heave	400 mm
Roll	±15°
Pitch	±15°
Yaw	±15°
Combined Lateral-Yaw	150 mm @ 15°
Combined Lateral-Longitudinal excursion radius	250 mm
Maximum velocity	
Translational	1.5 m/s
Rotational	>100°/s
Maximum acceleration	
Surge, Sway, Heave	15 m/s ²
Roll, Pitch, Yaw	>1000°/s ²

tion is generated by rockers with integral curved linear motors, each connected to the payload via rigid links pivoted at both ends, enabling full 6 DOF motion. While the resulting platform shares its general form with a Stewart platform, it operates without ballscrews or gearboxes, which are known to introduce backlash or mechanical preload and typically limit conventional Stewart platforms to approximately 30 Hz (Nieuwenhuizen, et al., 2008).

Control system

A vehicle model is executed on a hard real-time machine, providing 6 DOF acceleration signals to a safety-rated real-time industrial computer. These accelerations are converted to platform demands via motion cueing algorithms, and an inverse kinematics model generates the corresponding actuator commands. Communication between all system components is implemented via EtherCAT.

Payload

The required motion accuracy further dictated a stiff, resonance-free payload structure, realized as a four-sided carbon fibre *QuadPlate* with internal stiffening. All payload and motion generator components were systematically analyzed with respect to strength, stiffness, modal performance, and fatigue life.

Driving Simulation

Vehicle modelling is currently implemented in IPG CarMaker with Cosin FTire, running on a Concurrent real-time machine. Graphics and audio are provided by rFpro and a Varjo XR-4 headset. Driver controls comprise a pedal system and a Sensodrive steering system. The platform supports integration of additional software and hardware components as required.

Fields of application

At Magna, the Dynisma DMG-X is employed alongside physical prototypes throughout the full vehicle development process, representing the next step from objective MBS-based evaluation and subjective on-road testing. Key advantages include accelerated decision-making through frontloading, in-



Figure 1: DMG-X Driving Simulator at the ADSC, Graz University of Technology

creased cost-efficiency compared to physical prototypes, and the ability to modify vehicle setups and environmental conditions within seconds. Application areas include ride comfort, vehicle handling, and functional development through to autonomous driving and UI/UX studies. The platform is further offered to customers as part of Magna's engineering services portfolio.

Furthermore, the DMG-X forms a central component of the Advanced Driving Simulation Center (ADSC), established through the strategic partnership between Magna and Graz University of Technology. Beyond its role in vehicle development, it supports the education of automotive engineering students and provides a platform for both Master's and PhD research including vehicle modelling, control system design, and human perception. The simulator is further integrated with the existing suspension test bench, enabling consistent validation of chassis systems across component-level and full-vehicle assessments.

Methodology

Acceptance tests

Standard acceptance testing of the Dynisma Motion Generator includes frequency response and phase lag characterization. While conventional motion generator specifications typically require the response to remain within ±3 dB of the input (Kemmerling, 1979), the project partners specified a tighter tolerance of ±1.5 dB up to 60 Hz and ±3 dB up to 100 Hz. Bode plots of the platform acceleration frequency response function (FRF), defined as the ratio of measured to demanded acceleration across the frequency spectrum, are used to evaluate this requirement, where a magnitude close to 0 dB indicates a near 1:1 relationship between demand and measurement, and a phase above the dotted line indicates a latency below 4 ms. As shown in Figure 2, the system was found to exceed this requirement considerably.

The partners' focus on ride and NVH evaluation additionally prompted a series of non-standard validation tests, including replay of recorded data to defined accuracies and repeated frequency response measure-

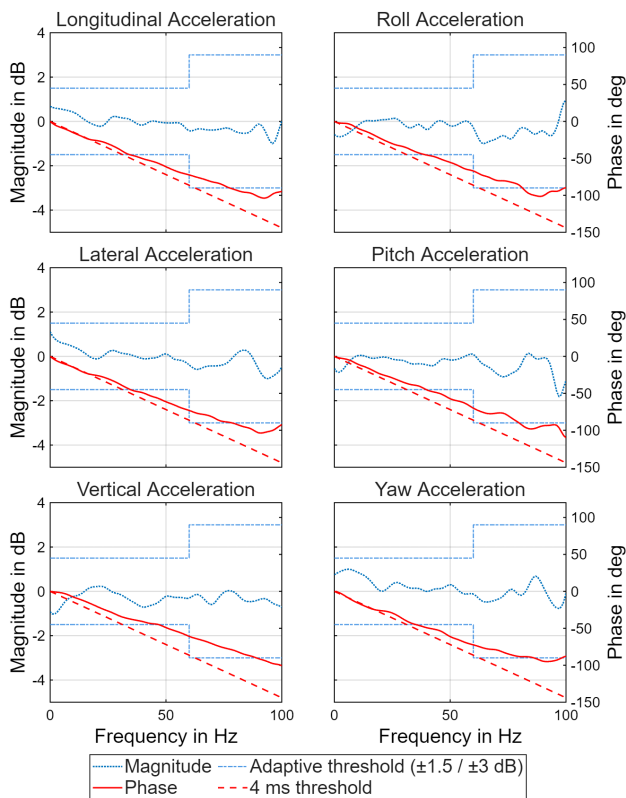


Figure 2: Bode plots of platform acceleration frequency response function (FRF)

ments at off-centre platform positions, aiming at validating motion fidelity across the full motion envelope.

Objective evaluation of Ride Replay

The primary methodology employed for ride comfort assessment within the driving simulator is the Ride Replay, which enables the reproduction of previously recorded motion data by replicating 6 DOF dynamics, comprising translational accelerations and rotational rates, at the driver seat position. Input signals for the Ride Replay are derived from two complementary sources: real vehicle measurements conducted on physical proving ground infrastructure and virtual motion signals generated through high-fidelity multi-body simulation. This approach provides a continuous and systematic means of evaluating ride comfort throughout the vehicle development process, enabling direct comparison against real benchmark and reference vehicles independent of prototype availability.

To ensure comparability and practical relevance, established road excitation profiles are employed, including single obstacle events and stochastic road surfaces. These profiles, which have been extensively utilized in vehicle development practice, are available both as physical road surfaces at the Magna proving ground as well as digitized road profiles for use in simulation environments.

Real vehicle measurements are acquired in accordance with the Magna standard ride comfort instrumentation protocol, with dynamic state sensors mounted on the driver seat rail. The measurement set encompasses all motion signals required for sim-

ulator Ride Replay input, specifically translational accelerations and rotational rates across all 6 DOF. As these signals are systematically recorded during all ride comfort evaluations conducted at Magna, a comprehensive ride comfort database has been established over the preceding years.

Virtual motion signals are generated using a high-fidelity Adams Car multi-body-simulation model of a series production vehicle developed by Magna, which serves as a common motion reference platform for both simulation and real vehicle measurements. The model has undergone extensive validation against physical vehicle measurements and captures comfort-critical dynamic phenomena through the incorporation of flexible bodies for the body-in-white and primary chassis components, frequency-dependent and hydro-elastic bushings, a fully parameterized powertrain and engine mount system, as well as a high-fidelity tire model based on Cosin FTire. Although the computational demands of this high-fidelity model prevent real-time execution, it does not compromise Ride Replay functionality, as motion signals are generated in offline processing. Consequently, the Ride Replay methodology also facilitates the integration of computationally intensive models, such as active damper control systems, which would otherwise be incompatible within real-time simulation constraints.

To assess the reproduction fidelity of the simulator, a measurement campaign was conducted in which predefined input signals were directly compared to the resulting platform motion. For this purpose, the same 6 DOF motion signals used as Ride Replay input were simultaneously recorded at the simulator platform using onboard sensor instrumentation. The dynamic behavior of the system was then evaluated by means of input-output correlation analysis, enabling a quantitative assessment of amplitude reproduction and frequency content between prescribed and realized motion.

Subject study

A within-subject study (N=21, M=42.5 years, 19% female, 15,200 km/year) was conducted to subjectively evaluate the ride comfort across different vehicle setups in the simulator. The assessment included two groups: 10 professional test drivers (all male, M=44.5 years) and 11 amateurs (40% female, M=40.5 years), all providing their informed consent. Participants completed 15 simulated rides (3 tracks with 5 vehicle setup variants each) in Ride Replay mode without visual stimuli, focusing exclusively on ride comfort. The tracks covered ramp, rough asphalt, and rural road surfaces, while the variants included a real vehicle measurement and four simulation setups derived from a high-fidelity MBS model, covering a typical tuning range for ride comfort assessment. Each setup variant was compared against a constant reference setup representing the baseline vehicle configuration, following the structure reference - variant - reference. Ride durations were 80 s (ramp, rough asphalt) and 110 s (rural road), while the total test duration was 30 minutes. After each ride, participants rated comfort relative to the reference (Question Q1: worse/equal/better), vibration intensity (Question Q2: 7-point Likert scale, 1 = low to 7 = high intensity), and simulator realism (Question Q3: 7-point Likert scale, 1 = low to 7 = high real-

ism). Ride sequences were randomized to counterbalance potential learning effects. The *Misery Scale* (MISC) questionnaire (Bos, et al., 2006) was administered pre- and post-test for monitoring motion sickness effects, using an 11-point rating scale ranging from 0 (no problems) to 10 (vomiting). Biosignals were recorded via the biosignalsplux system (PLUX Biosignals, 2025) at 100 Hz for separate analysis of correlations with perceived ride comfort. Statistical analysis used the software package R with mixed-effects models due to the repeated-measures design. Subjects and their individual rating tendencies were included as random effects, allowing fixed effects such as variant, track, and group to be estimated more accurately. A generalized linear mixed model (GLMM) with binomial family was applied for Q1, as the proportional odds assumption required for a cumulative link mixed model (CLMM) was violated, confirmed by the nominal and scale test from the ordinal package. For Q2 and Q3 (due to 7-point Likert scale, confirmed by adequate normality and homoscedasticity) a linear mixed model (LMM) was applied, using the lme4 package (Bates, et al., 2015). The significance of the fixed effects were assessed via likelihood ratio (LR) tests. Post-hoc comparisons applied estimated marginal means using the emmeans package (Lenth, et al., 2026) with Bonferroni correction ($\alpha = 0.05$). Limitations include the limited gender diversity, small sample size ($N=21$), as well as potential fatigue and learning effects not fully mitigated by randomization of ride sequence.

Results and discussion

Objective evaluation results

To characterize the dynamic behavior of the DMG-X, an input-output correlation analysis was performed by directly comparing the prescribed Ride Replay motion signals with the measured platform response. The Ride Replay data presented here are applied across all 6 DOF to the DMG-X motion-cueing interface. Motion-cueing inherently requires the compression of full-scale 6 DOF vehicle dynamics into the constrained workspace of a hexapod, which introduces deviations in peak reproduction and frequency content even when overall correlation remains strong. Furthermore, due to the kinematic architecture of the hexapod, cross-coupling between its DOF could occur, such that excitation along one axis may influence the response in others.

Figure 3 presents the time domain input-output correlation for a Ride Replay of a single obstacle event, obtained using a motion-cueing parameter set tuned for subjective driver evaluation. The results indicate generally strong agreement between commanded and measured platform motion, with deviations apparent at sharp peaks and rapid transients. Supplementary evaluations employing isolated single-DOF excitations reveal improved tracking performance and provide insight into the response characteristics of individual axes, suggesting that further parameter optimization may improve the overall reproduction fidelity of the motion-cueing strategy. For the purpose of subjective ride comfort assessment, the achieved reproduction fidelity is regarded as adequate.

The frequency domain input-output comparison, derived from FFT analysis of a Ride Replay on a rough asphalt track up to 50 Hz, characterizes the system

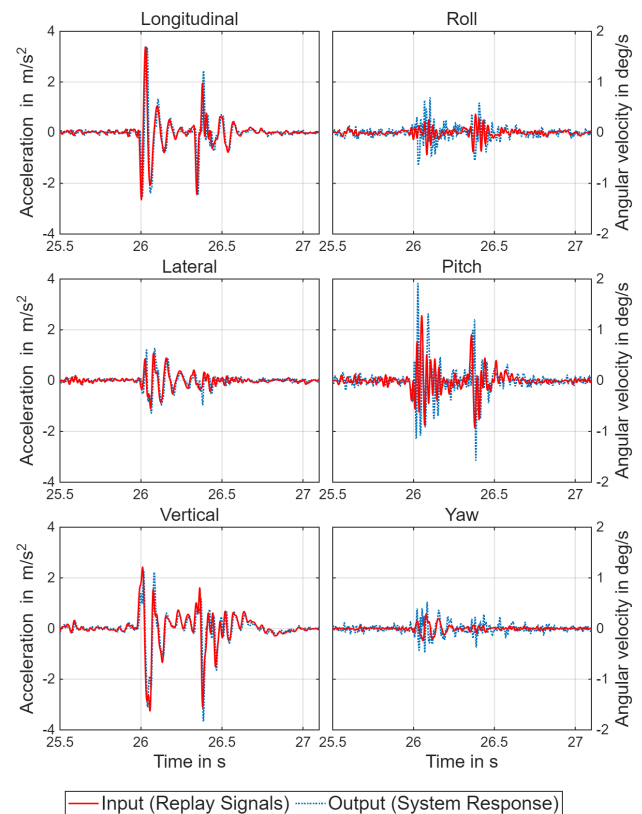


Figure 3: Input-output correlation for single obstacle track - time domain

response across a frequency range relevant to both primary and secondary ride assessment. As illustrated in Figure 4, the measured translational accelerations and rotational rates generally reproduce the dominant spectral content of the input signals, with moderate deviations present across the frequency band.

Overall, the input-output analysis shows that the DMG-X is capable of reproducing the dominant motion characteristics of the prescribed input signals with adequate fidelity for ride comfort assessment, while acknowledging the inherent limitations of hexapod kinematics and motion-cueing compression.

Subject study

Table 2 shows the rating distribution for perceived comfort (Q1) across all variants. The real vehicle measurement (V1) was predominantly rated as more comfortable than the reference (88.9%), suggesting the reference simulation may lack sufficient detail to capture real vehicle behavior. Likely due to the added frequency content, the flexible trimmed body simulation (V4) received mostly worse comfort ratings (61.9%), though some participants rated it better, possibly by distinguishing between primary and secondary ride. Both decreased EDU mount damping (V2) and increased damper curve (V5) yielded inconclusive results, with a majority rating them equal to the reference (50.8%) and significant shares in both directions, suggesting that these variants were not perceptually distinguishable from the reference. Reduced tire pressure (V3) was rated as equal (63.5%) or better (30.2%) by the majority of participants, indicating that the difference was imperceptible to most,

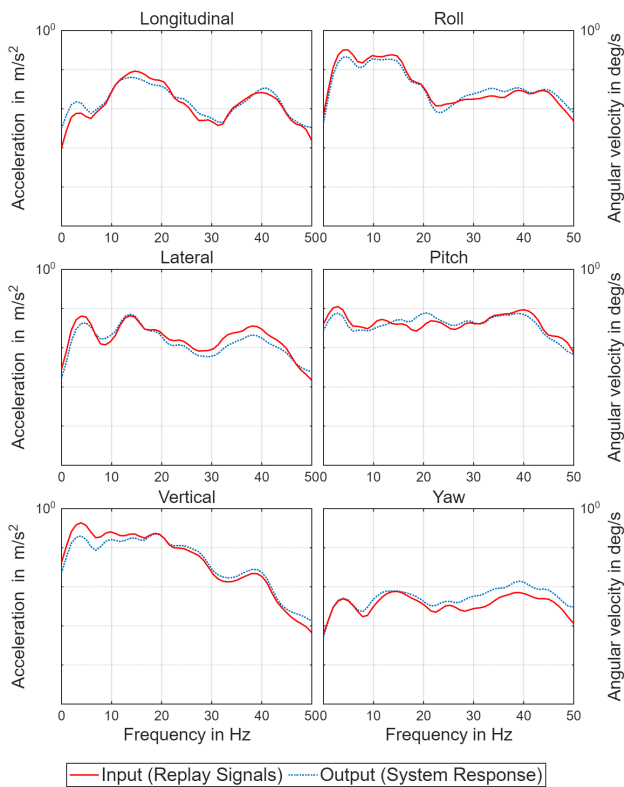


Figure 4: Input-Output correlation for rough asphalt track - FFT

while those who did perceive a difference rated it as more comfortable.

Table 2: Rating distribution between variants for perceived comfort compared to the reference setup (Q1)

Variant	Worse	Equal	Better
V1 (real vehicle measurement)	6.3%	4.8%	88.9%
V2 (EDU mount damping -25%)	30.2%	50.8%	19%
V3 (tire pressure -0.3 bar)	6.3%	63.5%	30.2%
V4 (flexible trimmed body)	61.9%	14.3%	23.8%
V5 (damper curve +25%)	30.2%	50.8%	19%

Table 3 summarizes the key statistical results of the applied GLMM, congruent with the rating distribution of Table 2.

Table 3: Post-hoc results for perceived comfort (Q1)

Variant	P(different)	Sig. different from	p-value
V1	.955	V2, V3, V5	< .001
V2	.492	V1, V4	< .001
V3	.360	V1	< .001
V4	.863	V2, V5	< .001
V5	.492	V1, V4	< .001

The LR test confirmed a significant combined effect of variant and its interaction with group on the ratings ($\chi^2(8) = 90.08$, $p < .001$). Track, group, and all remaining interactions were non-significant for the ratings. V1 and V4 showed the highest probability of being perceived as different from the reference, while V2, V3, and V5 were more likely rated as equal. V1 differed significantly from the ratings of V2, V3, and

V5, but not from V4. Ratings for V4 differed significantly from V2 and V5, while V2, V3, and V5 were not rated as significantly different (all $p = 1.0$). In summary, only the real vehicle measurement (V1) and the flexible trimmed body (V4) were clearly distinguished from the reference, while decreased EDU mount damping (V2), lower tire pressure (V3), and increased damper curve (V5) showed no significant differences from the reference or from each other.

Table 4: Descriptive measures for vibration intensity (Q2) between variants

Variant	Mean	Standard deviation
V1	3.8	1.3
V2	4.2	1.1
V3	4.2	1.1
V4	4.6	1.2
V5	4.4	1.3

Table 5: Descriptive measures for vibration intensity (Q2) between tracks

Track	Mean	Standard deviation
T1 (Ramp)	4.6	1.1
T2 (Rough asphalt)	4.1	1.2
T3 (Rural road)	4.0	1.3

For vibration intensity (Q2), V1 showed the lowest mean ratings ($M = 3.8$) and V4 the highest ($M = 4.6$), as shown in Table 4. These results are consistent with the comfort ratings in Q1, as the added frequency content of the flexible trimmed body variant corresponds to both better perceived comfort and higher vibration intensity, with inverse results for the real vehicle measurement V1. V2 and V3 were rated equally ($M = 4.2$), and V5 slightly higher ($M = 4.4$), consistent with physically expected behavior of the variants. Regarding tracks, the ramp yielded the highest intensity ratings ($M = 4.6$), while rough asphalt ($M = 4.1$) and rural road ($M = 4.0$) were rated similarly, as shown in Table 5.

Table 6: Statistical results for vibration intensity (Q2): significant effects and post-hoc contrasts

Effect	Contrast	Estimate	p-value
Variant	V1 vs. V4	-0.85	< .001
	V1 vs. V5	-0.62	< .01
Track	T1 vs. T3	0.64	< .01
	T1 vs. T2	0.50	< .05
Group × Variant	Professional drivers: V4 vs. all variants	0.86-1.26	< .01
	Amateurs: V2, V5 vs. V1	0.55-0.85	< .05

The analyses show that both variant and track had a significant effect on intensity ratings, with the influence of the variant depending on the participant group (variant × group: $\chi^2(8) = 49.88$, $p < .001$; track × group: $\chi^2(4) = 12.99$, $p < .05$; group-variant interaction: $\chi^2(4) = 19.23$, $p < .001$). Key post-hoc contrasts are summarized in Table 6. V4 was the only variant being rated with significantly higher vibrational intensity by professional drivers compared to amateurs (estimate = 1.1, $p < .001$), while no

overall group effect between amateurs and professional drivers exists (estimate = -0.2 , $p = .42$). In conclusion, the real vehicle measurement (V1) was rated with significantly lower vibration intensity compared to the flexible trimmed body (V4) and the increased damper curve (V5). The other variants were not rated significantly different in terms of vibration intensity. For tracks, the ramp (T1) received significantly higher vibration intensity ratings than rough asphalt (T2) and rural road (T3), while T2 and T3 did not differ significantly. Regarding group effects between professional drivers and amateurs, the professionals rated the flexible trimmed body (V4) with significantly higher vibration intensity compared to all other variants, while the amateurs only perceived the vibration intensity of the real vehicle measurement significantly lower than the decreased engine mount damping (V2) and increased damper curve (V5).

Table 7: Descriptive measures for simulator realism (Q3) between variants

Variant	Mean	Standard deviation
V1	5.3	1.4
V2	5.1	1.5
V3	5.2	1.5
V4	5.0	1.6
V5	5.1	1.5

Table 8: Descriptive measures for simulator realism (Q3) between tracks

Track	Mean	Standard deviation
T1	5.1	1.4
T2	5.2	1.6
T3	5.2	1.5

For simulator realism (Q3), ratings were consistently above the scale midpoint across all variants ($M = 5.0$ – 5.3 , shown in Table 7) and tracks ($M = 5.1$ – 5.2 , shown in Table 8). No significant effects on the realism ratings were found for variant, track, group, or any interaction (all $p = .71$ – $.72$). Additionally, no indications of simulator sickness were observed. Although MISC scores increased from pre-test ($M = 0.19$, $SD = 0.39$) to post-test ($M = 0.8$, $SD = 1.17$), all values remained well below the common exclusion threshold ($MISC \geq 6$) (Irmak, et al., 2021), indicating good tolerance of the Ride Replay mode.

The consistently high realism ratings suggest that simulator movements were perceived as sufficiently realistic regardless of setup or track, supporting its validity for ride comfort assessment.

Conclusion

The proposed solution, comprising the DMG-X driving simulator integrated into Magna's engineering portfolio and embedded in the infrastructure of the Advanced Driving Simulation Center (ADSC) at Graz University of Technology, demonstrates strong potential for improving cost efficiency and reducing development time in vehicle engineering. Validity was assessed both objectively and subjectively, covering manufacturer acceptance tests, input-output correlation analysis, and a subject study on perceived ride

comfort. Requirements were strictly fulfilled, and participants perceived the simulator as realistic across all scenarios.

Use cases for the driving simulator range from real-time applications in vehicle handling, ride comfort assessment, and autonomous driving development to Ride Replay scenarios with high-fidelity vehicle models. The high bandwidth and low latency of the DMG-X are a key enabler for virtual ride comfort evaluation, without which Ride Replay would not be a viable tool for vehicle development purposes.

The Ride Replay method investigated in this work represents an effective means of supporting on-road vehicle testing by frontloading initial subjective evaluations and enabling informed decisions earlier in the development process. This approach has the potential to reduce time to market while simultaneously improving overall product quality. By making simulation results perceptible from the driver's perspective, Ride Replay provides a tool for better understanding and interpreting simulation data. At later development stages, when physical prototype vehicles are available, the method can be used to continuously compare the current vehicle state with previous variants and to benchmark against other vehicles using Magna's ride comfort database.

The results presented in this paper represent a promising first step in demonstrating the potential of the Ride Replay approach. Future work will focus on integrating this tool into Magna's vehicle development process and on the continued refinement of the method, including the introduction of visual support, driver-in-the-loop simulations using IPG CarMaker in combination with Cosin FTire employing reduced-complexity vehicle models, and more comprehensive subjective evaluations with trained drivers.

References

- Bates, D., Mächler, M., Bolker, B., and Walker, S., 2015. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1), pp. 1–48. <https://doi.org/10.18637/jss.v067.i01>.
- Bos, J., Mackinnon, S., and Patterson, A., 2006. Motion Sickness Symptoms in a Ship Motion Simulator: Effects of Inside, Outside and No View. *Aviation, space, and environmental medicine*, 76, pp. 1111–1118.
- Irmak, T., Winkel, K. N. de, Pool, D. M., Bülthoff, H. H., and Happee, R., 2021. Individual motion perception parameters and motion sickness frequency sensitivity in fore-aft motion. *Experimental brain research*, 239(6), pp. 1727–1745. ISSN: 0014-4819. <https://doi.org/10.1007/s00221-021-06093-w>.
- Kemmerling, P. T., 1979. *Dynamic Characteristics of Flight Simulator Motion Systems*. Tech. rep. ADA075241. Defense Technical Information Center (DTIC). Available at: <https://apps.dtic.mil/sti/html/tr/ADA075241/index.html>.
- Lenth, R. V. and Piaskowski, J., 2026. *emmeans: Estimated Marginal Means, aka Least-Squares Means*. R package version 2.0.3. Available at: <https://rvlenth.github.io/emmeans/>.
- Nieuwenhuizen, F., Beykirch, K., Mulder, M., Van Paassen, M. M., Bonten, J., and Bülthoff, H., Aug. 2008. Performance Measurements on the MPI Stewart Platform. In: <https://doi.org/10.2514/6.2008-6531>.
- PLUX Biosignals, 2025. *biosignalsplux Documentation*. Available at: <https://support.pluxbiosignals.com/knowledge-base/biosignalsplux-documentation/> [Accessed Apr. 14, 2026].
- Schwarzhuber, T., Koglbauer, I., and Eichberger, A., 2026. Driving Performance and Preferences for Motion Cueing Algorithms of Amateurs and Professional Racing Drivers. *Automotive Innovation*. ISSN: 2522-8765. <https://doi.org/10.1007/s42154-025-00373-9>.

Engineering the Full Spectrum: The Evolution of VI-grade Driving Simulation

Stefano Bernini, Francesco Ambrogi, Fabio Formaggia

VI-grade: {stefano.bernini, francesco.ambrogi, fabio.formaggia}@vi-grade.com

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

The transition toward electrified, software-defined and increasingly automated vehicles is changing both the technical content of vehicle development and the way engineering decisions must be made. Vehicle behaviour is now shaped by stronger interactions between chassis, powertrain, control software, ADAS functions, human-machine interfaces and NVH performance. In parallel, development programs face pressure to reduce prototype loops and make robust decisions earlier. These challenges highlight the need for simulation environments that evaluate ride and handling, NVH, HMI, and driver-assistance functions in an integrated manner rather than as isolated domains. In this context, the driver becomes central to the development loop, requiring synchronization of visual, auditory, motion, and tactile cues to ensure realistic and reliable feedback. This drives the adoption of multi-attribute, human-centric simulation approaches, where multiple vehicle characteristics are combined within a single environment to support early-stage engineering decisions with greater confidence.

Research question

The work addresses the following question: how can a driving simulator architecture combine low-frequency vehicle motion, mid-frequency tactile vibration and high-frequency acoustic cues in a single, synchronized loop, so that engineers can evaluate coupled vehicle attributes before physical prototypes are available? The working hypothesis is that a full-spectrum simulator, integrating an advanced motion platform with localized vibration actuation and calibrated audio, increases the realism and usefulness of driver-in-the-loop assessment compared with conventional siloed setups. A second hypothesis is that expanding the usable simultaneous motion envelope of a 6-DoF platform improves motion cueing credibility in demanding combined manoeuvres, while preserving the quietness and responsiveness needed for NVH-oriented work.

Methodology

This paper presents the architecture and intended use of a full-spectrum simulation workflow built around three main elements. First, a motion simulator reproduces low-frequency vehicle dynamics related to handling and ride. Second, the HyperDock cockpit adds local excitation at the principal driver touchpoints, namely seat, steering wheel and heel plate, allowing reproduction of higher-frequency tactile content. Third, VI-NVHSim and the audio chain provide acoustic playback and frequency-domain integration. This architecture is configuring as a full-spectrum simulator spanning the frequency content of handling, ride comfort, vibration and acoustics.

This paper describes the evolution of this concept from the Compact FSS introduced in 2023, to the HyperDock cockpit launched in 2024, and to the HexaRev platform introduced in 2025. HyperDock is based on a carbon-fiber cockpit optimized for simulator use, with substantially reduced mass and inertia, a lower center of mass and higher structural stiffness than traditional cockpit-plus-topdisk solutions. These characteristics support more faithful motion transmission and higher-frequency tactile reproduction. The cockpit can also retain active systems used in dynamic simulators, including seat, belt and brake feedback, and can be adapted for production-like ergonomics through seat, pedal and control customization.



HyperDock cockpit on HexaRev Motion Platform

HexaRev serves as the motion base for this full-spectrum environment. Unlike traditional hexapod systems, which can lose available motion range when several axes are combined, HexaRev is designed to preserve a larger usable simultaneous motion envelope. This is especially relevant in scenarios combining yaw, pitch, roll and translational motion, such as transient handling, curb strikes, rough-road events or braking while cornering. The system delivers low mechanical latency, near-silent operation and compatibility with VI-ZeroLatency to improve immediacy and driver immersion.

The system can also support multiple signal-flow configurations. In a standard configuration, wheel-hub accelerations from the real-time vehicle model are sent to VI-NVHSim, where transfer functions reconstruct local accelerations at the driver touchpoints. More advanced configurations can use suspension-link forces, mount loads or multibody model outputs directly, enabling source-oriented studies of bushings, powertrain mounts or other subsystem contributions. Additionally, replay mode for measured or simulated data is supported for objective-to-subjective comparison.

Results and Discussion

The presented Full-Spectrum Simulation environment demonstrates how synchronized motion, tactile vibration and acoustics can be combined within a single Driver-in-the-Loop simulator for multi-attribute vehicle development. This work focuses on the simulator architecture and the engineering capabilities demonstrated through representative application scenarios. The system was tested across several driving conditions, including urban driving on laser-scanned roads with potholes and roundabouts, country-road driving, ride and comfort evaluation on events such as cleats, Belgian blocks and wave profiles, and dynamic handling manoeuvres involving aggressive steering inputs and curb interaction. Across these scenarios, the key benefit is not simply the independent reproduction of motion, vibration and sound, but their synchronized representation as a single driving event. This allows drivers to perceive the combined vehicle response in a way that more closely reflects real-world driving conditions.

From a technical perspective, the combination of HexaRev and HyperDock addresses several long-standing challenges in Driver-in-the-Loop simulation. The enlarged simultaneous motion envelope of the HexaRev platform enables more demanding combined manoeuvres to be reproduced while maintaining the responsiveness required for handling applications. At the same time, its low mechanical noise makes the platform suitable for NVH evaluations, where unwanted simulator noise could otherwise influence the driver's perception.

HyperDock complements the motion platform by providing localized tactile excitation at the seat, steering wheel and heel plate through a lightweight, high-stiffness carbon-fibre cockpit. Together, these characteristics create a simulation environment capable of reproducing vehicle behaviour across multiple frequency ranges within a single synchronized system.

The practical value of this approach lies in supporting multi-attribute vehicle development rather than optimizing individual subsystems in isolation. In real vehicles, drivers experience handling, ride comfort, vibration and acoustics simultaneously, and changes made to one subsystem often influence several aspects of the overall driving experience. Evaluating these interactions within a single Driver-in-the-Loop environment reduces the risk of improving one vehicle attribute while unintentionally degrading another. This is particularly relevant for electrified vehicles, where reduced powertrain noise can expose secondary NVH phenomena, and for software-defined vehicles, where updates to control strategies may influence subjective vehicle behaviour across multiple engineering domains.

Consequently, the presented Full-Spectrum Simulation architecture provides a practical framework for earlier subjective vehicle assessment, allowing engineers to compare a larger number of design variants before physical prototypes become available while maintaining the driver as the central element of the evaluation process.

Conclusions/Implications

This work presents Full-Spectrum Driver-in-the-Loop simulation as a practical approach for earlier and more integrated vehicle development. By combining the HexaRev motion platform, the HyperDock cockpit and the VI-grade NVH toolchain, the system enables synchronized evaluation of vehicle motion, tactile vibration and acoustics within a single simulation environment. The presented architecture supports multi-attribute vehicle development by allowing engineers to assess interacting vehicle characteristics before physical prototypes are available.

References

- Ambrogio, F. and Bernini, S., 2026. HexaRev & HyperDock full-spectrum simulator for multi-attribute development. VI-grade webinar presentation.
- VI-grade, 2026. HexaRev & HyperDock full-spectrum simulator for multi-attribute development. Webinar transcript and presentation material.
- VI-grade, 2025. Full-spectrum simulation development roadmap: Compact FSS, HyperDock and HexaRev. Company presentation material.

An ISO Initiative Towards Standardizing Driving Simulators

Tomoya Chida¹, Martin Fischer², Masaki Fujimoto³, Chantal Himmels⁴, Kazuki Kuwabara³, Hitoshi Kyogoku⁵, Johannes Lindner⁶, Hideyuki Muramatsu⁷, Fabian Ruß⁸, Marcus Schmitz⁹, Martin Sevenich¹⁰, Atsushi Tsubouchi¹¹, Stefan Vorderobermeier¹²

- (1) SUBARU Corporation, Japan
- (2) German Aerospace Center, Germany
- (3) TOYOTA MOTOR CORPORATION, Japan
- (4) BMW AG, Germany
- (5) Nissan Motor Co., Ltd., Japan
- (6) Technical University of Munich, Germany
- (7) S&VL Inc., Japan
- (8) Institut für Kraftfahrzeuge, RWTH Aachen University, Germany
- (9) Würzburger Institut für Verkehrswissenschaften, Germany
- (10) Aumovio SE, Germany
- (11) HONDA R&D Co., Ltd., Japan
- (12) AB Dynamics, Germany

Abstract - Different driving simulators are suited for different use-cases, as cross-simulator studies in the past confirmed. The question arises as to how to appropriately describe what driving simulator was used in a simulator study. A working initiative within the International Organization for Standardization (ISO) currently addresses this issue and is developing a standard to systematically describe driving simulators, or use-case requirements (planned as ISO 26014). Researchers and practitioners will benefit from a suchlike standard, which will introduce a better comparability across scientific studies and will allow a more precise communication between requesters and suppliers.

Keywords: driving simulator, standard, standardization, driver-in-the-loop

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

Early reports of driving simulator usage date back to the 1920s (Wachtel, 1995), which implies that today we are looking back on a hundred years of history. Especially since the 1970s (Allen, Rosenthal, and Cook, 2011), driving simulators have gained attention, and the contexts in which simulators were used broadened. Regardless of whether simulators are used for driver training, driver assessment, human factors research, and/or vehicle development, researchers and practitioners must be aware that different simulators are differently suited for varying applications (Bella, 2008; Himmels, et al., 2024). The aspect of simulator validity, referring to the transferability of results acquired in the simulator to the real world (Blaauw, 1982), must take into account this use-case dependency, and still gains importance in the automotive manufacturers' striving to replace real-world testing by virtual tests. In summary, a driving simulator is not generally required to depict all the cues that would be present in the real world, but only those that are relevant to the targeted use-case (Caird and Horrey, 2011; Diels, Robbins, and Reed,

2011; Kaptein, Theeuwes, and Van Der Horst, 1996). Already today, driving simulators are highly specified to their intended application in some cases, such as in motorsports.

Problem Statement

To fully exploit the potential of driving simulators, we must achieve an understanding which driving simulator specifications are suited to which application—beyond the scientific aspects of study validity and comparability. An attempt to achieve such a *driving simulators to use case* assignment was suggested by Fischer, et al. (2015) at the Driving Simulation Conference 2015, which mainly relied on expert opinions within one research institute.

In recent years, cross-simulator studies have gained attention (Himmels, et al., 2024), which helps researchers to conclude which of their own simulators is suited for what research question or which simulator they require to answer a particular question at hand. However, a common problem is that simulators are described in insufficient detail for valid simulator

study comparisons. Causal is the lack of a consensus on how to adequately describe a driving simulator, and potentially the page limits in scientific papers to describe a simulator in full detail.

A standard outlining a classification framework for driving simulators would foster comparability in scientific literature and support simulator hardware and software suppliers to better describe their portfolio. On the other hand, institutions ordering driving simulators or commissioning driving simulator studies benefit from being able to formulate their requirements more accurately.

The ISO Initiative

In 2022, a working group within the German standardization body DIN was founded, supported by Verband der Automobilindustrie (VDA). The first meeting was held on January 27th 2023 between stakeholders from the German automotive and driving simulator industries. Throughout the first meetings, the working group has agreed on a scope for initiating an ISO standard on the topic of driving simulator standardization: The standard shall apply only to passenger car simulators and only cover driver-in-the-loop simulators. It shall provide a modular description of different elements of the driving simulator, which can be used to systematically classify what a driving simulator offers, or what requirements a use-case comes along with. The assignment of what use-cases a driving simulator is suited for shall remain the responsibility of the user. A new work item proposal concerning the suggested standardization project has been accepted by the ISO in 2025, bringing the work on the ISO draft to an international level. Information can be found under: <https://www.iso.org/standard/92309.html>. The first international meeting, including especially colleagues from Japan beyond the German working group, was held on November 11th 2025.

The Working Draft

The current working draft, which is under discussion among the interested international partners within WG11 'Simulation' suggests a modular description of a driving simulator featuring the following modules: Cockpit, Simulation, Motion, and Visualization (Figure 2).

The cockpit module incorporates all necessary aspects of the input (such as pedals) and ambient devices (such as seat) in the simulator cockpit. A backward channel to the driver exists, since the components in the cockpit may not only receive inputs from the driver, but can also provide inputs (e.g., displays, force feedback devices). The simulation module includes all elements of the ego vehicle and environment simulation. A backward channel to the cockpit exists, since components such as displays require information from the simulation. The motion module includes all hardware and software which is necessary to elicit motion feedback to the driver. The visualization module contains all hardware and software specifications concerning the visual system. Each of the mentioned modules contains numerous submodules, representing important characteristics of the respective module. For each submodule, up to five types are defined, which in some cases are describing different

levels of fidelity (e.g., different limitations in the field-of-view) but in other cases simply different possible configurations (e.g., display type). A common basic category is "none" which means the submodule is not existent or not relevant for the specific use case.

The exact current description of modules, submodules, and categories must not be disclosed in exact detail due to ISO copyright, and is still subject of discussion.

The idea is that the specified modules and submodules are first used to specify the minimum requirements of a specific use case, i.e., a certain type of simulator study. In a second step, different simulators can be categorized according to the given use case classification (Figure 1). A comparison of the resulting classification overviews guides the simulator study designer's decision for a specific simulator and can be furthermore used to contextualize the study results in a standardized manner.

Please refer to Table 1 for an example of how a use case (in this case: dynamic tyre evaluation) is categorized and a description of an example driving simulator that is usually employed for human factors studies. It can be derived from the table where the use case's requirements and the simulator configuration match or mismatch. The standard shall be extended by a coding scheme to easily encode and decode what a driving simulator can offer / what a use case requires without being required to include lengthy tables, especially for scientific publication.

Outlook and Conclusion

The ISO 26014 has the potential to positively influence research with and about driving simulators. However, this potential can only be realized if the developed standard receives widespread acceptance and adoption, for which international collaborators are required. The standard is expected to be publicly available in late 2027.

When discussing the scope of ISO 26014, it was decided that the current standard will apply exclusively to passenger car simulators. Extensions to cover truck simulators, bicycle simulators, motorcycle simulators, or similar may be developed in the future as amendments to ISO 26014.

Acknowledgements

We would like to thank and value all contributors (in addition to this paper's authors) to our standardization attempt on a national and international level:

- Bernhard Schick, Kempten University of Applied Sciences, Germany
- Maximilian Böhle, Kempten University of Applied Sciences, Germany
- Mike Köhler, Kempten University of Applied Sciences, Germany
- Markus Bullinger, BMW AG, Germany
- Arben Parduzi, BMW AG, Germany
- Clemens Löffler, Audi AG, Germany
- Andreas Riener, Technische Hochschule Ingolstadt, Germany

We further thank Julia Karsch and Egbert Fritzsche from VDA for supporting us throughout all our meetings. Our thank also extends to ISO covenor Tak Taeoh from WG11.

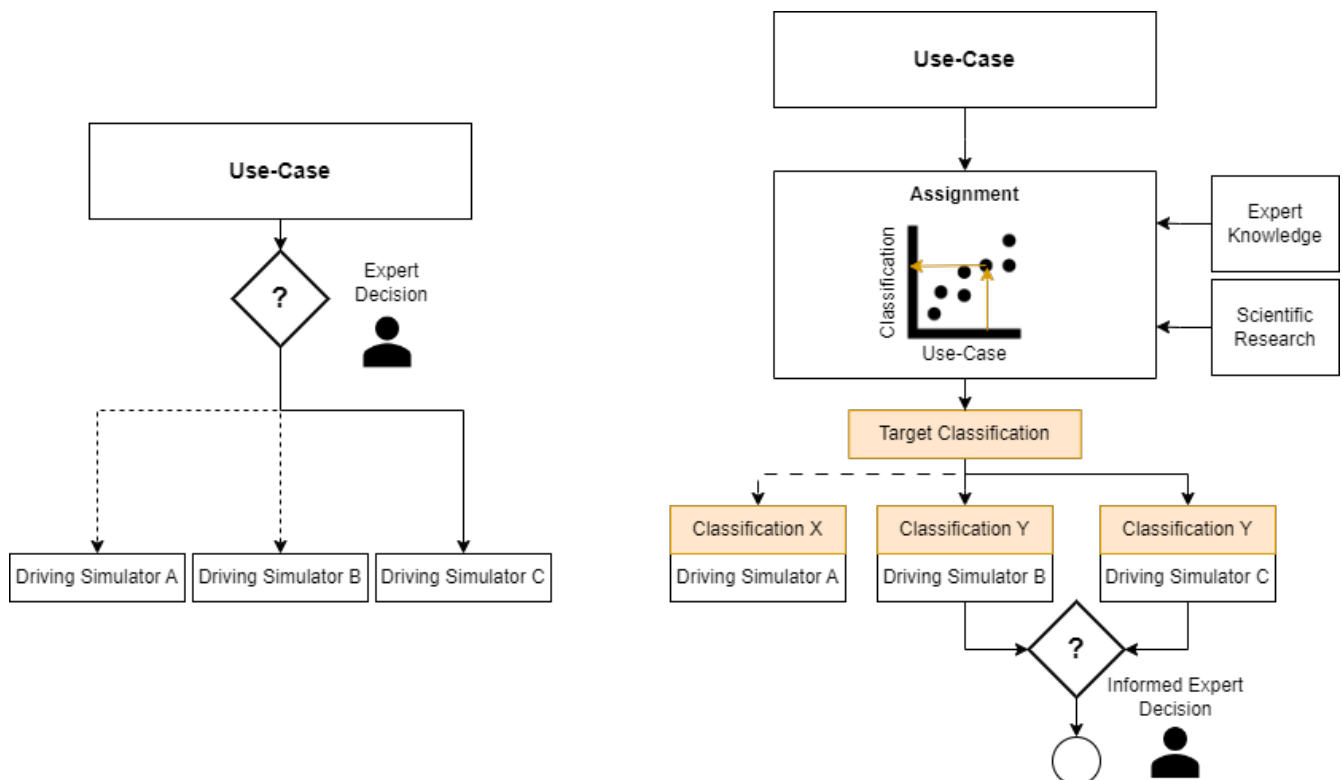


Figure 1: Left: Simulator selection process as is. Right: Simulator selection process supported by ISO 26014.

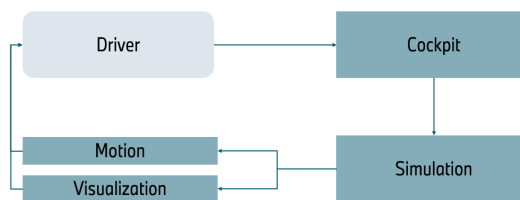


Figure 2: Module organization following the current status of discussions.

References

- Allen, R. W., Rosenthal, T. J., and Cook, M. L., 2011. A short history of driving simulation. In: D. L. Fisher, M. Rizzo, J. Caird, and J. D. Lee eds. *Handbook of Driving Simulation for Engineering, Medicine, and Psychology*. CRC Press. Chap. 1, pp. 1–1–1–34. ISBN: 978-1-4398-2137-1.
- Bella, F., 2008. Driving simulator for speed research on two-lane rural roads. *Accident Analysis & Prevention*, 40(3), pp. 1078–1087. <https://doi.org/https://doi.org/10.1016/j.aap.2007.10.015>.
- Blaauw, G. J., 1982. Driving experience and task demands in simulator and instrumented car: a validation study. *Human Factors*, 24(4), pp. 473–486. <https://doi.org/https://doi.org/10.1177/001872088202400408>.
- Caird, J. K. and Horrey, W. J., 2011. Twelve practical and useful questions about driving simulation. *Handbook of Driving Simulation for Engineering, Medicine, and Psychology*, 2.
- Diels, C., Robbins, R., and Reed, N., 2011. Behavioural validation of the TRL driving simulator DigiCar: phase 1: speed choice. In: *International Conference on Driver Behaviour and Training*. Vol. 5.
- Fischer, M., Labusch, A., Bellmann, T., and Seehof, C., 2015. A task-oriented catalogue of criteria for driving simulator evaluation. In: *Proceedings of the Driving Simulation Conference 2015*, pp. 139–150. Available at: <https://elib.dlr.de/99561/>.
- Himmels, C., Venrooij, J., Parduzi, A., Peller, M., and Riener, A., 2024. The Bigger the Better? Investigating the Effects of Driving Simulator Fidelity on Driving Behavior and Perception. *Transportation Research Part F: Traffic Psychology and Behaviour*, 101, pp. 250–266. ISSN: 1369-8478. <https://doi.org/https://doi.org/10.1016/j.trf.2024.01.007>.
- Kaptein, N. A., Theeuwes, J., and Van Der Horst, R., 1996. Driving simulator validity: Some considerations. *Transportation research record*, 1550(1), pp. 30–36. <https://doi.org/https://doi.org/10.1177/0361198196155000105>.
- Wachtel, J., 1995. Brief history of driving simulators. *Tr News*, (179).

Module	Submodule	Use Case Requirements Classification	Driving Simulator Classification
Cockpit	Mockup	min. B: Representative - Half vehicle	B: Representative - Half vehicle
Cockpit	Accelerator pedal	min. B: Closed loop (model feedback)	B: Closed loop (model feedback)
Cockpit	Brake pedal	min. B.1: Active, representative	B.1: Active, representative
Cockpit	Steering controller	min. B.1: Active, representative	A.1: Passive, representative
Cockpit	Control devices	None	None
Cockpit	Gear shift lever	None	None
Cockpit	Indicator	None	B: Representative
Cockpit	Seat	min. B: Representative, adjustable position	B: Representative
Cockpit	Seat belt	min. A: Three-point belt - passive	A: Three-point belt - passive
Cockpit	AC Cooling	None	A: Non-adjustable cooling
Cockpit	AC Heating	None	A: Non-adjustable heating
Cockpit	Air flow	B: Adjustable for the driver	A: Non-adjustable air flow
Cockpit	Driver Information System	min. A: Generic	B: Representative
Cockpit	Ambient lighting	None	None
Cockpit	Haptic feedback (as part of infotainment)	None	A: Generic - Steering wheel
Cockpit	Interior sensors	None	A: Generic
Cockpit	Environment sound	None	B: Other traffic participants + non-traffic related environment sounds - generic (e.g., generic city)
Cockpit	Vehicle sound	min. B: Representative	A: Generic
Cockpit	Information sounds	None	A: Generic
Cockpit	Entertainment sounds	None	A: Generic
Cockpit	Speaker hardware	min. A: Headset (2.0 system), 2D	B: Speakers - 4.1 system (four speakers, one subwoofer)
Simulation	Ego vehicle	min. C.1: Multi-body model with flexible body	A: Single-Track Model
Simulation	Tyres	min.: B.1: Empirical, lookup-table	A: General
Simulation	Vehicle Dynamics Control Systems	min.: B: Software-in-the-loop	None
Simulation	Sensors	None	None
Simulation	Automated driving systems	None	B: Software-in-the-loop
Simulation	Actor behavior	None	D: Complex actor movements and interactions
Simulation	Environmental conditions	None	D.2: Changing environmental conditions influence visual AND vehicle system, with sound
Simulation	Road surface model	min.: A: Generic	A: Generic
Motion	Degrees of freedom (DoF)	C.1: Hexapod + linear Y-system	B: 6 DOF hexapod
Motion	Acceleration forces in relevant axis	C: Acceleration suitable for high dynamic driving ($> 4m/s^2$)	A: Indication of acceleration ($< 1m/s^2$)
otionMt	Movement space in relevant axis	C: 1:1 manoeuvre	C: 1:1 manoeuvre
Motion	Latency	C: $< 50 ms$	A: $> 100 ms$
Motion	Frequency	B: Secondary Ride (up to 30 Hz)	A: Primary ride (up to 5 Hz)
Motion	Vibration actuator (hardware)	None	B: Multiple Location - high and low frequent
Motion	Motion Cueing: Adaptivity	B: Adaptive, scenario-specific	A: Minor scenario-specific adaptations
Motion	Motion Cueing: Operating principle	C: Direct feedthrough (except in X)	A: Filter-based (adaptive washout)
Motion	Motion Cueing: Interaction type	C: Online Cueing	A: Offline cueing
Visualization	Frame rate	C: ≤ 240 Hz (120 Hz sufficient)	A: ≤ 60 Hz
Visualization	Display type	B: Projection OR C: LED wall (subcode: any)	B.2: Projection, cylindrical OR D: Head-mounted display
Visualization	Horizontal field of view	min. B: $> 180^\circ$ & $< 360^\circ$	B: $> 180^\circ$ & $< 360^\circ$
Visualization	Vertical vision coverage	min. B: 1	B: 1
Visualization	Resolution	min. C: ≥ 100 & < 200 ppd	A: < 50 ppd
Visualization	Contrast	min. A: $< 5000:1$	A: $< 5000:1$
Visualization	Monoscopic vs. stereoscopic	A: Monoscopic	A: Monoscopic
Visualization	Mirror representation	A: Extra displays for rear view mirror and side mirrors	A: Extra displays for rear view mirror and side mirrors
Visualization	Anti-aliasing	min. A: Fast Approximate (FXAA)	C: Super-Sampling (MSAA)
Visualization	Lighting	min. A.1: Simple directional lights with shadows	B: Multiple light sources, e.g., sun, street lamps, headlights with shadows
Visualization	Material/texture	min. A: Materials with only base colour/texture	B: Materials with base colour texture and normal map without specular map

Table 1: Classification for an example simulator used in context tyre evaluations, organized by Module categories: Cockpit, Simulation, Motion, and Visualization. *min* denotes minimum requirements, which may be met by superior types other than the minimum specification, too.

ADAS

Scenario-Based Approach in System Engineering: Toward an Assessment of Coverage

Christophe Bohn¹, Erwan Revert¹, Faouzi Adjed¹

(1) Institut de Recherche Technologique SystemX, Palaiseau, e-mail: christophe.bohn@irt-systemx.fr, erwan.revert@irt-systemx.fr, faouzi.adjed@irt-systemx.fr

Abstract – This paper presents a structured methodology for the design, validation, and evaluation of automated driving systems built upon the scenario-based approach promoted by the UNECE through the NATM framework. The proposed method combines systems engineering process with a scenario-based approach to ensure assessable coverage of the system's operational domain: an engineering reference framework is introduced to formalize both system requirements and the associated scenario space for backing coverage assessment. The approach relies on the systematic partitioning of scenarios into Use Cases, enabling traceability between requirements, design choices, and validation activities. This process supports concepts aligned with standards such as ISO 3450x and ISO 21448. To operationalize this approach, a prototype toolchain developed within IRT SystemX is presented, enabling scenario management, requirement formalization, and test plan specification.

Keywords: Driving scenarios, Dynamic environment modeling, Conception, Validation, Traceability, Automated vehicles

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

Automated road vehicles operate in open and dynamic environments, which presents a significant challenge: their design and validation must ensure comprehensive coverage of the wide range of situations they may encounter.

The informal working group (IWG) VMAD (Validation Methods for Automated Driving) of UNECE (United Nations Economic Commission for Europe) has defined the methodological guide NATM (New Assessment/Test Method for Automated Driving), which introduces a scenario-based approach within a regulatory framework to validate automated transport systems (ESC, 2023). Numerous papers and standards have focused on clarifying the formalism for scenario description. Others have aimed to establish the link between scenarios and simulation.

In this article, we aim for a broader approach that repositions scenarios within the engineering process for ensuring a comprehensive and assessable coverage. We detail how the scenario-based driving approach complements the classical systems engineering approach and how this method can be supported with tools to make it efficient.

This work on methods and tools follows from research carried out over the past 10 years within collaborative research projects at IRT SystemX with its industrial partners.

Product solution

The proposed solution is a structured method for designing and validating automated driving systems, combining classical systems engineering methods with the scenario-based approach to ensure a comprehensive and assessable coverage. To achieve this, the method complements classical systems engineering processes with the concept of scenarios and with the use of reference scenario libraries, *i.e.*, expert-built databases.

To support this process, the solution is implemented through a prototype toolchain using dedicated tools and standard frameworks. These tools help organize scenarios and use-cases, maintain links with system requirements, and ensure consistency throughout the development lifecycle.

Methodology

Use of driving scenarios in design

Creation of an engineering reference framework

The first phase of this method consists in creating what is referred as an *engineering reference framework*. It includes both a formalization of the needs that the system must address, and a set of documented observed or engineered driving

situations that the system may encounter. This set constitutes the scenario space, encompassing all the situations the system will face and to which it must be able to respond to. This space serves as the reference scenario space for evaluating coverage.

The concepts related to the scenario-based approach such as the scenario typology (functional, abstract, logical and concrete scenarios), use-cases concepts, and definition of TOD (Target Operational Domain) and ODD (Operational Design Domain) and their structure are aligned with the ISO 3450x serie.

Scenario partitioning into Use Cases

For each logical scenario of the scenario space, the designer must define the different possible system responses, specified as Use Cases. A distinct Use Case is identified for each expected behavior. These Use Cases are defined at a logical level and inherit all the characteristics of the corresponding logical scenario in terms of actors, parameters, and storyboard. For each parameter, it is necessary to determine how the value range of the initial scenario is partitioned across the different Use Cases. Beyond identifying the different responses to be specified, the main challenge at this stage lies in precisely defining the boundaries between Use Cases, especially when these depend on the coupling between multiple parameters. It is essential that the Use Case spaces are contiguous, non-overlapping, and collectively cover the entire logical scenario to which they relate.

Concept selection and coverage of the scenario space

We have seen how a particular scenario can be addressed using a finite set of Use Cases corresponding to each expected behavior. During design, this work must be carried out not only for all logical scenarios in the engineering reference framework, but also with the objective of defining generic concept choices capable of addressing multiple scenarios. Therefore, the definition of all Use Cases is an iterative process carried out in parallel with the definition of concept choices and their specification in the form of requirements.

As the design progresses, concepts evolve and therefore modify the Use Cases relative to the initial partition, in terms of expected behavior or parameter value ranges to be considered. The scenario and Use Case-based approach, supported by appropriate tools, ensures coverage of the scenario space throughout this iterative process. The scenario and Use Case management tool must be able to identify gaps in parameter value ranges or, conversely, overlaps.

System robustness relies on the ability of concept choices to cover a wide range of situations. The use

of localized solutions to handle specific cases should be minimized to avoid over-specializing the implementation. The scenario-based approach and the early definition of Use Cases are key enablers for traceability, robustness, and achieving the necessary level of system simplicity.

At the end of the design phase, the scenario engineering reference framework is enriched with all Use Cases that trace the system's expected responses across the entire space to be covered. Throughout the iterative process, the designer ensures consistency between these Use Cases and the requirements that enable system implementation.

Use of driving scenarios for Verification/Validation, evaluation, and SOTIF

Driving scenarios in Verification and Validation activities

The activities at this stage aim to verify and validate that the system implementation complies with its specification and ultimately with the intended need. For a system such as an automated vehicle operating in an open environment, taking driving scenarios into account is necessary to verify certain requirements that need to be assessed within their operational context, even at the unit test level. This may include, for example, unit tests on environmental perception sensors.

When defining the Verification and Validation (V&V) plan, the means required to test each requirement are identified. For each test, success criteria are defined. For functional requirements that need to be tested in context, Use Cases are used directly, as they include not only the scenario but also the expected behavior. As in a classical V&V plan, the strategy determines which Use Cases are the most relevant for testing (worst case, sampling, representative cases, etc.).

Once this mapping between requirements and Use Cases has been established in the test plan, test cases and test procedures can be defined in a conventional manner. Tests can be performed in simulated, controlled, or real environments.

Use of driving scenarios for system evaluation

While V&V considers the system as a white box to verify each requirement, evaluation treats the system as a black box and follows a performance-based approach. For such an approach, it is necessary to define an evaluation objective consistent with customer needs and system usage. Since the system is viewed as a black box during

evaluation, its behavior cannot be assumed; therefore, contextualization must rely on scenarios rather than Use Cases. The definition of success criteria, depending on the evaluation objective, must be established either globally for the entire evaluation or contextually for each identified scenario.

Evaluation plan makes it possible to identify all relevant scenarios needed to qualify the system. Since the scenario engineering reference framework is assumed to cover all situations the system may encounter, the selected scenarios are drawn from this framework.

The evaluation success criteria enable the transition from scenarios to evaluation Use Cases. Variations in success criteria depending on parameter values lead to the definition of different Use Cases.

The final step is to move from Use Cases to test cases and test procedures. The consolidation of actually tested situations can also be performed in the same way.

Use of driving scenarios in SOTIF evaluation

Regarding SOTIF (ISO 21448, 2022), the first step of the implementation consists of an analytical approach to hazard identification and risk assessment (clause 6). This is followed by the study of functional insufficiencies and triggering conditions, and their impact on the system (clause 7). At the end of this analytical phase, the analysis of the specifications makes it possible to determine whether the system is robust with respect to SOTIF. If it is not robust, an iteration of the specification must be carried out (clause 8). If it is robust, the scenario-based SOTIF evaluation phase can be initiated.

The evaluation phase aims to verify the system's behavior for known scenarios and to explore its behavior for unknown scenarios at the boundaries of known ones. The process used is similar to what has been described for previous steps. This is considered evaluation in the sense that the objective is to address SOTIF goals rather than to verify or validate the system as a whole. The definition of the strategy, as set out in clause 9, makes it possible to define the evaluation objectives and their implementation. However, the system behavior is assumed to be known, as the goal is to test its robustness against uncertainties (functional insufficiencies and triggering conditions). The evaluation can therefore build on relevant Use Cases that already incorporate the expected behavior and thus the test success criteria. For the evaluation of unknown scenarios, the standard specifies that these scenarios should be assessed at the boundaries of known scenarios. It is therefore

possible to build on the same Use Cases by extending certain parameter value ranges in accordance with the strategy defined in clause 9.

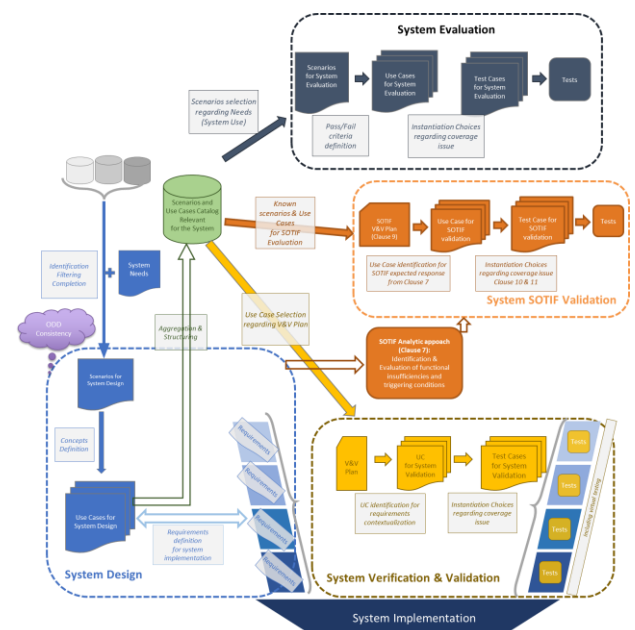


Figure 1: Diagram of the complete process of the scenario-based approach presented at the Lambda-Mu 24 conference (Julien et al., 2024)

Fig. 1 illustrates all the steps described in this article, starting with the reference catalog, followed by system design and evaluation, then SOTIF validation, and finally system verification and validation.

Solution implementation

This method is being supported by dedicated tooling and implemented as a prototype in the context of the CVH project (*Chaîne outillée pour la Validation et l'Homologation*). The prototyped toolchain is based on the MOSAR platform concepts developed by IRT SystemX. It is structured around four main components that enable the instantiation of the objects and traceability required for the workflows described above.

Scenario management

The first required component is a scenario manager. Its primary role is to store reference scenario libraries and ensure connectivity with existing scenario databases. For example, this tool must be able to reference scenarios from external sources, such as the scenario dataspace part of the European SYNERGIES project.

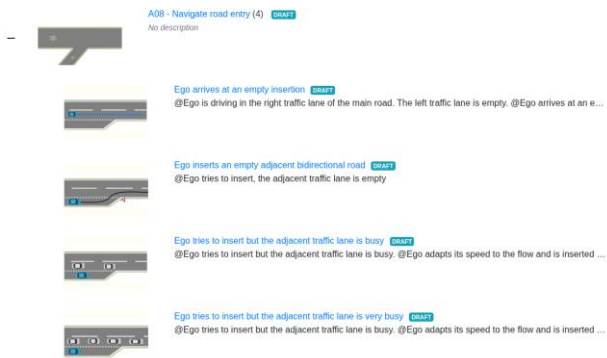


Figure 2: Samples from the subset of the scenario space that constitutes the engineering framework for the TJC in MOSAR Scenario Manager

System engineering framework documentation

The second component must enable the formalization of system requirements in a format that can be leveraged by automation tools. To this end, a dedicated component has been developed within the MOSAR System Proof tool. In particular, it allows the documentation of the system’s TOD (Target Operational Domain) and ODD (Operational Design Domain) using a taxonomy compatible with scenario description standards, easing domains and scenarios space comparison. Furthermore, this specification is interoperable with third-party tools through the use of the ASAM OpenODD format.

Dynamic Elements	
Traffic Agents	
Attribute	Requirements
Traffic Agents: Type	INCLUDED required target moving vehicle
	Presence of target vehicle in front
	INCLUDED Presence of moving road motor vehicle
Traffic Agents: Speed	INCLUDED is up to 60kph
Traffic Agents: Special Vehicles	EXCLUDED Presence of special vehicles
	Ambulance, Police, other vehicles equipped with visual and sound sirens

Figure 3: Illustration of the Target Operational Domain definition of a system in MOSAR System Proof

Use case design

A dedicated component has also been developed to design and manage use cases. It enables the addition to a scenario (which captures a specific driving situation) of the elements defining the use case: primarily the restriction of situations to those relevant for the system, and the specification of the expected behavior. Use cases also supports the addition of elements that facilitate the conversion of this asset into a simulable entities.

Related scenario			
M1.1.1 CIPV Brakes Simple - Annotated Good			
This Elementary scenario is characterized by : Situation: - The EGO vehicle & the V1 (CIPV) drive on the main highway road (speed > 5 Km/h) Events & Actions: - V1 (CIPV) brakes for a reason unrelated to the infrastructure			
Initial Scene	V1 Brakes	V1 End Of Braking	Final Scene
start_trigger	brake_pos	cutin_end_position	final_timing
Input parameters			
Name	Description	Value	Nominal value
ego_lane	Qui liber cum et mortem contemnit	Lane 0	Lane +1
target_deceleration	Praeterea bona praeterita grata recordatione renovata delectant.	< 0 m/s²	34 m/s²
ego_initial_speed	Qui liber cum et mortem contemnit	> 5 km/h	12 km/h
cutin_start_position	Praeterea bona praeterita grata recordatione renovata delectant.	[0 ; 200] m	55 m
cutin_end_position	Qui liber cum et mortem contemnit	> 0 m	34 m

Figure 4: Sample use case specifying system behavior over a scenario in MOSAR System Proof

Assessment campaign and test protocol specification

Finally, the last component in this chain is a tool that enables the creation of a system assessment campaign. This campaign includes information on the objectives of the tests performed and on the strategy for their implementation, particularly in terms of coverage. It establishes the link between the system engineering reference framework and the tests carried out in simulation, on test tracks, or in real-world conditions.

Test scenarios			
Cut-In v1-SNAPSHOT			
Cut-In			
Test description		Expected	
For Cut in scenario: The lateral wandering distance the vehicle will normally wander within the lane is 0.375m. The perceived boundary for cut-in occurs when the vehicle exceeds the normal lateral wandering distance (possibly prior to actual lane change) The distance a _i is the perception distance based on the perception time [a _i]		Final_relative_distance	≥ 0 m
		Ego_speed_final	≥ 60 km/h
		Cutter_speed_final	> 0 km/h
Parameters			
Name	Description	Suitable range	Tolerance
Cutter_max_lateral_acceleration	Maximum lateral acceleration of the cutter when it performs the lane change.	Not defined	± 0.1 m/s²
ITC	Time To Collision between the vehicle under tests and the cutter when it performs the lane change.	2 s	± 0.5 s
a _i	Perception time of the danger by the V1 IT	0.72 m	± 0.1 m

Figure 5: Example of a test protocol based on a use case defined in the MOSAR System Proof

Results and discussion

This toolchain, which lays the foundations for a systems engineering approach combining a scenario-based methodology, is currently being used for application to a TJC-type system (Traffic Jam Chauffeur) as a Proof-of-Concept application. This study notably makes it possible to validate the formalism of scenarios, use cases, as well as domain specifications such as ODD or TOD for this type of

application. Indeed, the automations leveraging these formats are operational and enable interoperability between tools from different sources while ensuring end-to-end traceability.

The coverage of situations that are part of the engineering framework can be demonstrated by design, with an end-to-end traceability between system requirements, the scenario space, resulting use-cases and tests results.

However, research activities must continue to integrate reliable and proven metrics that will provide proofs about the coverage of the system's operational domain by the reference engineering framework.

Conclusion

The proposed method and its associated tools make it possible to leverage the knowledge provided by existing scenario databases while ensuring coverage of the combinatorial space. This coverage is achieved by applying the approach as closely as possible to the needs of a specific system. By combining the scenario-based approach with systems engineering, we promote its adoption in industrial development cycles: the tooling enables the traceability required for quality assurance and supports scalability.

References

- Brini, M., Arnoux, E., Foyer, B., Bresson, G., Durville, L., Gaillard, F., & Pajon, M. (2020). MOOVE & MOSAR Projects : A scenario library for designing & validating ADS.
- Chang, Y., Xi, C., & Luo, Z. (2025). Specific Scenario Generation Method for Trustworthiness Testing of Autonomous Vehicles Based on Interaction Coding. *Applied Sciences*, 15(19), 10656. <https://doi.org/10.3390/app151910656>
- DGITM/DMR/TUD. (2024). Démonstration de sécurité des transports routiers automatisés Approche par scenarios Quel apport des listes de scénarios fonctionnels ?
- ESC, E. and S. C. (2023). New Assessment/Test Method for Automated Driving (NATM) Guidelines for Validating Automated Driving System (ADS). <https://unece.org/sites/default/files/2023-04/ECE-TRANS-WP.29-2023-44e.pdf>
- Filière Automobile & Mobilité, P. (2019, novembre 7). DES SCÉNARIOS POUR LA CONCEPTION, LA VALIDATION ET L'HOMOLOGATION DES VÉHICULES AUTONOMES DE NIVEAU 3 ET PLUS.
- Gutiérrez-Moreno, R., Barea, R., López-Guillén, E., Arango, F., Sánchez-García, F., & Bergasa, L. M. (2024). Enhancing Autonomous Driving in Urban Scenarios : A Hybrid Approach with Reinforcement Learning and Classical Control. *Sensors*, 25(1), 117. <https://doi.org/10.3390/s25010117>
- ISO 21448. (2022). Véhicules routiers—Sécurité de la fonction attendue (1re éd.). <https://www.iso.org/fr/standard/77490.html>
- ISO 34501. (2022). Véhicules routiers—Scénarios d'essai pour les systèmes de conduite automatisée—Vocabulaire (1re éd.). <https://www.iso.org/fr/standard/78950.html>
- ISO 34503. (2023). Véhicules routiers—Scénarios d'essai pour les systèmes de conduite automatisée—Spécification du domaine de conception opérationnelle (1re éd.). <https://www.iso.org/fr/standard/78952.html>
- ISO 34504. (2024). Véhicules routiers—Scénarios d'essai pour les systèmes de conduite automatisée—Catégorisation des scénarios (1re éd.).
- Julien, N., Vincent, H., & Bohn, C. (2024). CVH -Chaîne outillée pour l'Homologation de Véhicules automatisés.
- Revert, E. (2023, juin). MOSAR scenario manager and framework insights. ADAS & Autonomous Vehicle Technology Conference. <https://hal.science/hal-04372324>
- Revert, E., Dulewicz, L., Reisinger, K., Daskalaki, E., Bolovinou, A., Rial Martínez, M., Alirezaei, M., van Hassel, E., Gkentsidis, K., Griffon, T., Di Piazza, S., Irvine, P., Jeyachandran, J., & Zhang (Jason), X. (2024). D2.2 Storylines definition and technical and interoperability requirements. European Project.
- Riedmaier, S., Ponn, T., Ludwig, D., Schick, B., & Diermeyer, F. (2020). Survey on Scenario-Based Safety Assessment of Automated Vehicles. *IEEE Access*, 8, 87456-87477. <https://doi.org/10.1109/ACCESS.2020.2993730>
- Scholtes, M., Westhofen, L., Turner, L. R., Lotto, K., Schuldes, M., Weber, H., Wagener, N., Neurohr, C., Bollmann, M. H., Kortke, F., Hiller, J., Hoss, M., Bock, J., & Eckstein, L. (2021). 6-Layer Model for a Structured Description and Categorization of Urban Traffic and Environment. *IEEE Access*, 9, 59131-59147. <https://doi.org/10.1109/ACCESS.2021.3072739>
- Schuldt, Fa. (2017). Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen.

ASAM QSQ: Quantifying Simulation Quality for Credible Virtual Validation

ASAM e.V.

ASAM e.V., Germany, e-mail: diego.sanchez@asam.net

Abstract - Simulation is now a key part of the development and validation of advanced driver assistance and automated driving systems. As virtual methods move closer to engineering decisions and future regulatory acceptance, the quality of simulation frameworks and models must be described in a transparent and reproducible way. Today, this is still difficult because metrics, assumptions, and reporting styles differ across domains and organizations. ASAM QSQ addresses this gap through an ongoing standardization effort that aims to structure how simulation quality is described, documented, and exchanged. The current state already provides a clear direction. It links test objectives, validation methods, and metrics, and it aims to represent this information in a machine-readable form that supports automation and traceability. The target state is a standard that enables consistent, comparable, and reusable quality descriptions across virtual validation workflows.

Initial results across sensor simulation, vehicle dynamics, and system-level use cases show that existing validation practices can be organized into a coherent framework. They also show that simulation quality is not a single score. It is a contextual description of evidence, assumptions, and fitness for purpose.

Keywords: Simulation quality, virtual validation, metrics, credibility, ASAM

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

Simulation enables scalable validation, but its credibility depends on how well its quality is described. Today, this description is often inconsistent. Different domains use different metrics, assumptions are frequently implicit, and results are difficult to compare. This makes it harder to communicate quality in a clear and reusable way.

ASAM QSQ addresses this problem. The current state is a growing body of practice and domain knowledge, but without a common structure for how quality is described. The target state is a standard that makes quality explicit, structured, and exchangeable.

The value of this effort is practical. Tool vendors need a way to clearly describe the capabilities, while tool users need a way to express requirements consistently. Authorities and assessors need a way to understand which evidence supports a given virtual validation claim. QSQ aims to provide this common language.

Current state and target direction

At present, simulation quality is usually described in isolated ways. Metrics may exist, but they are not always linked to a clear purpose. Validation evidence may exist, but it is often difficult to exchange or compare. In some areas, strong domain-specific procedures already exist, while in other areas the description remains informal.

The direction forward is therefore not to replace existing validation practice, but to structure it. QSQ moves from scattered descriptions towards a framework that connects purpose, method, and evidence.

The current state can be summarized in three points. First, quality descriptions are fragmented. Second, existing metrics are not always tied to the intended use. Third, validation evidence is difficult to reuse beyond its original context.

The target state aims for three corresponding improvements. First, every quality statement should be linked to a specific objective. Second, every metric should be interpreted in context. Third, quality information should be exchangeable in a form that supports both human understanding and automated processing.

Methodological core

A central structure organizes the QSQ approach:

- Why: the test objective
- How: the validation method
- What: the metric or KPI

This structure ensures that each metric has a defined role. It prevents the use of numbers without interpretation and supports consistent communication across domains.

The same logic also supports a generic validation workflow:

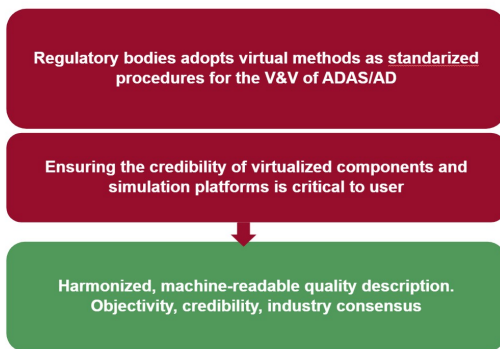


Figure 1: From fragmented quality descriptions to a structured and exchangeable representation.

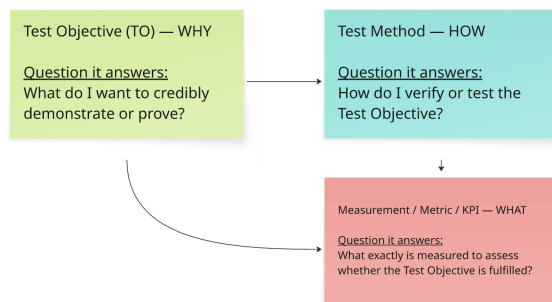


Figure 2: Core structure of QSQ linking objective, method, and metric.

1. Define the requirement or objective
2. Define the reference or measurement setup
3. Recreate the setup in simulation
4. Extract measurable outputs
5. Compare results using appropriate metrics
6. Interpret the outcome using defined criteria

This workflow is useful because it is simple, transparent and already compatible with established engineering practice. It does not impose one universal method on every domain. Instead, it provides a stable structure within which different domains can describe their own evidence consistently.

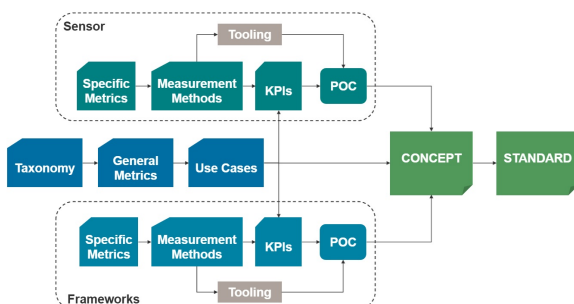


Figure 3: Generic validation workflow supported by the QSQ structure.

Application across domains

The strength of QSQ is that the same structure can be applied across different simulation domains while still respecting their differences.

In sensor simulation, existing laboratory procedures provide a strong starting point. Real measurements can be reproduced in simulation and outputs can be compared using defined metrics. This shows that QSQ does not require a completely new validation methodology. Instead, it organizes established methods into a clearer and more exchangeable form.

For camera simulation, the required model fidelity depends strongly on the use case. Some applications can work with simplified image generation, while others require a more detailed representation of optics and sensor behavior. This confirms that quality cannot be discussed independently of purpose.

In LiDAR and radar simulation, the focus is often on whether simulated sensor behavior aligns with measured behavior under defined conditions. Here, the important contribution of QSQ is not to define a new sensor metric, but to structure how the objective, method, assumptions, and result are documented.

In vehicle dynamics, quality relates to how closely simulated behavior reflects physical behavior. This can refer to motion quantities such as position, velocity, and acceleration, or to deeper model behavior depending on the intended use. Again, the main point is that the meaning of quality depends on what the simulation is expected to support.

At system-level, simulation quality becomes even more contextual. A full chain involving environment, sensors, perception, planning, and control cannot be described by a single universal number. Instead, the quality statement becomes a structured argument built from multiple pieces of evidence.

Example: LiDAR target detection validation

A simple end-to-end example can illustrate how a QSQ quality statement may be instantiated, exchanged and consumed in an industrial toolchain.

The intended use is the validation of LiDAR target detection performance for a predefined set of test scenarios. Based on this, the relevant LiDAR effects, object properties, and environmental conditions are assessed. This assessment is then used to design a measurement campaign with the real LiDAR sensor and sensors for the reference data of all involved objects and environmental conditions, the so-called Ground Truth.

In this example, the selected scenario addresses the base performance of the sensor under good environmental conditions. A Lambertian calibration target is placed at a defined distance and pose relative to the sensor. The real measurement setup includes the LiDAR sensor to be simulated and reference information such as target geometry, material or BRDF properties, sensor pose and target pose.

The same scenario is then recreated in simulation using the available reference data as ASAM OpenX artifacts. The measured and simulated LiDAR outputs are compared using selected metrics.

The resulting QSQ report links the test objective, validation method, reference data, assumptions and metrics. It can be exported as a machine-readable package and consumed by a CI validation dashboard or evidence-management tool.

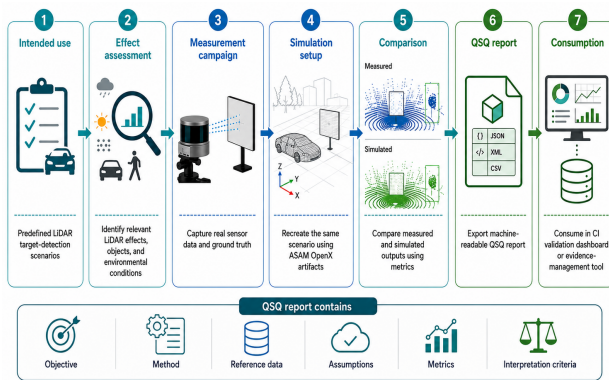


Figure 4: End-to-end example LiDAR target detection validation.

Key observations

The current understanding of QSQ can be summarized in four observations.

First, simulation quality is multi dimensional. It cannot be reduced to one universal value without losing essential meaning. Different aspects such as accuracy, robustness, completeness, and suitability may matter in different ways depending on the application.

Second, simulation quality depends on use case. A model that is sufficient for one task may be insufficient for another. This is true for sensor fidelity, real-time behavior, determinism, and many other properties.

Third, simulation quality is evidence-based. Quality statements must refer to some form of reference, measurement, comparison, or well justified assumption. Without this link, quality remains a vague claim.

Fourth, simulation quality must be exchangeable. This is where QSQ creates its main value. The goal is not only to understand quality internally, but to describe it in a way that others can interpret, compare, and process.

Implications for QSQ

These observations lead to a practical implication: a QSQ description should contain a small but clear set of elements.

It should state the intended use. It should explain the validation method. It should define the relevant metrics. It should document the reference data or reference principle. It should state the interpretation criteria. It should also make the scope and assumptions explicit.

This structure improves transparency in two ways. First, it makes the basis of a quality statement visible. Second, it supports automated checking and traceability when represented in a machine readable form.

That is the direction from where we are today to where QSQ aims to go. Today, quality information is often implicit and difficult to exchange. The target is a reusable description that supports documentation, communication, and automated processing.

Discussion

A useful aspect of the current QSQ status is that value is already visible before the standard is finalized. The efforts of the expert members of the project already help align terminology and structure across domains. It makes quality discussions more explicit and reveals where assumptions, limitations, and evidence still need clarification.

Another important point is that the work builds on existing standards and practices. This is essential because it keeps QSQ grounded. It is not trying to replace mature domain procedures. It is creating a common layer that helps express them in a more consistent way.

At the same time, several difficult questions remain open. How should evidence from subsystems be combined at system level? How should unknown scenarios be addressed? How should realism and scalability be balanced? These questions are not signs of weakness. They are central questions in credible virtual validation, and QSQ is valuable precisely because it creates a structure in which such questions can be expressed more clearly.

Conclusion

ASAM QSQ addresses a practical need in virtual validation: the lack of a structured, comparable, and exchangeable description of simulation quality.

The current QSQ work already shows a clear direction. Quality can be described through a stable relationship between objective, method, and metric. This structure can be applied across domains while still respecting their differences.

The target state is equally clear. QSQ aims to turn fragmented quality descriptions into transparent and reusable quality statements that support communication, comparison, and automation.

The central message is simple. Simulation quality is not a single number. It is a structured description of evidence and fitness for purpose.

References

- ASAM e.V., 2025. *ASAM QSQ Quantifying Simulation Quality Project Proposal P202504*. ASAM OpenX.
- Braun, A., 2023. Automotive mass production of camera systems Linking image quality to AI performance. *tm Technisches Messen*, 90(3), pp. 205–218. <https://doi.org/10.1515/teme-2022-0029>.
- DIN, 2025. *DIN SPEC 91527:2025-12*. Goals, methods and metrics for automated and semi automated runtime monitoring of AI systems for non adversarial performance degradations.
- DIN SAE, 2023. *DIN SAE SPEC 91471*. Performance assessment for LiDAR sensors.
- European Commission, 2022. *Commission Implementing Regulation (EU) 2022/1426 Annex III Part 4 Principles for credibility assessment for using virtual toolchain in ADS validation*.
- IEEE, 2025. *IEEE Standard for Automotive System Image Quality P2020*.
- ISO, 2022. *ISO 19206-3*. Road vehicles target related test specification part 3.
- MD Engineering Services, 2021. *SimCert initiative Rating Simulation Solutions*.
- Mueller, P., Lehmann, M., and Braun, A., 2020. Simulating tests to test simulation. In: *IS and T International Symposium on Electronic Imaging Autonomous Vehicles and Machines*, pp. 149–149–8. <https://doi.org/10.2352/ISSN.2470-1173.2020.16.AVM-149>.

From Concept to Validation: Mastering Scene and Scenario Fidelity with the RoadRunner Product Family

Simone Hämmerle¹, Dr. Philipp Hafemann¹

(1) MathWorks, Weißenstephaner Str. 6, 81673 Munich, Germany, e-mail: {shaemmer, phafeman}@mathworks.com

Abstract – This paper presents a workflow for creating, managing, and reusing simulation content for ADAS and automated driving across different levels of scene and scenario fidelity. The approach integrates established interactive authoring with newer programmatic and automated methods. A central element is the reconstruction of simulation content from recorded multi-sensor vehicle data, enabling realistic representations of road infrastructure, traffic participants, and dynamic driving behavior.

The workflow supports seamless transitions between low- and high-fidelity representations and allows content to be refined, parameterized, and reused across different validation tasks. To enable integration into industrial environments, the approach leverages standardized exchange formats such as ASAM OpenDRIVE® and ASAM OpenSCENARIO®, ensuring compatibility with heterogeneous simulation platforms.

Industrial examples illustrate the application of these workflows in driving simulation and ADAS validation

Keywords: Scene and scenario fidelity, multi-fidelity simulation, scenario generation from real-world data

Copyright: © 20xx by the authors.
Licensee Driving Simulation Association.

Introduction

Simulation is a cornerstone of modern ADAS and automated driving development, but its effectiveness depends strongly on choosing the appropriate fidelity level for the task at hand. Excessive fidelity can slow down development and testing, while insufficient fidelity can lead to unrealistic results and missed failure modes.

The RoadRunner product family supports scene and scenario creation across a wide range of fidelity levels, from low-fidelity conceptual scenarios to high-fidelity representations of real-world environments. While interactive editing has long been an established method for detailed scene authoring and refinement, recent extensions focus on scalable and automated workflows, including programmatic APIs, which can leverage using agentic workflows. The workflows for reconstructing scenes and scenarios from recorded vehicle sensor data have been significantly enhanced, enabling the creation of realistic simulation content that can be further refined interactively.

The generated scenes and scenarios can be reused across a wide range of simulation toolchains by relying on standards such as ASAM OpenDRIVE® and OpenSCENARIO®.

Why Fidelity Matters in Simulation

Simulation is used across the entire ADAS and automated driving systems development lifecycle, from early concept development to system validation (Li et al., 2023). However, not all simulation tasks require the same level of fidelity, as it depends on the test objective, the system under test, and the current development phase (Oh, et al., 2024). According to Koopman, simulation fidelity level describes the extent to which it simplifies the real world (Koopman&Wagner, 2018). The following fidelities play a role when broken down for ADAS systems:

- Scene fidelity: realism and complexity of scenes
- Scenario fidelity: number and types of actors and movement complexity
- Dynamic fidelity: physical behavior of vehicles and actors
- Visual fidelity: level of graphical detail for visualization or sensor simulation
- Sensor fidelity: accuracy and realism of simulated sensor outputs, including noise characteristics, limitations, and environmental effects (e.g., weather, occlusions, materials)

Fidelity Levels: Typical Applications

Depending on the application, different fidelity levels are required. This paper focuses on scene and scenario fidelity, which directly affects how well simulation scenarios represent a real-world environment.

As Tab. 1 shows, different applications map to different fidelity requirements:

- Concept development and prototyping: Low fidelity scenes and scenarios with simplified layouts and traffic logic allow fast iteration and early feasibility studies.
- Functional safety and robustness testing: Medium-to-high scenario fidelity with correct road infrastructure, traffic rules, and interactions enable systematic testing of system behavior.
- Perception and sensor testing: High fidelity scenes, including detailed geometry, textures, and environmental context, support realistic sensor input generation.
- System level simulation: Very high fidelity across scenes and scenarios is required to validate complete system behavior, often with real-time constraints (HIL simulation).

Table 1: Fidelity requirements for different applications

Application	Scene fidelity	Scenario fidelity
Concept development and prototyping	Low	Low
Function safety and robustness testing	Medium to high	Medium to high
Perception and sensor testing	High	Low to high
System level simulation	High	High

This paper shows how multi-fidelity scene and scenario authoring, combined with standards-based interoperability, enables scalable and reusable driving simulation workflows. The proposed tool, RoadRunner, enables users to adapt scene and scenario fidelity to the application, rather than enforcing a single “one-size-fits-all” solution.

Scene and Scenario Creation

RoadRunner supports multiple scene and scenario creation methods, allowing users to choose the most appropriate method depending on fidelity needs and available data. Tab. 2 shows an overview of methods and their corresponding use cases.

Table 2 Overview of different authoring approaches and their typical use cases in ADAS development

Authoring approach	Typical uses
Interactive authoring	Concept design Detailed individual scenes and scenarios
	Enrichment, cleaning up and final polishing
Programmatic APIs	Scaling up Create variations
	Integration into CI pipelines
AI-assisted authoring	Make scene and scenario design easier accessible
	Speeding up design
Scenario Builder	Replication of real-world scenes and scenarios
	Creation of scene and scenario libraries

Interactive Authoring

RoadRunner provides a powerful interactive editor for manually creating and modifying road networks, junctions, tunnels, bridges, and environments. This approach is ideal for conceptual scenarios, design studies and manual refinement, and visual inspection. Creating and using snappable scene templates can accelerate scene generation significantly. Another method for accelerating scene generation is to import scene data from HD maps like HERE[®] HD and TomTom[®] HD or to import data from OpenStreetMap[®].

The interactive editor remains essential even in automated workflows, as it allows users to clean up, validate, and enhance automatically generated scenes. Common tasks include refining junction corners, fixing height jumps, adjusting lane marking and sidewalks, and placing additional buildings.

RoadRunner Scenario allows users to design actor motion and scenario logic on top of existing scenes. Actor behavior is specified using map-aware paths and explicit trajectory definitions. It contains a graphical logic editor for describing event-based interactions using actions, conditions, and triggers. This allows users to model behaviors such as lane changes, cut-ins, braking, or speed adjustments.

Programmatic and AI-Based Scene and Scenario Generation

For large-scale scene and scenario creation, manual approaches become very time consuming and costly. Therefore, RoadRunner provides APIs that enable scenes and scenarios to be created, modified, and managed programmatically.

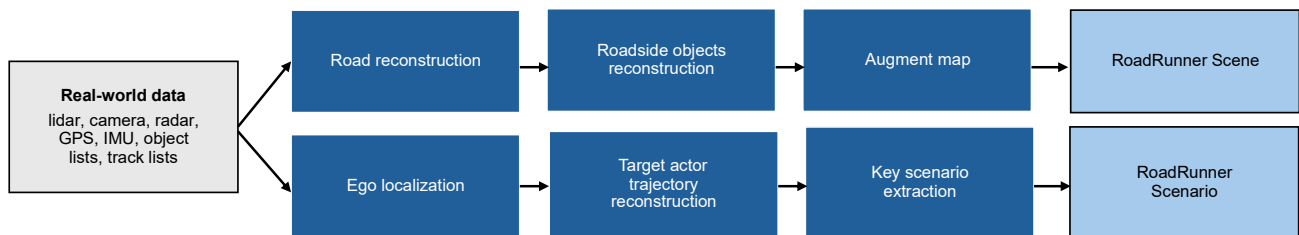


Figure 1 Automated scenario generation workflow in Scenario Builder. Real-world multi-sensor data is processed to reconstruct road networks, roadside objects, ego localization, and actor trajectories.

APIs For Scene and Scenario Editing and Project Handling

The MATLAB® API for scene authoring enables scene element programmatic creation and modification such as road geometry, lane configuration, and lane markings. Users can generate synthetic scenes with these APIs, create systematic variations, and update scene content without relying on interactive editing. In addition, MATLAB® provides APIs for scenario authoring, allowing users to programmatically add and modify actors and assets, define routes, and design event-based scenario logic. This enables efficient creation of large numbers of scenario variants and supports use cases such as coverage analysis.

Beyond authoring, MATLAB APIs are also available for scene and scenario project management, including opening, saving, importing, and exporting scenes and scenarios. This supports automated pipelines where scene and scenario content is generated, validated, and exchanged across tools.

AI-Assisted Authoring

The MATLAB scene and scenario APIs can be combined with AI agents, for example, by connecting them to external generative or agentic systems. This increases productivity by reducing manual scripting effort and improving the speed and comfort of scene and scenario generation, while keeping users in full control of the resulting content.

A key enabler for this integration is the MATLAB Model Context Protocol (MCP) Core Server, which provides a standardized interface between MATLAB and external AI applications. Using MCP, AI agents can interact directly with a running MATLAB environment to generate, analyze, and execute MATLAB code. This allows AI systems to invoke the RoadRunner scene and scenario API's programmatically, without manual copy and paste workflows.

Integration Into Heterogeneous Toolchains

In addition to the MATLAB API, RoadRunner provides a gRPC-based API. This API enables programmatic scene and scenario management from external programming languages such as C++ or Python. This allows integration into automated workflows, including CI/CD pipelines.

Automated Scene and Scenario Generation From Recorded Sensor Data

Scenario Builder for Automated Driving Toolbox™ (MathWorks®, 2026) is a support package that extends Automated Driving Toolbox and enables the automatic creation of scenes and scenarios from recorded real-world driving data. It supports the ingestion, synchronization, smoothing, and preprocessing of heterogeneous data sources, including camera, lidar, radar, GPS/IMU, vehicle trajectories, and object lists. Fig. 1 shows the general workflow.

From the synchronized inputs, Scenario Builder automatically extracts core scene and scenario elements such as:

- Road and lane geometry, including lane boundaries, curves, junctions, and lane add/drop.
- Traffic signs and roadside objects, like buildings and trees.
- Traffic participants with type, size, pose, and time-dependent trajectories.
- Critical driving events, such as braking, cut-ins, and lane changes.

In addition to full scene reconstruction, Scenario Builder supports an SD map augmentation workflow. Existing digital maps can be corrected and enriched using recorded sensor data. For example, Scenario Builder enables users to update lane counts, lane widths, road elevation, or geometric inaccuracies. Users can reuse existing map assets while increasing scene fidelity based on real-world measurements. The workflow is described in Fig. 2.



Figure 2 Workflow: Augment Standard Definition (SD) maps using recorded sensor data

To increase realism, Scenario Builder also provides automatic asset generation from images. Using AI-based methods, custom 3D assets—such as vehicles or static roadside objects—can be generated from single camera images and imported directly into RoadRunner scenes and scenarios. Compared to generic library assets, these data-driven assets closely reflect region-specific appearance, geometry, and texture, contributing to higher visual and sensor fidelity in simulation environments. Fig. 3 shows an example.

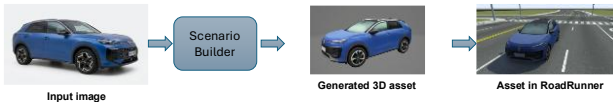


Figure 3 Automatic image to asset conversion.

Together, these workflows enable the reconstruction of concrete scenarios that closely reflect real-world driving behavior, forming a high-fidelity baseline for simulation and analysis.

Integration Into Simulation Toolchains

RoadRunner and RoadRunner Scenario provide direct export of all scenes, scenarios, and assets to widely used exchange formats. Scenes authored in RoadRunner—whether created interactively, programmatically, or from recorded sensor data—can be exported directly to ASAM OpenDRIVE, while scenarios defined in RoadRunner Scenario can be exported to ASAM OpenSCENARIO XML or DSL. In addition, scene geometry and assets can be exported to common 3D formats such as Filmbox (FBX®) or glTF™, enabling integration with visualization engines, rendering pipelines, and simulation toolchains. Together, these export capabilities enable seamless integration of RoadRunner content into any simulation, validation, and visualization tool that supports those standards.

Results and Industrial Experience

The proposed approach has been applied in several industrial ADAS/AD validation workflows. These applications provide practical evidence of its use in real-world development contexts.

BMW reported the use of RoadRunner as part of a virtual world generation pipeline for driving simulation. The workflow relies on RoadRunner to create detailed road and environment models and to integrate them into existing simulation infrastructures through open interfaces and exchange formats, including ASAM OpenDRIVE (Cao, 2023).

Aptiv reported simulation scenario reconstruction from recorded vehicle data, combining ego trajectories, traffic participants, and road infrastructure. The

generated scenarios were used in closed-loop simulation for ADAS development and regression testing, enabling the creation of large libraries of safety relevant scenarios derived from real driving behavior (Koravadi, 2023).

Ashok Leyland described the use of scenarios derived from real-world data for virtual validation of warning based ADAS functions. The reconstructed scenarios contain realistic 3D assets generated from images for perception validation (Gururaaj & Divya, 2024).

VUFO has created 300 ASAM openDRIVE files and reconstructed more than 1700 crash-scenarios based on real-world accident data using RoadRunner (MathWorks, 2025).

Across these applications, the results show that automatic scenario generation from real-world driving data enables efficient creation of realistic and reusable simulation content for industrial validation. A common observation is the benefit of supporting multiple fidelity levels within a single workflow, allowing scenarios to be adapted to different validation objectives. The use of standards-based scenario representations further enables reuse across heterogeneous simulation environments, decoupling scene and scenario authoring from execution and supporting varying industrial tool chains.

Conclusion

Effective simulation requires not only realistic content, but also the ability to select and manage the appropriate level of scene and scenario fidelity. RoadRunner supports this need by offering multiple authoring approaches, from interactive editing to fully programmatic generation.

Scenario Builder extends this capability by enabling automated, data-driven, and AI-assisted scenario creation, allowing users to scale realistic scenario generation while preserving flexibility and control. Together, these tools provide a practical solution for industrial simulation workflows across development stages and fidelity requirements.

The approach combines multi-fidelity scenario authoring within a single tool with standards-based interoperability, enabling flexible integration into diverse simulation environments without tool or vendor lock-in.

Looking ahead, agentic workflows are expected to accelerate scene and scenario creation. It also enables users to generate scenes and scenarios directly from natural language descriptions, making this approach more accessible for casual users and non-experts. RoadRunner is designed to support such workflows through its programmatic interfaces and standardized integration mechanisms, including MCP-based connections, enabling seamless interaction with external AI systems.

References

- Cao, H., 2023. Virtual world generation for BMW driving simulation. *MathWorks Automotive Conference Europe*, 2023.
- Gururaaj, P and Divya, T. 2024. Virtual validation of warning-based ADAS features. *MathWorks Automotive Conference India*, 2024
- Koravadi, K. 2023. Scenario harvesting using Automated Driving Toolbox and RoadRunner Scenario. *MathWorks Automotive Conference North America*, 2023.
- Koopman, P. and Wagner, M., 2018 "Toward a Framework for Highly Automated Vehicle Safety Validation." *WCX World Congress Experience*, Detroit, Michigan, United States, pp. 2018-01-1071. Available at: <https://doi.org/10.4271/2018-01-1071>.
- Li, Y. et al., 2023 "Choose Your Simulator Wisely: A Review on Open-source Simulators for Autonomous Driving." arXiv. Available at: <https://doi.org/10.48550/arXiv.2311.11056>.
- MathWorks® (2026). Scenario Builder for Automated Driving Toolbox Support Package: <https://www.mathworks.com/matlabcentral/fileexchange/117450-scenario-builder-for-automated-driving-toolbox>
- MathWorks (2025) *Using RoadRunner to Create Realistic Crash Simulations*. Available at: https://www.mathworks.com/company/user_stories/advancing-road-safety-with-automated-scenario-generation.html (Accessed: June 11, 2026).
- Oh, T., Cho, S. and Yoo, J., 2024 "A Reliability Evaluation Methodology for X-in-the-Loop Simulation in Autonomous Vehicle Systems," *IEEE Access*, 12, pp. 193622–193640. Available at: <https://doi.org/10.1109/ACCESS.2024.3519713>.

Developing ADAS with High-Fidelity Vehicle Models and Integrated Controllers

Alyssa Scheske¹, Ahmet Keles², Bharath Kaimal³

(1) Applied Intuition, Balanstrasse 71a Munich 81541 Germany, e-mail: alyssa@applied.co

(2) Ford Otosan, Akpınar, Hasan Basri Cd. No:2, 34885 Sancaktepe/Istanbul, Turkey, e-mail: akeles7@ford.com.tr

(3) Applied Intuition, 2350 Greene Rd, Ann Arbor, MI 48105, USA, e-mail: bharath.kaimal@applied.co

Abstract – This product solution presents a methodology for validating Advanced Driver Assistance Systems (ADAS) through high-fidelity simulation, demonstrated via a collaboration between Ford Otosan and Applied Intuition. The objective was to construct a virtual truck model that closely replicates real-world dynamic behavior, reducing dependence on costly physical road testing. A three-stage validation framework is proposed, encompassing open-loop plant model validation using TruckSim, integration of proprietary low-level controllers, and high-level ADAS algorithm testing, with this study focusing on the first two stages. The parameterized TruckSim model was correlated against real-world drive logs across longitudinal and lateral maneuvers, achieving R^2 values as high as 0.9977 following controller integration—a marked improvement over a kinematic bicycle model baseline. Error was quantified using L2, R^2 , and Sprague-Geers metrics across multiple test scenarios. Residual discrepancies are attributed to environmental variability, positional offset, and generic vehicle-class parameters. These findings establish a trusted simulation platform supporting SIL, MIL, and HIL workflows for ADAS development, reducing the time and resources required for physical validation.

Keywords: Advanced Driver Assistance Systems (ADAS), vehicle dynamics simulation, TruckSim, model correlation, virtual validation

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

Confidence in a simulated vehicle's ability to replicate real-world behavior is fundamental to virtual development. For engineers designing ADAS and autonomous systems, this confidence depends on the fidelity of the underlying vehicle model and the controllers driving it. If the simulated vehicle does not respond as its physical counterpart would, every downstream test, from control logic to perception, risks being built on false assumptions.

To address this challenge, Ford Otosan partnered with Applied Intuition to integrate high-fidelity TruckSim vehicle dynamics models with the Applied Development Platform (ADP) and Ford Otosan's proprietary low-level controllers. This collaboration demonstrates how accurate vehicle modeling and real-world controller integration can provide a trusted foundation for virtual validation.

VehicleSim, the umbrella suite of CarSim, TruckSim, BikeSim, and SuspensionSim, is widely recognized as an industry standard for physics-based modeling. Each product is integrated directly into Applied

Intuition's simulation ecosystem to support high-fidelity dynamics across a variety of vehicle platforms. Ford Otosan leveraged TruckSim to create a detailed, physics-driven truck model as the foundation for their virtual testing and controller validation.

These tools enable an ADAS engineer, not a vehicle dynamics specialist, to move the needle on their vehicle model. By switching from a simplistic kinematic bicycle model to a TruckSim model, results improve drastically. The increase in fidelity is well worth the effort, even if the ADAS engineer may incorporate more assumptions into their high-fidelity models.

The goal: to build a virtual truck that behaves indistinguishably from its physical counterpart. With this level of correlation, Ford Otosan can develop, evaluate, and validate ADAS and autonomous systems primarily in simulation—reducing the cost, time, and risk associated with physical test drives.

Ford Otosan's Goals

Ford Otosan's focus on high-fidelity simulation aligns with Applied Intuition's mission to accelerate the

world's adoption of safe and intelligent machines. Together, the teams built a workflow that enables engineers to develop, test, and validate software using trusted, physics-based vehicle models—reducing the burden of costly and time-consuming physical tests.

The collaboration began with the parameterization of a base vehicle model, also referred to as the vehicle plant model, followed by the integration of Ford Otosan's low-level controllers for simulation. These controllers manage and execute control actions for subsystems such as steering, transmission, and braking. Integrating them into the virtual vehicle creates a close representation of real-world performance.

Ahmet Faruk Keleş, an ADAS SIL/HIL Engineer at Ford Otosan, led the integration of the proprietary control logic into the model. His goal was to achieve a high degree of correlation between the physical vehicle and its simulated counterpart. "Achieving a close match between the physical truck's behaviour and the simulated model provides confidence that the model and workflow are accurate," Keles said. "This validation is critical for projecting the vehicle's response in further simulations."

He added, "The main goal of creating a correlated vehicle model is to reduce the dependency on vehicle testing for ADAS development and replace it with SIL, MIL, and HIL workflows. With a correlated model, we can trust that what we observe in simulation matches what occurs on the road. This reduces effort and resources spent on physical testing, particularly during parameter tuning, and creates a safe environment for exploring new feature concepts."

This goal—grounded in modeling accuracy, controller fidelity, and iterative correlation—sets the foundation for a more efficient virtual development process. With these fundamentals established, the next step is to examine the methodology Ford Otosan used to correlate its vehicle model.

Vehicle Model Methodology

To deliver on vehicle model correlation, Ford Otosan adopted a methodology that systematically verifies model accuracy, then controller integration—ensuring that every simulation step represents real-world behavior.

VehicleSim models are designed to be modular. Applied Intuition provides extremely robust physics models with templates for vehicle subsystems like powertrain, suspension, tires and more. Customers can quickly get started with these templates or they can choose to bring in their own external subsystems.

The majority of vehicle subsystems include control modules, such as traction control systems, or transmission shift controllers for the powertrain. We recognize that these control modules contain a customer's IP, hence we provide full flexibility to customers to integrate their control strategies into VehicleSim. Running the same controller logic in simulation that runs on-vehicle adds an additional layer of confidence when building a virtual vehicle model. The plant model and control systems work in tandem to provide the best virtual representation of a vehicle.

The vehicle model verification process is approached in three steps: open-loop TruckSim model validation, followed by integration of low-level controllers with the TruckSim model, and finally adding any high-level controllers that might be used in AV/ADAS. This project focused only on the first two stages. See Figure 1 to gain a better understanding of a system representation with Stages 1 and 2.

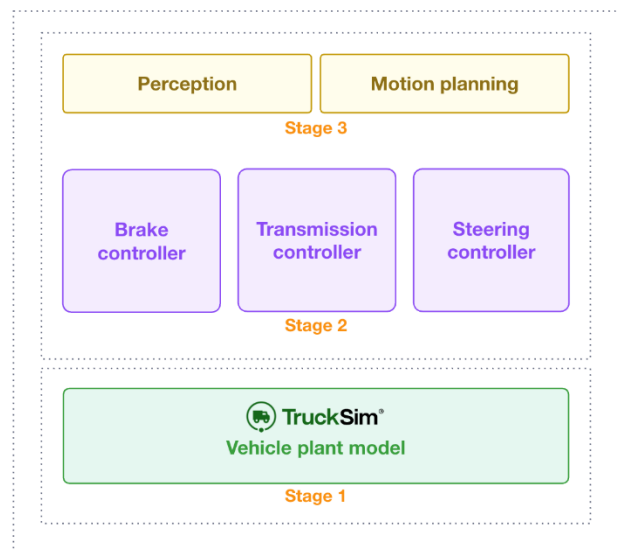


Figure 1: Modular, virtual representation of a vehicle in three stages

Applied Intuition's + Ford Otosan's Approach

Given the modularity of TruckSim models, Ford Otosan's integration bypasses the internal engine and transmission controllers and directly sends ADAS desired control commands and transmission signals, either from collected vehicle logs or their stack, to drive the wheels. TruckSim's configurable I/O makes this possible. Figure 2 describes the signal exchange between the low-level controllers and the TruckSim module after stage 1 and 2 are completed.

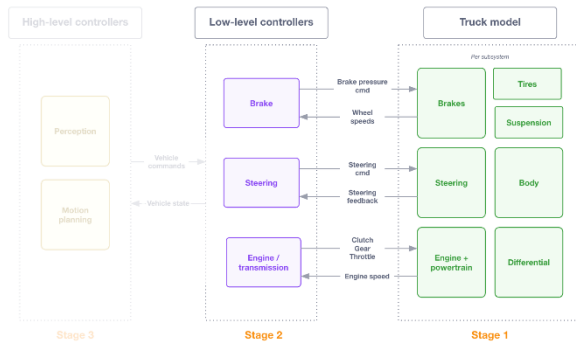


Figure 2: Signal exchange between low-level controller and TruckSim

TruckSim is seamlessly integrated into ADP as a selectable vehicle model. ADP allows for the creation of full scenarios to simulate the model and integrated controls with appropriate assumptions. This validation procedure compared test logs of a physical truck to the simulated truck running in open-loop simulation. The test log control signals are directly fed into the simulation, the logs define the scenario and only the ego's initial position needs to be defined on the map.

An HD map and terrain file of Ford Otosan's testing area were also imported into ADP. Using the same map in both drive logs and in simulation ensured consistency between the real-world and simulation environments, which is crucial for validating the vehicle model.

Let's look at Stages 1 and 2 in detail.

Stage 1: Open-loop TruckSim model validation

The first stage focuses on parameterizing the vehicle plant model by using the TruckSim product. TruckSim's system-level math models simulate a vehicle's physically measured responses without needing detailed knowledge of component materials or intricate suspension designs. To do this, they require data on suspension kinematics and compliances, tire properties, powertrain, brake and steering data, and environmental conditions. This approach ensures the models remain accurate enough for evaluating new designs and control strategies, even with limited reference data.

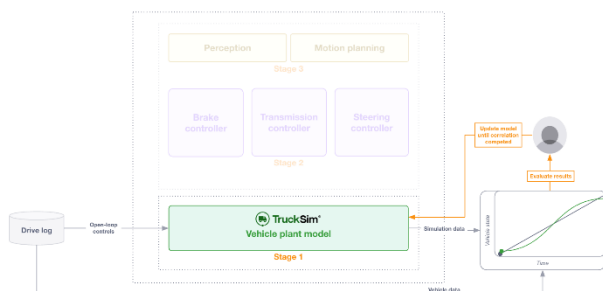


Figure 3: Open-loop plant model validation workflow

After parameterizing a model, we evaluate the pure dynamic response of the TruckSim model over a broad range of longitudinal and lateral maneuvers. We then verify the model by isolating open-loop controls for throttle, brake, and steering from the collected drive data into the plant model, ensuring the vehicle response matches as expected. Figure 4 explains this process.

Ahmet's Stage 1 results compare log data against a TruckSim model. The log's throttle signal is used as open-loop input to the TruckSim model. In Figure 5, the R2 error is 0.9741.

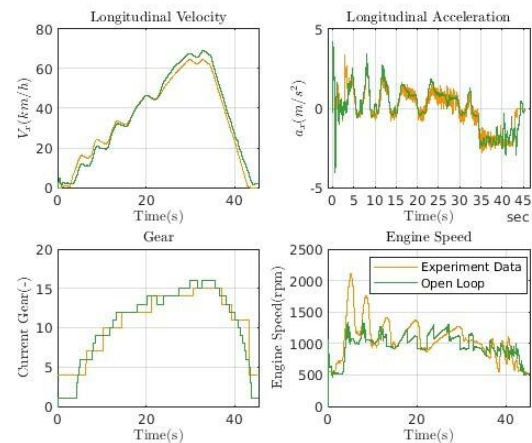


Figure 4: Open-loop comparison of stage 1 TruckSim model and the test log. The recorded control signals from the test log are used as input.

The TruckSim model should remain unmodified after completion of Stage 1. When correlating the next phase with low-level controllers in the loop, this minimizes the risk of overcompensating sim-to-real gaps from the controls in the vehicle model. This vehicle is representative and will now unmask the controls deficiencies going into the second correlation stage.

Stage 2: Validation with low-level controllers + TruckSim model

The second stage focuses on integrating low-level control logic with the parameterized vehicle model. For example:

- Integrating transmission control logic to command the gear in the vehicle model
- Integrating a brake controller sending desired brake pressure to the vehicle model

Ford Otosan integrated low-level logic for longitudinal and lateral controllers. For longitudinal control, they integrated their brake, and transmission

controllers, and for lateral control they implemented their steering controller.

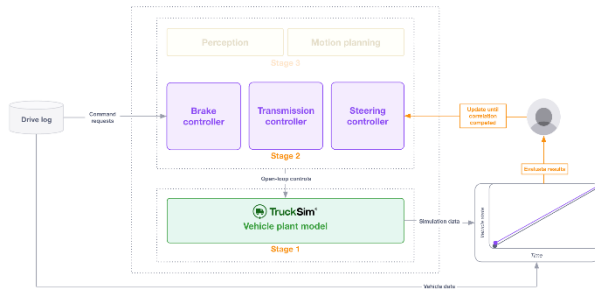


Figure 5: Low-level controller validation workflow

To validate correlation at this stage, we input open-loop data to the integrated low-level controllers. The output of the low-level controllers was then input directly to the vehicle model, and the fully integrated behavior was evaluated in isolation without closed-loop response. Ideally, if controllers have exactly the same interfaces as the real-world systems, no additional correlation effort is required. But if there is any parity difference, the integration requires support from the controls team in calibrating the low-level controllers.

Ahmet's Stage 2 results compare log data against a TruckSim model with Ford Otosan's custom controllers. In Figure 7, the R2 error from Stage 1 is 0.9741, whereas results improved to 0.9977 with Ford Otosan's integrated custom controllers in Stage 2.

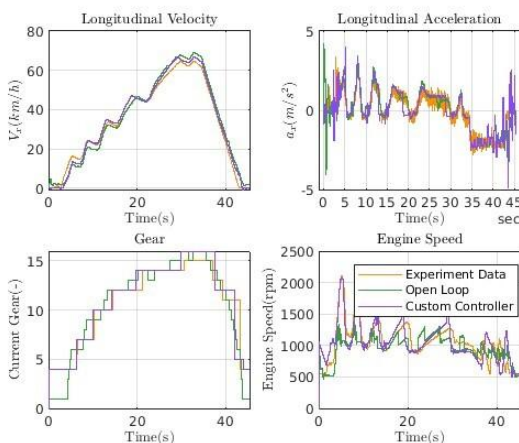


Figure 6: Comparison of Experiment Data, Open Loop Stage 1 & Low-Level Controller Stage 2

Table 1 captures the results with different error methods applied against the log data, including [L2](#), [R²](#), and [Sprague-Geers](#) methods.

Table 1: Plots shown for Log ID: can1_br_2_02_125_uh_2 shown above in Figure 6

Log ID	L2 Error Stage 2	L2 Error Stage 1
can1_br_1_325_uh_1	2,02	2,12
can1_br_2_0275_uh_1	2,06	2,85
can1_br_2_02_125_uh_1	1,98	2,41
can1_br_2_02_125_uh_2	2,32	3,36
can1_br_2_04_uh_1	1,84	2,58
can1_br_2_04_uh_2	2,45	2,75
can1_br_2_065_uh_1	1,61	2,07

Eliminating unknowns as complexity is added to the vehicle model - this is the key advantage to this approach. Since the vehicle model response is already correlated, it can be considered “locked,” and focus can be spent on ensuring the low-level controllers function as expected in the simulation environment. For example, the gain on a yaw controller might be adjusted to compensate for the gap between simulation and real world.

3: Validation of high-level controls (i.e., ADAS/AV algorithms)

While we focused on vehicle model and low-level controller validation, ADP enables users to rigorously test the full ADAS/AV system, all while using the same high-fidelity VehicleSim model. This ensures a realistic vehicle dynamics response for ADAS/AV development. However, stage 3 is out of scope for this blog post.

Now let's review the results by maneuver to prove the methodology.

Results and Discussion

To validate the correlated vehicle model and integrated controllers, Ford Otosan compared real-world test logs with simulation outputs across a series of key scenarios - from longitudinal acceleration and deceleration, to lateral maneuvers such as double lane changes. By evaluating metrics relevant to each scenario, such as vehicle speed, acceleration, gear selection, yaw rate, and lateral acceleration, the team confirmed high agreement between physical and virtual vehicle behaviors.

The correlation of the vehicle's integrated low-level controllers is complete. Keles concludes, “These results give our ADAS team confidence to trust the results of simulations. With this model, we are sure that the phenomenon we observe in simulation will also be observed in road testing, reducing the need for road tests for controller tuning and ADAS development.” The current snapshot of the TruckSim model is sufficient for Ahmet's ADAS testing. If Ford

Otosan desired to further optimize their model, they could iterate and collect vehicle test data for each vehicle subsystem.

Longitudinal 0 - 60 kph - 0:

This scenario involved testing the vehicle's acceleration and deceleration capabilities. By comparing against the 7 real-world logs and simulated data, we validated the longitudinal dynamics of the model. Vehicle speed, acceleration and gear selection were compared in Figure 8.

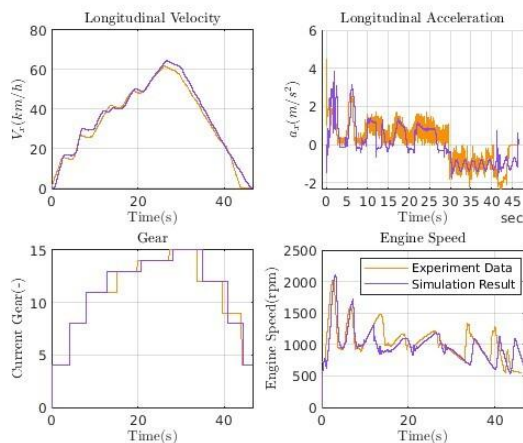


Figure 7: Longitudinal 0 - 60 kph - 0 results

Table 2: Longitudinal 0 - 60 kph - 0 error calculations across 7 individual logs

Experiment ID	L2 Error	R ² similarity
1	2.35	0.9834
2	3.89	0.9585
3	3.62	0.9420
4	4.9	0.9440
5	3.2	0.969
6	4.09	0.9533
7	1.87	0.9899

Double Lane Change:

This scenario tested the vehicle's lateral dynamics. Comparing the real-world and simulated data for a lane change helped validate the model's ability to accurately simulate the steering controls and subsystem. In Figure 9, the R2 error for the yaw rate is 0.93 and 0.83 for lateral acceleration.

The log data did not contain roll angle; the relevant yaw rate and lateral acceleration signal that were present in the log data are evaluated.

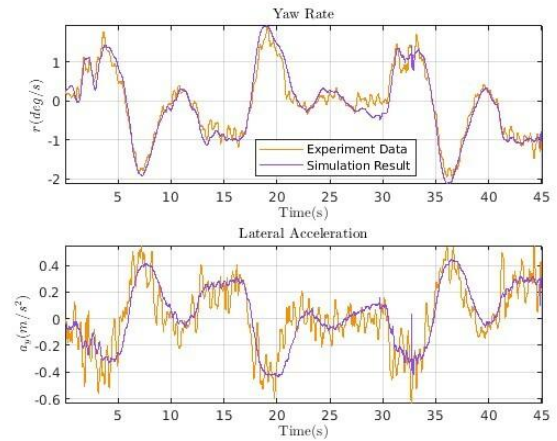


Figure 8: Double Lane Change results

Table 3: Double Lane Change yaw rate error calculations across 3 logs

Experiment ID	L2 Error	R ² similarity
1	0.003	0.122
2	0.005	0.182
3	0.003	0.276

Table 4: Double Lane Change lateral acceleration error calculations across 3 logs

Experiment ID	L2 Error	R ² similarity
1	0.092	0.130
2	0.071	0.154
3	0.068	0.214

ADAS Development:

Comparing the correlated model against a simplistic kinematic bicycle model can highlight the importance of using a more detailed model for ADAS development. The difference between the correlated TruckSim model and the kinematic model can demonstrate the value of trust in simulation backed by real data.

The **Longitudinal 0 -> 60 kph -> 0** scenario was executed with both the kinematic bicycle and TruckSim models. To run this, acceleration input from the log is normalized to a throttle input, leading to a larger difference in the kinematic bicycle. Figure 10 highlights the gap in acceleration performance with the kinematic bicycle model.

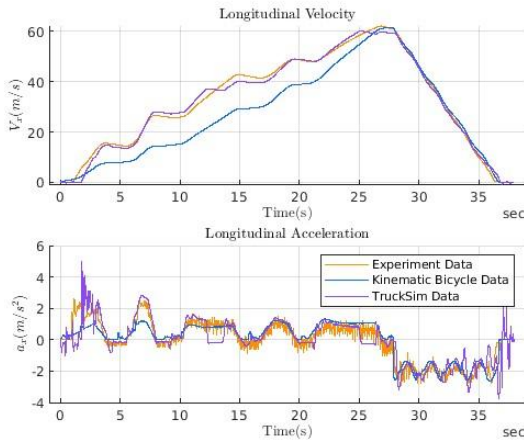


Figure 9: Longitudinal 0 -> 60 kph -> 0 results comparing Ford Otosan's integrated vehicle model against a kinematic bicycle model

Table 5: Longitudinal 0 -> 60 kph -> 0 error calculations of a kinematic bicycle model across 7 logs

Style	L2 Error	R ²
can1_br_1_325_uh_1	8,41	0,7668
can1_br_2_0275_uh_1	14,46	0,4286
can1_br_2_02_125_uh_1	5,91	0,8468
can1_br_2_02_125_uh_2	11,78	0,68
can1_br_2_04_uh_1	7,33	0,8433
can1_br_2_04_uh_2	6,75	0,8685
can1_br_2_065_uh_1	7,78	0,8256

The **Double Lane Change** scenario was executed with both the kinematic bicycle and TruckSim models. Figure 10 highlights the gap in yaw rate performance the kinematic bicycle model cannot close against the log data and integrated low-level controllers with TruckSim.

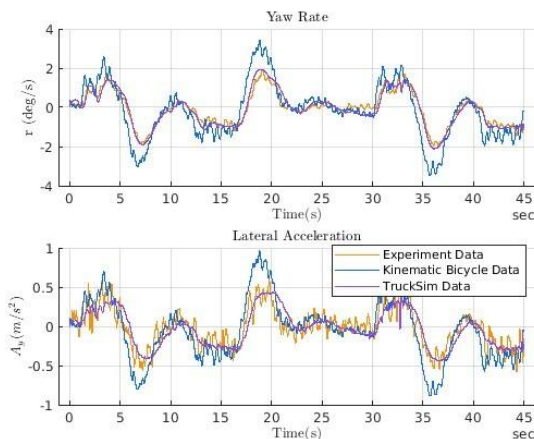


Figure 10: Double Lane Change results comparing Ford Otosan's integrated vehicle model against a kinematic bicycle model

Across multiple experiments, the vehicle model and controllers demonstrated tight correlation in both longitudinal and lateral dynamics, as indicated by low L2 and Sprague-Geers errors and high R² similarity. Where discrepancies existed the team

acknowledged and documented these factors, maintaining transparency about the strengths and limitations of the model. The largest gaps in the results are due to the following factors:

1. Variation in starting position between the log data and simulation (thus some logs might follow a slightly different road gradient),
2. A test day with extreme temperatures (altering parameters such as effective rolling resistance and surface μ), and
3. Tire and powertrain properties using vehicle-class generic values (for example, some inertias or rotational stiffnesses).

This validation establishes a trusted simulation platform for ADAS development, replacing costly physical testing for many parameter tuning and system validation tasks. As Ford Otosan and Applied Intuition continue to expand their virtual validation workflows, these results demonstrate the real-world efficiency and reliability achievable with high-fidelity vehicle models and integrated control systems.

Conclusion

As the automotive and trucking industries continue to face challenges from the mounting complexity of software systems, accurate vehicle dynamics remain a crucial element of development. Through this case study with Ford Otosan, we explored how Applied Intuition's tools support high-fidelity vehicle model validation. As more automotive software development and testing is virtualized, developers can improve their efficiency only by trusting these underlying physics-based models.

Learn how VehicleSim and ADP are enabling next-generation virtual validation for mobility leaders worldwide.

References

- "High-Fidelity Adas Models." Applied Intuition, Applied Intuition, 17 Nov. 2025, www.appliedintuition.com/blog/adas-with-high-fidelity-vehicle-models.

Test Management Process within SCANeR and the 3DEXPERIENCE Platform

Reine Gros¹, Sandrine Loembe², Franck Corbier¹

(1) Dassault Systèmes SE, 1 Impasse Annette Zelmman, 54000 Nancy, France, e-mail: {Reine.Gros, Franck.Corbier}@3ds.com

(2) Dassault Systèmes SE, 1/3 Rue Jeanne Braconnier, 92360 Meudon La Forêt, France, e-mail: Sandrine.Loembe@3ds.com

Abstract – The rapid growth of software-driven products is at the core of innovation across multiple industries such as avionics, aerospace, railway, and especially automotive. Modern vehicles increasingly rely on complex embedded functions—ranging from sensor perception and fusion to advanced driver assistance systems and autonomous driving.

To manage this complexity, Systems and Software Engineers depend on best-in-class tools supporting requirements management, system and E/E architecture definition, design, and code development. However, equal attention must be given to the right side of the V-cycle, where Test and Validation activities play a critical role. Solution such as SCANeR enable realistic and scalable driving simulation environments.

Test Management provides the foundation for orchestrating all testing activities while controlling associated assets and artifacts. Using the example of NCAP requirements for Advanced Emergency Braking Systems, this presentation illustrates how to define structured test specifications, create and manage test cases and SCANeR simulation scenarios, execute simulations and capture results as verifiable test evidence and ensure full traceability between requirements, test cases, and validation outcomes.

Keywords: AD/ADAS, xIL, System Validation, Test Management, Traceability

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

Systems and Software Engineers depend on best-in-class tools supporting requirements management, system architecture and E/E architecture definition, disciplines design (vehicle dynamics, suspension, chassis, battery, control...), and software development.

However, equal attention must be given to the right side of the V-cycle, where Test and Validation activities play a critical role.

Test Engineers leverage advanced tools for early validation through simulation (e.g., CATIA Dymola, CATIA ControlBuild and others...) as well as for integration and system-level testing, including Model-, Software-, Hardware-, Driver-, and Vehicle-in-the-Loop (xIL) approaches.

CATIA SCANeR toolchain allows to build and execute realistic and scalable driving simulation environments (terrain, scenarios, correlated EGO vehicles and sensors, traffic with cars, bicycles, trucks, pedestrians & animals, control law models, ECU models, ...).

Key challenge

A key challenge remains: how to efficiently manage changes, versions, and product diversity throughout the system lifecycle while ensuring safety, performance, and certification compliance. Beyond robust development processes, establishing a well-defined, independent Test and Validation strategy is essential—and often mandated by safety standards.

Methodology

Test Management (Figure 1) provides the foundation for orchestrating all testing activities while controlling associated assets and artifacts.

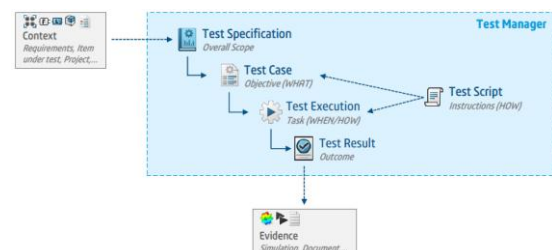


Figure 1: Test Manager high level data model

We will use, as an example, the NCAP (European New Car Assessment Program) requirements for AEBS (Advanced Emergency Braking Systems) to illustrates how to define structured test specifications aligned with functional, logical, and physical system representations.

AEBS is a device that allows a car or a truck to detects an imminent collision with a vehicle, pedestrian, bicycle or other detected obstacle and automatically activate the vehicle's braking system to slow the vehicle down and thus avoid or mitigate a collision.

Note that SCANer provide an Euro NCAP pack containing configurations and scenarios allowing test engineer to validate AD/ADAS functions and for sure the AEBS system.

We can define or link / translate the requirement documents inside the 3DEXPERIENCE Platform using the TRM (Requirement Management) product like we can see below in the Figure 2.

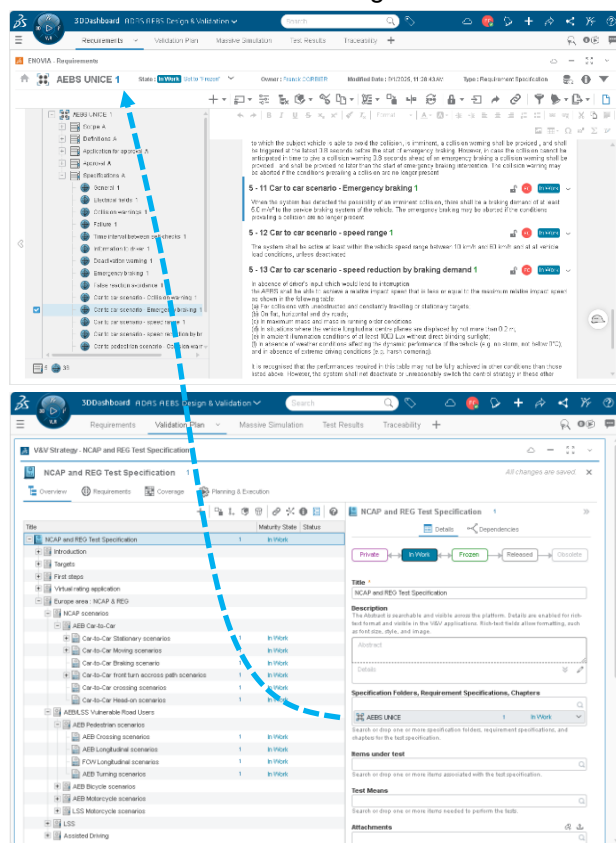


Figure 2: Test Specifications covering Product/Regulation Requirements

Then the test engineer can start the definition of the test specifications that will covers de product requirements. The test engineer will use the V&V Strategy application of the Test Manager role.

From the Test Specification, the test engineer can create one or multiple Test Cases. In fact, the Test Cases of the Test Specification are those that cover one or multiple Requirements (Figure 3).

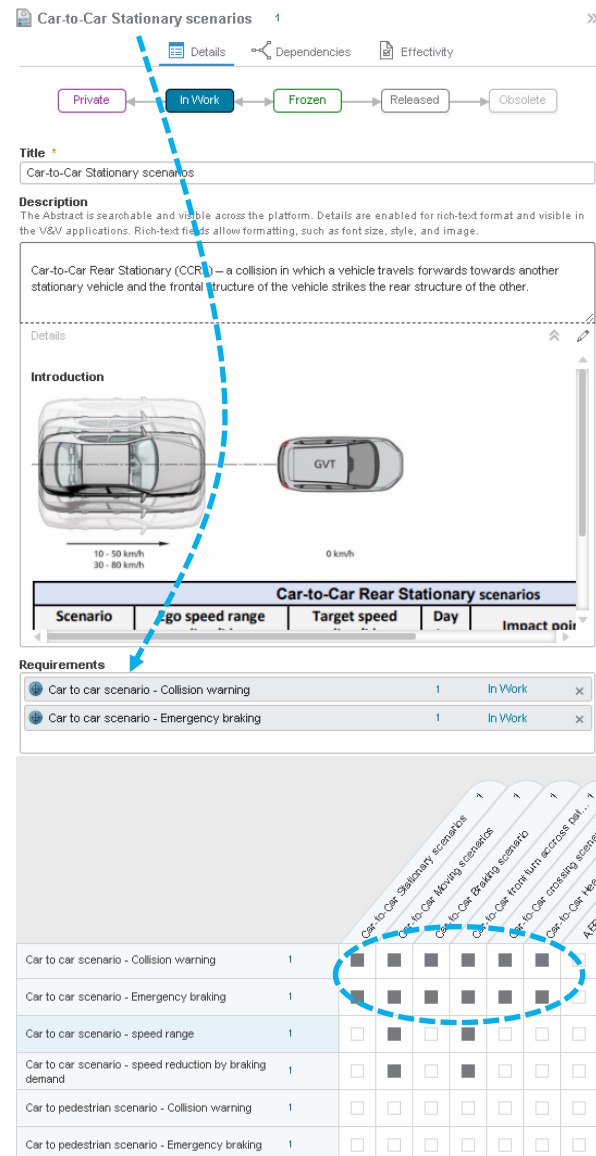


Figure 3: Test Case covering Requirement

For example, test engineer define that the Car-to-Car stationary scenarios will cover the Car-to-Car scenario - Collision Warning and also the Car-to-Car scenario – Emergency braking.

When all the test cases are defined, we can see that Car-to-Car scenario – Emergency braking covers multiple requirements by using the traceability matrix feature.

This is an important feature: test engineer knows what Test Cases have to be used for the validation of a new system under tests but also what are the relevant test cases to execute when one requirement has changed.

Now the Test Engineer can link what it called Test Scripts to each Test Cases. A Test Script can be a textual list of actions and associated check but also, for our usage, a SCANer scenario (Figure 4).

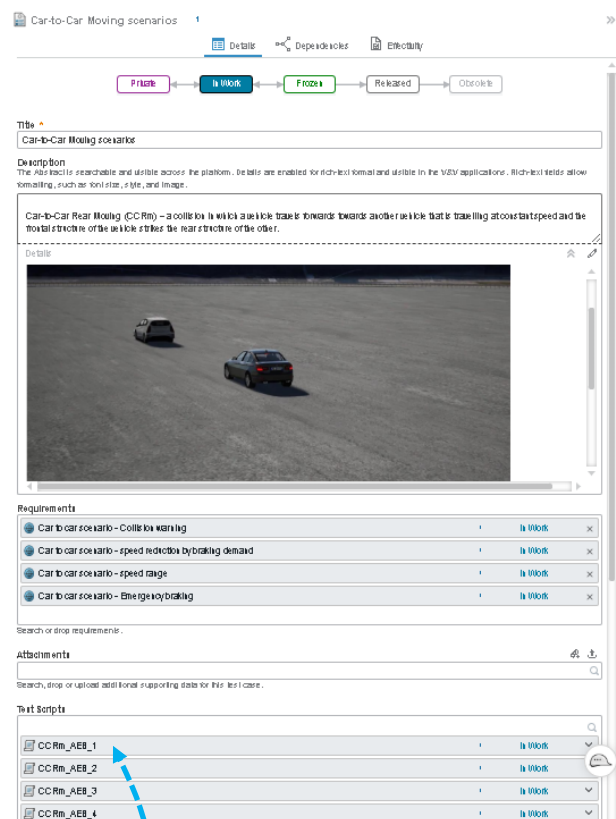


Figure 4: Test Script

Last objects to be associated is a Test Execution which provide important information (Figure 5) for the execution the test. For example, what kind of execution (Model In the Loop, Software In the Loop, Hardware In the Loop), the test Means (Computer, HPC for Massive Simulation, Co-Simulation Platform, Test Bench...), who is assigned to this test, and the needed assets and models (Terrain, Vehicles, Drivers, FMUs, Traffic, Scenario...).

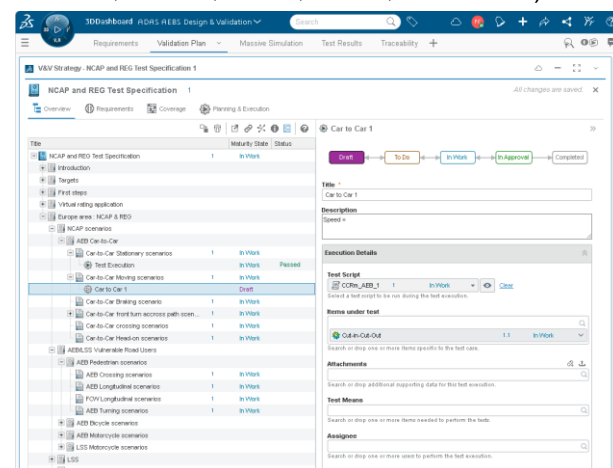


Figure 5: Test Execution and execution detail list

Now the defined Test Specification is ready to be applied on a test session of the AEBS. Even if the test is manual on a computer or HIL bench, or massive on the Cloud, the results are saved in the 3DEXPERIENCE Platform. For each executed test, we can get the status (passed, failed) of the test and the time stamped record of pre-selected simulation variables (Figure 6).

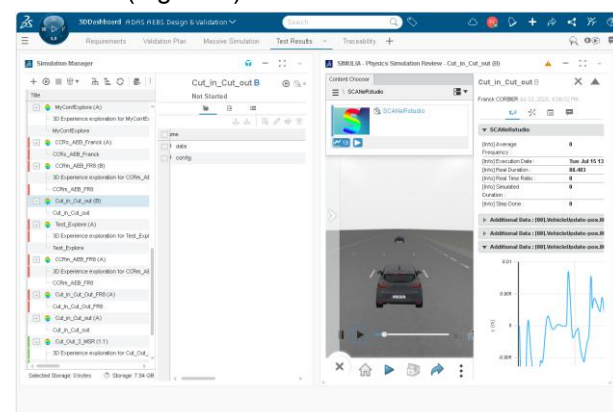


Figure 6: Test Result analysis

Test Analyst can also use the test records to replay and observe the scenario execution and the recorded values of parameters and variables.

All data associated with the test management process can be consolidated and monitored through the System Synthesis role. Beyond simple visibility, this enables end-to-end traceability across requirements, test specifications, test cases, and execution results.

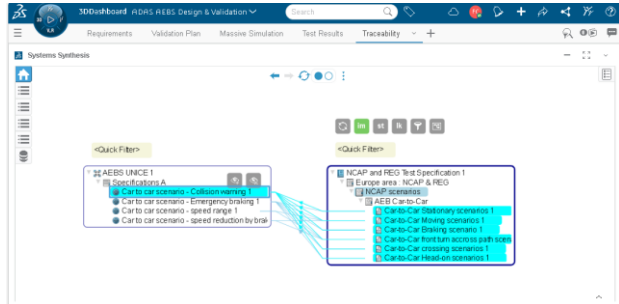


Figure 7: Full traceability management

This approach allows precise identification of the validation scope for a given Feature or System Under Test, ensuring that all necessary test activities are clearly defined and executed. In addition, it supports systematic impact analysis by highlighting the consequences of any change on both the test and design domains.

As a result, engineers can immediately determine which elements need to be updated or created, including specific sections of the test specification, associated test cases, test scripts, and corresponding test executions thereby maintaining consistency and completeness across the entire V&V process.

Results and discussion

The Test Manager role within the 3DEXPERIENCE Platform enables organizations to define, execute, and monitor comprehensive Validation & Verification (V&V) strategies. It centralizes both virtual simulations and physical testing activities, providing a complete and consistent view of the validation process.

The benefits of deploying a Test Management process are multiples:

- Optimize test coverage while minimizing redundancy through effectivity management
- Monitor V&V coverage via requirement-to-test traceability matrices
- Plan and schedule simulations in alignment with project timelines
- Organize and execute tests efficiently using test suites
- Consolidate results from virtual and physical testing environments
- Track validation progress and design maturity through unified dashboards

Conclusions/implications

The main advantages lie in increased coverage of the testing process and early detection of development issues.

They are also manifested through harmonized governance and seamless execution of verification and validation strategies at every stage of the development cycle.

Every product requirement is now reflected in the validation evidence, ensuring complete traceability throughout the entire process.

Finally, test campaigns become more consistent, rigorous, and efficient, paving the way for reliable and reproducible results.

HIL DIL SIMULATION DESIGN

Extending a Modular Simulation Framework to Support Immersive Virtual Reality for Human-in-the-Loop Driving Studies

Peter Baker, Joseph Mitchell, Emil Chodowiec, Xizhe Zhang, Siddartha Khastgir, Paul Jennings

Safe Autonomy Research Group, WMG, University of Warwick, Coventry, UK CV4 7AL,
e-mail: {peter.baker, joe.mitchell, emil.chodowiec, jason.zhang, s.khastgir.1, paul.jennings}@warwick.ac.uk

Abstract - This paper presents the extension of a modular, standards-aligned simulation framework to support immersive virtual reality for human-in-the-loop driving studies. Rather than developing VR as a standalone simulator, the approach integrates immersive visualisation and interaction as interchangeable components within an existing simulation architecture built around OpenSCENARIO and OpenDRIVE standards. This enables identical scenario assets and execution logic to be used across VR, desktop, and large scale simulator configurations without modification. A portable VR setup was made up using a laptop to run the simulator, a Meta Quest 3 headset and a foldable rig, which included a steering wheel, pedals and seat. This supported both manual and automated driving modes within a single session. The system was deployed at multiple public engagement events, where it was operated by a single facilitator and accommodated participants of all ages with no prior simulation experience. Observational findings indicate that immersive VR enhanced participant engagement and sense of presence compared to desktop simulation, particularly during automated driving sequences. The results demonstrate that VR can be incorporated into a broader simulation ecosystem without compromising standardisation or reproducibility, forming a suitable basis for future controlled human factors studies.

Keywords: virtual reality, driving simulation, human-in-the-loop, modular simulation, automated driving

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

Safety Assurance and Virtual Testing

The safety assurance of Automated Driving Systems (ADSs) cannot be achieved through real-world testing alone. The mileage required by ADSs to confidently demonstrate safety makes physical testing alone impractical (Kalra and Paddock, 2016). A scenario-based testing approach where targeted simulation of conditions defined by a system's Operational Design Domain (ODD), focuses validation effort where it matters most (Zhang, Khastgir, and Jennings, 2023). An ODD defines the specific conditions under which a system is intended to function safely (SAE, 2021). ISO 34503 (ISO, 2023) and BSI 1883:2025 (BSI, 2025) provide a common vocabulary for defining the specific conditions under which the system is designed to operate, in a human-readable format. There are also machine readable formats such as ASAM OpenODD (ASAM, 2025). Scenarios exist in common formats including those belonging to the ASAM OpenX family, which enable scenario definitions to be easily shared across tools and virtual testing environments.

Virtual scenario-based testing provides substantial reductions in the time and resources required for

real-world setups, while also eliminating the risks involved with human-in-the-loop testing. The purpose of virtual testing is not to replace real-world trials, but to significantly reduce their scope, retaining real-world testing for the scenarios that matter the most while exploring the broader parameter space virtually. This places significant demands on simulation fidelity, where environments must be realistic enough to produce meaningful results, execution must be reproducible and scalable, and outcomes must be comparable across simulated and physical testing configurations.

Simulation Architecture

A simulation execution framework developed in prior work provides the technical basis from which this work extends. As shown in Fig. 1, the framework supports retrieval from the Safety Pool™ scenario database (Safety Pool, 2024) alongside automated environment generation and dynamic scenario execution based on ASAM OpenDRIVE (ASAM e.V., 2022a) and OpenSCENARIO XML (ASAM e.V., 2022b). A 3D rendering of scenarios forms the sensor simulation and visualisation layer, realised in this work through a scalable Unreal Engine-based testing platform (Mitchell, et al., 2024). Simulation data is

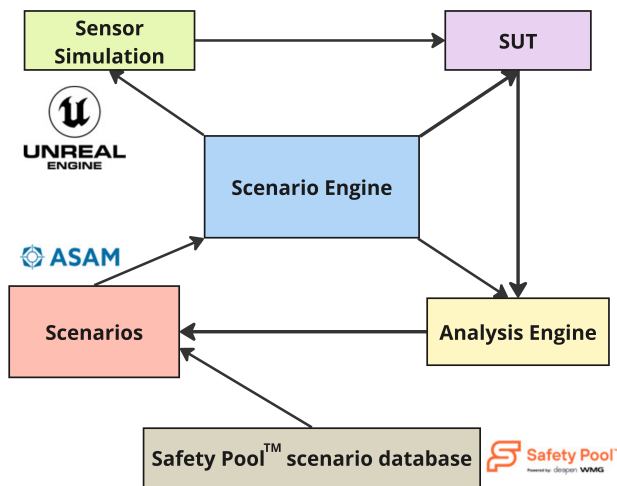


Figure 1: The modular simulation framework

transmitted to the system under test (SUT) via ASAM OSI (ASAM e.V., 2024) which serves as a standardised data exchange interface. In parallel, simulation data is output to the analysis engine for processing either at runtime or after simulation execution. Based on the simulation results, the analysis engine guides the parameter space exploration (defined in scenarios) with a goal to stress test the system and identify safety edge cases.

The same framework was used within the 3xD simulator at WMG, a 360° immersive, 'Drive-In Driver-in-the-Loop' facility capable of hardware-in-the-loop testing, connectivity research, and human factors studies (Mitchell, et al., 2025). Despite forming the backend of a semi-virtual semi-physical testing environment, this framework does not yet address the role of head-mounted display (HMD) based virtual reality (VR).

Immersive Virtual Reality for Driving Simulation

In parallel with the advancement of simulation based testing, there is growing interest in the use of immersive virtual reality for driving research. In this context, VR refers specifically to head-mounted display based immersive visualisation, as distinct from large screen, projection based, or CAVE systems. VR offers a level of spatial presence and perceptual engagement that is difficult to achieve with conventional desktop based simulators, where participants view the driving environment through a flat screen at a fixed distance. By placing the user inside the virtual environment with stereoscopic depth, head tracked viewpoints, and spatial audio, VR can improve the ecological validity of human-in-the-loop studies, which is the degree to which participant behaviour in simulation reflects real world behaviour.

This is particularly relevant for research involving automated driving, where the participant's role shifts from active driver to co-pilot or passive monitor. Evaluating how users perceive, trust, and respond to automation requires that the simulation convincingly conveys the experience of being inside a moving vehicle under automated control. Desktop simulators,

while effective for many engineering validation tasks, are limited in their ability to elicit the subjective and physiological responses, such as comfort, unease, or attentional engagement that are central to human factors evaluation.

The use of HMDs in driving simulation has been studied across several dimensions. Blissling and Bruzelius (2018) reviewed five years of driving simulation conference papers and found that, of 203 interactive driving studies identified, only ten used HMD based setups, though around 40% could feasibly have been conducted with HMDs. Subsequent work has examined the effects of HMDs on driving behaviour (Blissling, Bruzelius, and Eriksson, 2022), the validity of primary driving tasks in HMD simulators (Hartfiel and Stark, 2021), and differences in simulator sickness and sense of immersion between HMD and screen based configurations (Himmels, et al., 2023; Malone and Brünken, 2021). However, the HMD driving simulators described in these studies are typically standalone systems with bespoke scenario pipelines. There is limited work on integrating immersive VR as an interchangeable component within a broader, standards-aligned simulation framework, where the same scenario definitions, execution logic, and road network data are shared across VR and non-VR configurations.

Integrating VR into established simulation workflows also presents practical challenges. These include meeting the real time rendering demands of stereoscopic display, maintaining consistency with existing scenario definitions and execution pipelines, and ensuring that experimental results obtained in VR remain comparable with those from other simulation configurations. Addressing these challenges requires VR to be incorporated within a broader simulation architecture rather than developed as an isolated tool.

Research Question

This work addresses the following research questions:

1. How can a scalable, modular, and customisable simulation framework designed around common scenario and interface standards be extended to support immersive VR based driving simulation without compromising reproducibility or experimental control?
2. To what extent can such a framework support diverse human-in-the-loop use cases, including manual driving, co-piloting with automation, and user acceptance evaluation, within a unified simulation environment?

The underlying hypothesis is that VR should not be treated as a standalone simulator, but rather as a specific realisation of a broader simulation framework, in which immersive visualisation and interaction are integrated as interchangeable components. By adopting this approach, VR based experimentation can benefit from the same scenario definitions, execution logic, and system interfaces used across non-immersive simulation and other test configurations.

Methodology

VR System Integration

To support immersive human-in-the-loop experimentation, a virtual reality configuration was developed as an extension of the existing visualisation and interaction layers within the simulation framework. Rather than building a standalone VR simulator, the approach treats VR as an alternative instantiation of the framework's rendering and user interface components, as described by Mitchell, et al. (2024) and Mitchell, et al. (2025), preserving full compatibility with the underlying scenario definitions, execution logic, and road network data. This means that any OpenSCENARIO (ASAM e.V., 2022b) paired with an OpenDRIVE (ASAM e.V., 2022a) road network can be executed identically in VR and non-VR configurations, with no modification to the scenario files or execution pipeline.

The VR setup employs a Meta Quest 3 head mounted display which is connected to a laptop (for portability) via a wired link, ensuring low latency that is suitable for real time driving. A specific rendering configuration was created in order to meet the performance and quality requirements of VR, which are to render over 9 million pixels per frame and then have a stable framerate which doesn't dip below 72fps, meaning per second there would be over 600 million pixels to be rendered. Therefore, the configuration included transitioning to a forward rendering pipeline, disabling unreal engine geometry features not yet performant with VR rendering, adjusting anti-aliasing and post-processing settings to avoid visual artefacts in the headset, and tuning pixel density to balance clarity with frame rate stability. These adaptations were necessary to maintain the consistent frame rate aligned with the headset's native refresh rate, as frame drops in VR can cause discomfort and degrade the immersivity. Despite these VR related rendering changes, the visual content of the digital twins, such as road geometry, traffic participants, environmental assets, and signage, was required to remain consistent with the desktop simulation output, as both configurations draw from the same underlying scene data.

Map transitions between scenarios are handled using level streaming rather than full level reloads. This was particularly important given that the motorway map used in the event experience was a 10km route and so a sizable digital twin. Hence, this approach significantly reduces load times between scenarios and avoids breaking the participant's sense of immersion, as the next map can be loaded in within a couple of seconds.

Hardware Setup and Interaction Design

The physical setup, shown in Fig. 2, comprises of a Logitech G923 rig, including a force feedback steering wheel and pedal assembly mounted onto a folding out automotive style seat, providing tactile inputs for steering, throttle, and braking. The seating position, pedal reach, and wheel angle are adjustable to accommodate a range of participant body types, which is important for the wide range of ages who engaged in the experience. Paddle shifters on



Figure 2: VR driving simulation setup at a public engagement event

the steering wheel provided an alternative accelerate (left paddle) and brake (right paddle) method for participants who could not comfortably reach the pedals even when in their shortest configuration.

Within the virtual environment, the participant occupies a first person perspective from the driver's seat of the ego vehicle. A virtual steering wheel is rendered in the cockpit, aligned with the physical wheel, to reinforce spatial consistency between the real and virtual environments. The rotation of the virtual steering wheel is also synchronised with the physical steering wheel, which helps with the immersion when a participant turns the steering wheel. Spatialised vehicle engine audio is generated at runtime based on the vehicle speed, with sound characteristics adapting during acceleration, deceleration, and idle states. Other traffic participants also produce speed dependent engine sound, which contributes to the overall environment audio.

There is a VR specific user interface which is rendered in 3D space rather than the typical 2D UI used in non-VR applications. The interface design prioritises conciseness and clarity, using minimal text and iconography to avoid cognitive overload during the experience.

Modes of Operation and Vehicle Control

The VR configuration supports both manual and automated driving modes of vehicle operation within a single simulation session, enabling direct comparison of participant driving styles and their subjective experience of being inside an automated vehicle.

In manual mode, the participant has full control of the Ego vehicle through the physical steering and pedal inputs, as illustrated in Fig. 3. Vehicle dynamics, including acceleration response, braking behaviour, and steering sensitivity are modelled to provide a plausible driving experience, though the focus is on perceptual fidelity rather than high fidelity vehicle dynamics simulation. In automated mode, the Ego vehicle is driven by the scenario execution engine according to the trajectory and behavioural logic defined in the OpenSCENARIO file, while the participant ob-



Figure 3: Left eye view from within event experience on motorway scenario

serves from the driver's seat without active control. This mode allows participants to experience automated driving from a first person perspective, supporting assessment of comfort, trust, and situational awareness. Wheel animation and engine sound remain synchronised across both control modes to ensure behavioural and perceptual continuity throughout the experience.

A scoring system was developed to provide participants with performance feedback following manual driving scenarios. Metrics such as speed limit compliance, avoiding collisions and staying on the road are evaluated and presented as a summary at the end of the experience, supporting engagement and gamifying the experience, which was very well received by both children and adults who participated.

Scenario Design and Experimental Procedure

Scenarios used within the VR configuration are drawn from a curated library designed to cover a range of driving situations relevant to both human factors research and safety assessment. The scenario for the event experience was focused on a Motorway scenario and was developed in line with current regulatory frameworks for Automated Lane Keeping Systems (ALKS) (United Nations Economic Commission for Europe, 2021).

The motorway scenario incorporated traffic interactions such as car following, cut-in events from adjacent lanes, responses to decelerating lead vehicles, and encounters with stationary obstacles revealed by lane changes of preceding traffic. This is intended to showcase a range of situations that may face a driver on a motorway in the worst cases.

An experimental procedure was established and designed to be operable by a single facilitator and to

accommodate participants with no prior VR or driving simulation experience.

Participants are seated in the physical rig, the steering wheel position and pedals are adapted to the user and then the controls are explained to them, allowing them to physically test the controls. Then, they put the VR headset on and a calibration step is performed in which the participant looks straight ahead, allowing the system to align the virtual cockpit orientation with the participant's natural head position. Then they go into a tutorial scenario where they can practice using steering, throttle, and braking controls and can become accustomed to the visual environment and interaction mechanisms. The tutorial is followed by going into the motorway scenario as mentioned previously where they are tested in a set of tricky situations.

Throughout the session, an operator manages scenario progression via an external display which mirrors the participant's VR view, using a single input to advance menus and initiate scenarios on behalf of the participant. This separation of control allows the operator to pace the experience appropriately. For example, pausing between scenarios to check participant comfort, without requiring the participant to navigate complex menus while immersed. The operator can also reset to the tutorial or skip forward if needed. Following the immersive session, participant feedback is collected to support qualitative and subjective assessment of the experience.

Results and Discussion

The VR configuration was successfully integrated into the existing simulation framework as a modular extension, without requiring changes to the scenario execution pipeline, scenario definitions, or road network data. All OpenSCENARIO and OpenDRIVE assets used in desktop and 3xD simulator configurations (Mitchell, et al., 2025) were executed identically within the VR environment, confirming that immersive visualisation can be treated as an interchangeable rendering layer rather than a standalone simulation tool. This interoperability validates the modular design principle and ensures that results or observations obtained in VR can be meaningfully compared with those from other simulation configurations, supporting the hypothesis that VR should function as a specific instantiation of a broader simulation framework rather than an isolated system.

Deployment and Event Experience

The VR setup was deployed at multiple public engagement and demonstration events, including the Cheltenham Science Festival and Goodwood Festival (Fig. 4). Across these deployments, the system was operated by a single facilitator and accommodated a wide range of participants, including both children and adults with no prior experience in VR or driving simulation. The portable hardware configuration, comprising a laptop, VR headset, and folding steering rig, enabled rapid setup and teardown, with the system ready for use within 10-15 minutes of arrival.

The ability to reset or skip scenarios without breaking the VR session provided flexibility to adapt to in-



Figure 4: A young participant engaging with the VR driving experience at the Goodwood Festival.

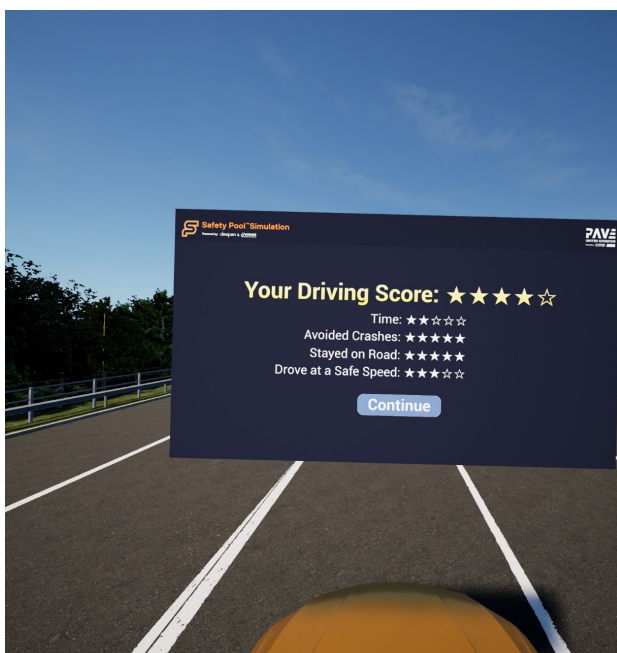


Figure 5: Left eye view of Driving score summary at end of event experience

dividual participant needs, such as providing additional practice time for younger users. The scoring system, which evaluated speed limit compliance, collision avoidance, and staying on the road, was particularly well received.

Participants across all age groups engaged with the gamified feedback, with the star based summary at the end of each session (Fig. 5) generating visible enthusiasm and repeat interest and some participants immediately got into the queue again. This suggests that performance feedback mechanisms can serve both as engagement tools for public outreach and as structured metrics for future research use.

Participant Observations

From a user experience perspective, immersive VR enhanced engagement and perceived realism com-

pared to desktop based simulation. Participants reported a strong sense of being inside the vehicle, particularly during automated driving sequences where they could observe traffic interactions from the driver's seat without active control. The dual mode operation, allowing participants to experience both manual driving and automated driving within the same scenario context, enabled direct qualitative comparison.

Discussion

The event deployments provided practical validation that the system is robust and repeatable in uncontrolled environments, where participants vary widely in age, technical familiarity, and physical characteristics. Stable frame rates at or above the headset's native 72fps refresh rate were maintained throughout all event sessions, with no participant seen to face any performance issues. The single facilitator operational model, combined with the adjustable hardware setup and simplified VR interface, proved sufficient to manage continuous participant throughput without technical failures or significant delays.

Limitations

It is important to note that the findings presented here are primarily observational, based on public engagement deployments rather than controlled experimental studies. No formal statistical analysis of participant behaviour, physiological responses, or subjective metrics such as trust, workload, or presence was conducted as part of these deployments. Participant exposure times were short, at two to three minutes per session, to suit event contexts, which limits the extent to which longer term effects such as sustained attention, learning, or simulator sickness can be assessed. While no participants reported significant discomfort, VR specific risks such as cybersickness, which has been shown to be more prevalent in head-mounted display setups than in screen based configurations (Dennison, Wisti, and D'Zmura, 2016; Malone and Brünken, 2021), cannot be ruled out and were likely mitigated by the brief session durations. The current work focuses on demonstrating the feasibility and readiness of the integrated VR system for structured human-in-the-loop research. Future work will build on this foundation by conducting controlled human factors studies with quantitative behavioural and subjective metrics collected during scenario execution.

Conclusions and Implications

This work demonstrates how a scalable, modular simulation framework built around common scenario and interface standards can be extended to support immersive VR-based driving simulation for human-in-the-loop use cases. While HMD-based driving simulators have been studied extensively (Blissing and Bruzelius, 2018; Blissing, Bruzelius, and Eriksson, 2022; Hartfiel and Stark, 2021), these are typically standalone systems with bespoke scenario pipelines. By contrast, the approach presented here treats VR as a configurable instantiation of the framework's rendering and interaction layers, preserving full compatibility with existing scenario definitions, execution

pipelines, and road network data (Mitchell, et al., 2024; 2025). This directly addresses the first research question, confirming that VR integration need not compromise reproducibility or experimental control.

The deployment at public engagement events validated the system's robustness and accessibility across diverse participant groups, demonstrating that the framework can support manual driving, observation of automated driving, and dual-mode comparison within a unified environment. This addresses the second research question, showing that the modular architecture supports a range of human-in-the-loop use cases within a single simulation configuration.

The implications of this work extend to both research and industry. For researchers, the approach offers a practical pathway for incorporating immersive VR into existing simulation workflows without fragmenting toolchains or duplicating scenario assets. For industry, the standards aligned architecture supports integration of VR based human factors evaluation into scalable validation pipelines alongside other simulation and testing activities.

Future work will build on this foundation by conducting formal human factors studies with quantitative behavioural and subjective metrics collected during scenario execution.

References

- ASAM, Apr. 2025. *ASAM OpenODD® 1.0.0*. Available at: https://publications.pages.asam.net/standards/ASAM_OpenODD/ASAM_OpenODD/latest/specification/00_preface/00_foreword.html.
- ASAM e.V., 2022a. *ASAM OpenDRIVE*. <https://www.asam.net/standards/detail/opendrive>.
- 2022b. *ASAM OpenSCENARIO*. <https://www.asam.net/standards/detail/openscenario-xml>.
- 2024. *ASAM OSI*. <https://www.asam.net/standards/detail/osi/>.
- Blissing, B. and Bruzelius, F., 2018. Exploring the Suitability of Virtual Reality for Driving Simulation. In: *Driving Simulation Conference 2018 Europe*. Available at: <https://proceedings.driving-simulation.org/proceeding/dsc-2018/exploring-the-suitability-of-virtual-reality-for-driving-simulation>.
- Blissing, B., Bruzelius, F., and Eriksson, O., 2022. The Effects on Driving Behavior When Using a Head-mounted Display in a Dynamic Driving Simulator. *ACM Transactions on Applied Perception*, 19(1), pp. 1–18. <https://doi.org/10.1145/3483793>.
- BSI, 2025. *PAS 1883:2025 Operational design domain (ODD) - Implementing BS ISO 34503*. Available at: <https://www.bsigroup.com/en-GB/insights-and-media/insights/brochures/operational-design-domain-implementation-of-bs-iso-34503/>.
- Dennison, M. S., Wisti, A. Z., and D'Zmura, M., 2016. Use of physiological signals to predict cybersickness. *Displays*, 44, pp. 42–52. <https://doi.org/10.1016/j.displa.2016.07.002>.
- Hartfiel, B. and Stark, R., 2021. Validity of primary driving tasks in head-mounted display-based driving simulators. *Virtual Reality*, 25, pp. 819–833. <https://doi.org/10.1007/s10055-020-00496-w>.
- Himmels, C., Andreev, V., Syed, A. A., Lindner, J., Denk, F., and Riener, A., 2023. Are Head-mounted Displays Really Not Suitable for Driving Simulation? A Comparison with a Screen-Based Simulator. In: *IEEE Intelligent Vehicles Symposium (IV)*. IEEE. <https://doi.org/10.1109/IV55152.2023.10186592>.
- ISO, 2023. *ISO 34503:2023 – Road vehicles – Test scenarios for automated driving systems*.
- Kalra, N. and Paddock, S. M., Dec. 2016. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94, pp. 182–193. ISSN: 0965-8564.
- Malone, S. and Brünken, R., 2021. Hazard Perception, Presence, and Simulation Sickness—A Comparison of Desktop and Head-Mounted Display for Driving Simulation. *Frontiers in Psychology*, 12, p. 647723. <https://doi.org/10.3389/fpsyg.2021.647723>.
- Mitchell, J., Baker, P., Chodowicz, E., Zhang, X., Khastgir, S., and Jennings, P., 2024. A Scalable Unreal Engine Based Testing Platform for the ASAM OpenX Standards and Safety Pool™ Scenarios. In: *Driving Simulation Conference 2024 Europe*. Strasbourg, France. Available at: <https://dsc2024.org/a-scalable-unreal-engine-based-testing-platform-for-the-asam-openx-standards-and-safety-pooltm-scenarios>.
- 2025. 3xD Re-Imagined: A Configurable and Scalable Cross Domain Simulator. In: *Driving Simulation Conference 2025 Europe*. Stuttgart, Germany. Available at: <https://wrap.warwick.ac.uk/id/eprint/193375>.
- SAE, Apr. 2021. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles.
- Safety Pool, 2024. *Safety Pool Scenario Database*. <https://www.safetypool.ai>. Accessed: 2026-04-13.
- United Nations Economic Commission for Europe, 2021. *UN Regulation No. 157 – Automated Lane Keeping Systems (ALKS)*. UN ECE.
- Zhang, X., Khastgir, S., and Jennings, P., Apr. 2023. An ODD-Based Scalable Assurance Framework for Automated Driving Systems. *SAE Technical Papers*. ISSN: 0148-7191.

Development Environment for the Validation of Steer-by-Wire Systems

Dr. Alexander Ahlert¹, Martin zur Heiden¹, Philipp Schmidt¹, Henning Kemper¹, Thomas Karasiewicz¹
Guillaume Miller¹

(1) IPG Automotive, e-mail: {alexander.ahlert, martin.zurheiden, philipp.schmidt, henning.kemper, thomas.karasiewicz, guillaume.miller}@ipg-automotive.com

Abstract – Steer-by-wire systems represent a paradigm shift in the development of steering systems. Although the elimination of a mechanical connection opens up new degrees of freedom for design and functional layout, it requires comprehensive validation of safety, ride comfort, and system integration. With a specially developed test bench and the virtual full-vehicle simulation CarMaker, IPG Automotive offers a test system that holistically addresses these challenges.

Keywords: Steer-by-wire systems, virtual full-vehicle simulation, system integration, vehicle development, system validation.

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

The introduction of Steer-by-Wire (SbW) systems marks a fundamental technology leap in vehicle development. Unlike conventional steering systems with a rigid mechanical connection in the form of a steering column between the steering wheel and the wheels, as well as a fixed steering ratio, SbW systems open up completely new prospects. This particularly applies to the targeted tuning of steering feel and steering effort, lateral vehicle dynamics, and for automated driving (Pfeffer and Harrer, 2013; Wesche, 2025). In addition, they enable innovative control concepts and driver interfaces, for instance, in the form of joysticks or alternative input devices (Kirihaara, 2025). However, this technological freedom entails growing requirements for safety, reliability, and authenticity of steering feel as well.

To enable appropriate calibration and validation of these SbW systems, an innovative test bench concept has been developed. It merges the core elements of SbW technology – Electronic Control Units (ECUs), software and control functions, signal transmission, as well as sensors and actuators – in an integrated development and test environment and makes them available for system validation.

Product Solutions

In an SbW system, the mechanical connection is replaced by sensors, ECU algorithms, and electrical actuators (Pfeffer and Harrer, 2013). This enables targeted adjustment of the steering responses to different driving situations, such as reduced steering effort while parking, greater precision at high velocities, or targeted damping of external disturbances on the steering system. Two architectures are typically used for this purpose, FIGURE 1. The first architecture still uses a steering rack, which is controlled via a central actuator with a related ECU – henceforth referred to as power pack. The second architecture uses wheel-individual power packs connected to the wheels via a transmission and a tie rod. Both architectures have various advantages and disadvantages that will not be explained in greater detail in this article (Pfeffer and Harrer, 2013; Nievelstein et al., 2023).

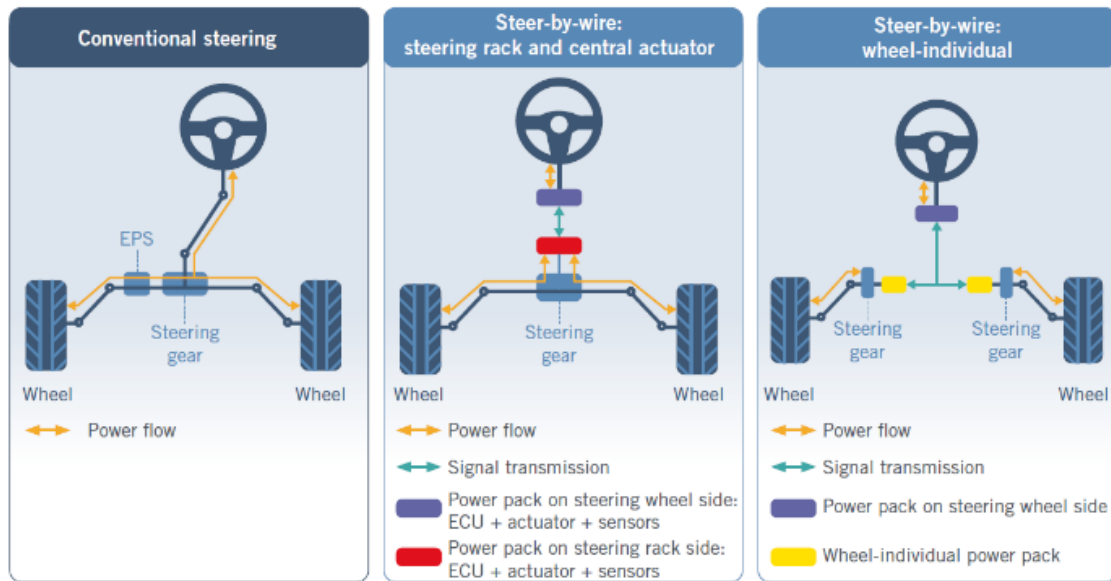


Figure 1: Examples of architecture and operating principle – conventional steering (left) compared to the two SbW systems central power pack (center) and wheel-individual power pack (right) (EPS = Electric Power Steering; ECU = Electronic Control Unit)

A specially developed test bench is used to enable testing of SbW systems under realistic conditions in a laboratory. It links the major control elements of the SbW system – the physical power packs including signal transmission – with simulation models representing further components such as the steering rack and the steering wheel and (if necessary) the total vehicle's behavior in a virtual environment. The composition of the test bench varies, depending on the SbW configuration, but the core principle remains identical. Core components of the SbW test bench are torque actuators (servo motors) connected to the power packs of the physical SbW system with torque sensors in between. This applies to both the composition on the side of the steering wheel/ driver and the part on the

side of the wheel and tire. The number of power packs may vary depending on the architecture, enabling individually wheel-controlled front and rear steering as well (Pfeffer and Harrer, 2013). The test bench actuators can be operated based on command signals or act as the interface to the virtual environment.

A central test bench computer (Xpack Real-time System from IPG Automotive as an Hardware-in-the-Loop (HiL) system) controls the actuators on the test bench side, processes the measured data, handles the bus communication with the SbW system and the test bench hardware, and performs the full-vehicle simulation including real-time synchronization.

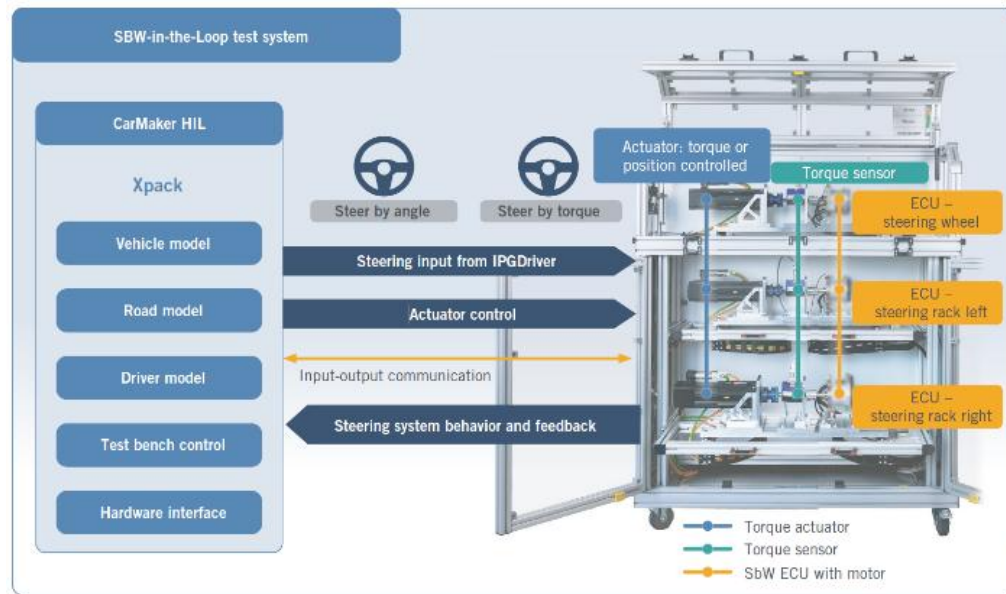


Figure 2: Concept of the SBW-in-the-Loop test system

While the test bench realistically simulates individual steering components the full benefit only develops in its linking with a full-vehicle simulation environment such as CarMaker (Tosolin, 2020). In that way, system tests and system analyses can be integrated in highly complex, realistic driving situations, and interactions with all relevant subsystems can be considered.

Methodology

Test system architecture: Using a SbW configuration with three power packs and a central steering rack as an example, the event chain in the interaction with the simulation environment is explained. The test bench can be actuated in two ways: either by a specified steering angle (steer-by-angle) or a steering torque (steer-by-torque). A virtual driver assumes control. The physical properties of the steering wheel and steering column, such as inertia, damping, and friction, are simulated, resulting in a test bench actuation that excites the physical power pack with torque or a rotational angle. The power pack provides a steering response torque depending on the simulated vehicle response and the SbW functional logic. A torque sensor captures the resulting torque that is fed back to the test bench control system, FIGURE 2.

The ECUs of the SbW system communicate with each other: The power pack on the steering wheel side transmits a response signal to the power packs on the steering rack. In a physical vehicle they would subsequently generate a torque acting on the steering rack via the steering gear, thus moving the wheels.

From the power packs onward, the entire further event chain – including the steering gear, steering rack, wheel suspension, tires, and the full-vehicle response – is virtually represented on the test bench. In the simulation, the torques measured on the test bench between the test bench actuators on the side of the steering rack and the power packs are converted via a virtual steering gear into forces acting on the simulated steering rack.

Simulation models considering the relevant forces and torques are used to calculate the motion of the steering rack. They take into account mass and inertia effects, friction and damping models, and restoring forces and torques from the tire and the overall vehicle behavior. This results in steering rack acceleration, speed, and position that are fed back in real time to the test bench control system. The test bench actuators on the steering rack side subsequently adjust the resulting angle at the power pack by taking the virtual steering gear into account. This creates realistic feedback to the SbW system with a closed control loop and a consistent effect chain.

Test coverage: This approach enables testing and validation of the SbW system on the full-vehicle level under realistic conditions, with all relevant interactions between the driver, steering system, wheel suspension, tires, and the vehicle behavior being retained. In that way, comfort, dynamics, and safety can systematically be optimized in early development stages, the SbW software and signal transmission be comprehensively validated, and various vehicle and steering configurations be flexibly tested.

For simpler tests such as plausibility checks or basic functionality tests, the test bench actuators can be

operated individually and signal-based. In addition, using a fail-safe tester, selected actuators, sensors, or internal data lines, can systematically be physically separated. This makes it possible to simulate various operating conditions, failure modes, and failure scenarios, and to evaluate the behavior of the SbW system in an automated way. Consequently, the test bench concept shown enables reliable verification and validation of the demanding requirements to be met by modern steering system architectures.

Driver-in-the-Loop: The Steer-by-Wire Test System is extended through the integration of a remote steering wheel, hence enabling DiL testing and subjective assessment of the steering response. This is achieved through a high-resolution steering device, effectively sensing the Driver's steering angle, and reproducing the steering torque calculated by the vehicle model. The communication between the real-time target and the steering actuator is synchronized with the simulation loop at 1 kHz, ensuring low-latency torque feedback. Provided that the vehicle dynamics model is well correlated, the system enables subjective evaluations from the early stages of software development, supports rapid short-loop validation, and facilitates the initial calibration of steering functions.

Results and discussion

A relevant customer scenario illustrates the high-level requirements to be met by SbW systems: Because there is no mechanical connection between the steering wheel and the steering rack and the steered road wheels, respectively, a suitable system initialization and synchronization of the power packs is necessary in terms of the vehicle's startup behavior. In the test, the SbW system's behavior is investigated during the starting phase after the driver turned the steering wheel while the vehicle was shut down.

A related acceptance test could look like this: After the vehicle has been parked, the position of the steering wheel is changed. When the start button is pushed again the system must ensure that the positions of the steering wheel and the steering rack match again. To do so, the power pack on the steering wheel side turns the steering wheel back into the correct position before control is handed over to the driver. On the test bench, this behavior can be tested and validated in an automated and reproducible process. FIGURE 3 shows the possible progression of time in such a test.

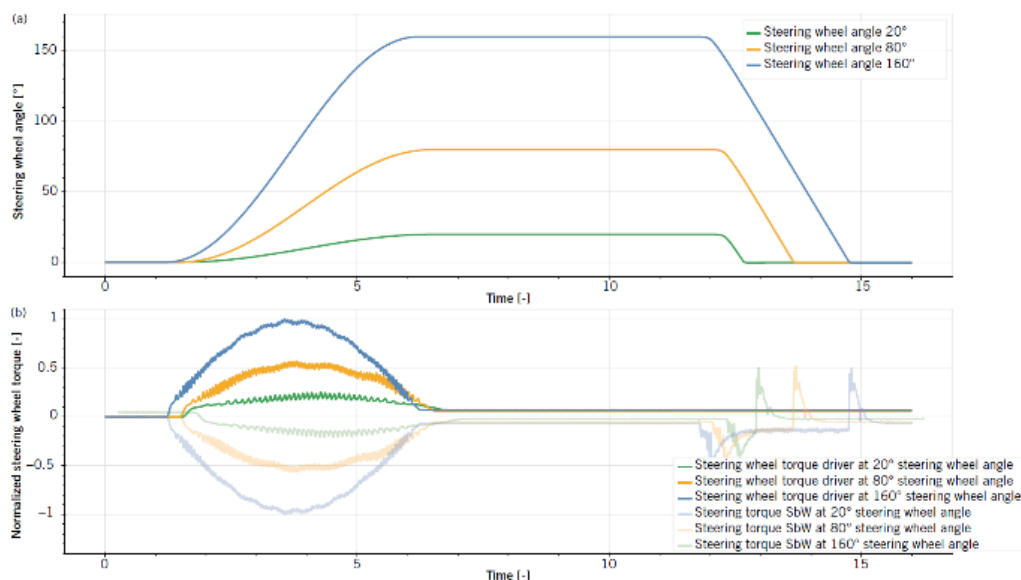


Figure 3: Startup behavior of an SbW system for the steering wheel angle (a) and the steering wheel torque (b) in comparison to a rigid mechanical connection; in the left section, the human driver steers the vehicle (up until approximately second 7), on the right, the SbW system is in control

In accordance with the scenario, after the simulated vehicle has been deactivated, the test bench actuator sets three steering wheel angles (20°, 80°, 160°). After about 12 s, the vehicle is activated again.

As part of the startup behavior, the SbW system provides a suitable torque. The measured torques and steering wheel angles show that the power pack returns the steering wheel to the center position

without intervention by the test bench actuator while the steering rack retains its position. Because the SbW system shows the expected behavior, all three tests were passed successfully.

Additionally, extended situations can be investigated that may occur in real-world use or in the case of operating errors – for instance, if the driver intervenes during the synchronization process. The large number of possible combinations and variants, even during seemingly simple processes such as the startup behavior, illustrates the increasing testing requirements for SbW systems.

Aside from specific validation scenarios, classic driving maneuvers (open- and closed-loop) to evaluate steering behavior and vehicle characteristics (Pfeffer and Harrer, 2013; Meyer-Tuve, 2008; Ersoy and Stefan, 2017) can be performed on the SbW test bench. By way of example and based on a non-validated example model, FIGURE 4 shows the open-loop maneuver “sine with dwell” according to ISO 19365 (but only up to a maximum lateral acceleration of approximately 3 m/s^2 in order to stay in the linear lateral dynamics operation range) in which the virtual driver executes defined steering wheel angle inputs. These ones are adjusted via the test bench actuator on the steering wheel side.

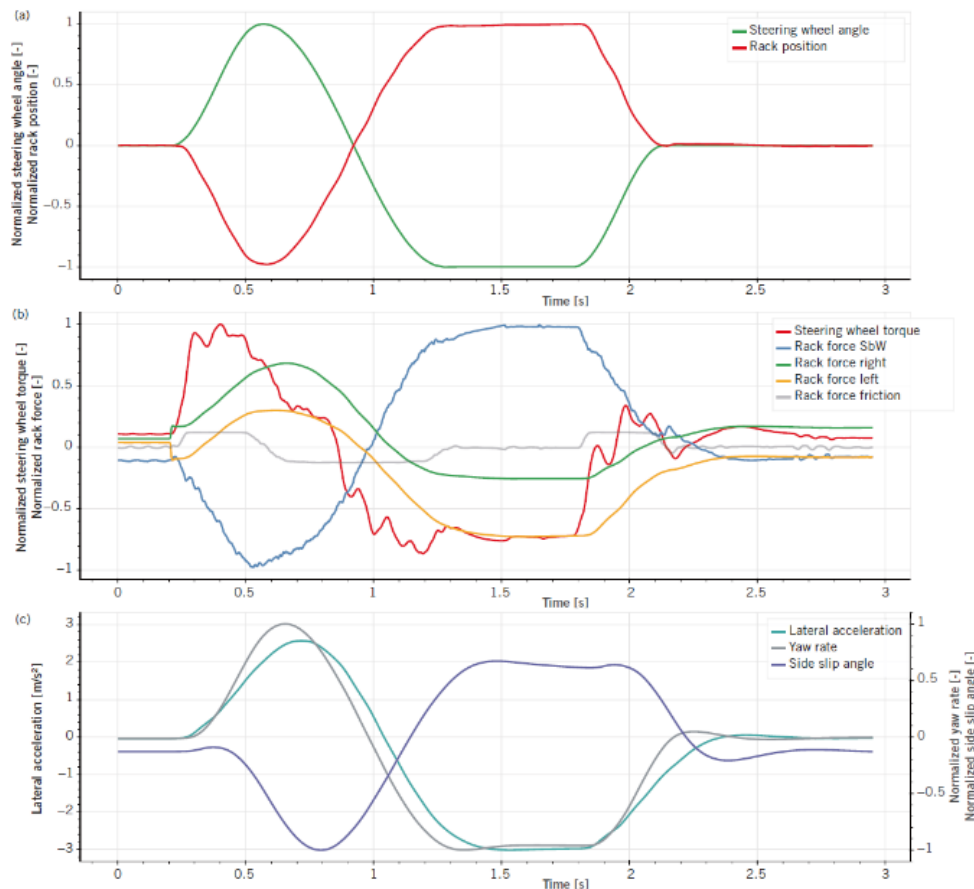


Figure 4: Sine-with-dwell test according to ISO 19365, plotted for steering wheel angle (a), steering wheel torque (b), and lateral acceleration (c)

Moreover, with the help of the CarMaker simulation environment, closed-loop maneuvers such as double lane changes according to the ISO 3888-1 standard can also be tested by a virtual driver model

in a virtual environment. FIGURE 5 (a) to (c) shows an exemplary result; the representation of the variables corresponds to FIGURE 4.

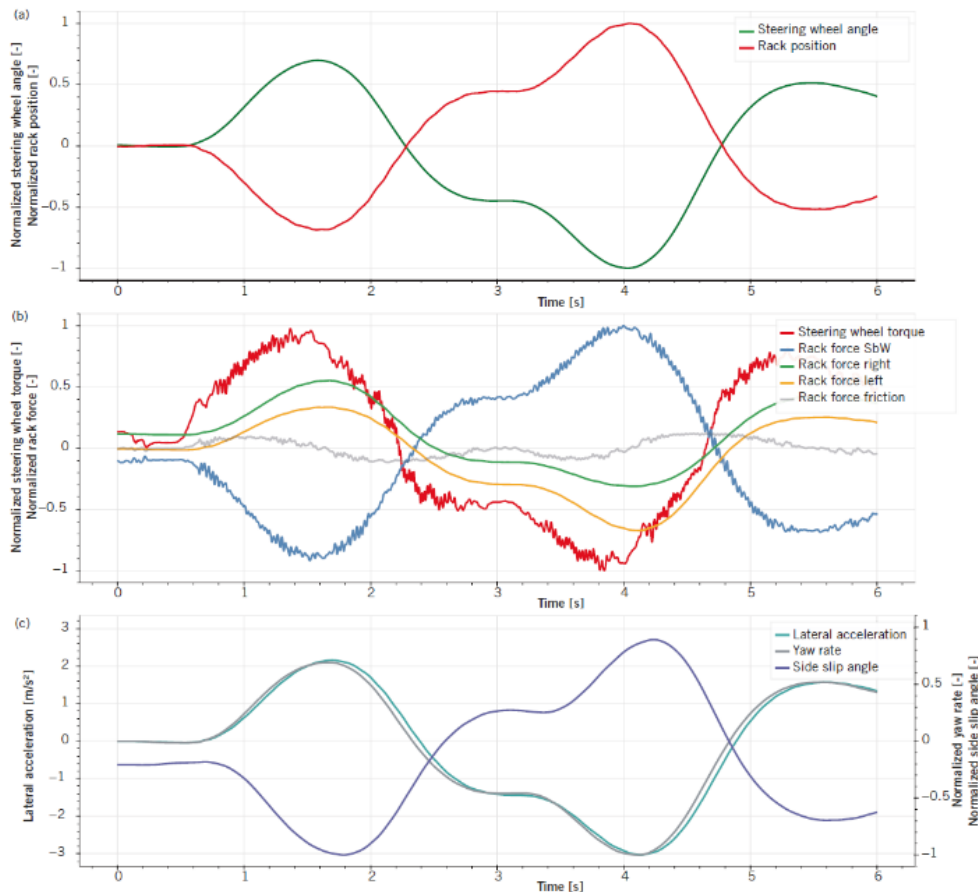


Figure 5: Double lane change test according to ISO 3888-1

Conclusions/implications

The presented integration of a steer-by-wire test bench with the CarMaker full-vehicle simulation environment provides a comprehensive platform for the development and validation of software-defined steer-by-wire systems. By combining hardware testing with realistic vehicle-level simulation, the solution enables repeatable, representative, and efficient validation from the earliest stages of development.

The platform enables progressive validation of the steer-by-wire motor control units and their embedded software within a full-vehicle simulation environment. By virtualizing the remaining steering components, it allows system-level behaviour to be assessed without requiring a complete physical steering system. In combination with holistic test strategies, it enables continuous execution of unit, integration, fail-safe, system, and acceptance tests while significantly reducing the dependence on physical vehicle prototypes.

The integration of a remote torque-feedback steering actuator further extends the platform by enabling Driver-in-the-Loop evaluations and subjective assessment of steering feel. Engineers can therefore combine objective performance metrics with human

perception in a deterministic and completely safe environment, including extreme driving manoeuvres and fault scenarios that would be impractical or unsafe to reproduce on a test track.

The proposed architecture therefore strikes an effective balance between purely virtual simulation and hardware-intensive validation approaches. By limiting the physical hardware to the steer-by-wire motor control units while virtualizing the remaining steering components, it enables early, repeatable and scalable validation while reducing both development time and overall validation costs compared with approaches relying exclusively on vehicle prototypes or complete steering test benches.

References

- Ersoy, M., and Stefan, G. (eds.), 2017. *Fahrwerkshandbuch: Grundlagen – Fahrdynamik – Fahrverhalten – Komponenten – Elektronische Systeme – Fahrerassistenz – Autonomes Fahren – Perspektiven*. Springer-Verlag, Wiesbaden.
- Kirihara, K., 2025. Steer-by-Wire – Future Operating System instead of a Steering Wheel. In: *ATZworldwide* 6/2025, pp. 28-33
- Meyer-Tuve, H., 2008 *Modellbasiertes Analysetool zur Bewertung der Fahrzeugquerdynamik anhand von*

- objektiven Bewegungsgrößen. Dissertation, Munich, University of Technology.
- Nivelstein, M., Sandmann, T., Olschewski, I., and Wegener, D., 2023. Steer-by-Wire – Rethinking the steering system to realize improved performance, layout and comfort. Whitepaper, Online: https://fka.de/images/Steer_by_Wire_Whitepaper.pdf, access: September 30, 2025
- Pfeffer, P., and Harrer, M., 2013. Lenkungshandbuch. Lenksysteme, Lenkgefühl, Fahrdynamik von Kraftfahrzeugen. Wiesbaden, Springer Vieweg.
- Tosolin, G., 2020. Fitting ESP hydraulic parameters using CarMaker for Simulink and optimization algorithm. Lecture, Apply & Innovate, Tech Weeks, Karlsruhe, 2020
- Wesche, M., 2025. Development of dependable Steer-by-Wire steering systems. Dissertation, Clausthal, University of Technology.

A Distributed Driving Simulation Framework utilizing Data Distribution Service

Christian Bauer¹, Chris Schwarz¹

(1) Driving Safety Research Institute, University of Iowa, 2401 Oakdale Blvd, Coralville, IA 52241, USA, e-mail: {christian-bauer, chris-schwarz}@uiowa.edu

Abstract – Modern human-in-the-loop driving simulators for research must adapt rapidly to evolving needs, enabling integration of unconventional functionality, modification of existing behavior, and connection to external simulation tools—while providing a robust, scalable, and real-time-capable foundation. Legacy architectures based on fixed data layouts or proprietary synchronization hardware make such evolution increasingly difficult.

This paper presents DSRISimConnect, a distributed simulation framework designed to address these challenges by structuring subsystem interaction around shared simulation state and scene information. The framework allows shared data structures and dynamic scene entities to be defined, modified, and extended at runtime, enabling incremental adaptation without disrupting existing components. Coordinated execution is achieved through a global simulation time base and frame-based scheduling, while communication and synchronization are handled transparently over standard networking hardware, utilizing the Data Distribution Service (DDS) for high-performance and scalable data distribution.

DSRISimConnect has been integrated into a large-scale driving simulator through staged migration from a proprietary shared-memory architecture and is used in distributed rendering, multi-simulator experiments, and real-time co-simulation with external engines. Results demonstrate that DSRISimConnect provides a real-time-capable, flexible, and extensible platform that supports a wide range of current and future driving simulation research.

Keywords: Human-in-the-loop, Distributed simulation.

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

Human-in-the-loop driving simulators used in research are evolving systems that must balance strict real-time requirements with continual change. New experimental designs, driving scenario capabilities, sensor concepts, rendering capabilities, and co-simulation partners regularly require modifications to simulator subsystems and to the information exchanged between them, while preserving timing predictability and operational stability.

Various simulation platforms and standards support distributed deployment and, in many cases, real-time operation, including federation-based approaches such as IEEE 1516 High-Level Architecture (HLA), robotics-oriented middleware such as ROS-based systems (Macenski, et al., 2022), simulation

environments built around game engines or autonomous-driving toolchains (Carla, ASAM Open Simulator Interface, AutoWare), and commercial driving simulation platforms. These standards and systems typically define fixed communication patterns and data layouts. Extending them to accommodate experiment specific functionality or communication with new subsystems is often difficult and time consuming; or in case of proprietary platforms at times not feasible. When dynamic, heterogeneous, or experiment-specific data structures must be introduced over time, these architectures become increasingly restrictive.

For human-in-the-loop research simulators, the central challenge is therefore not only providing standard capabilities but also allowing subsystems, shared data, and experiment configurations to evolve quickly within an operational system with real-time requirements.

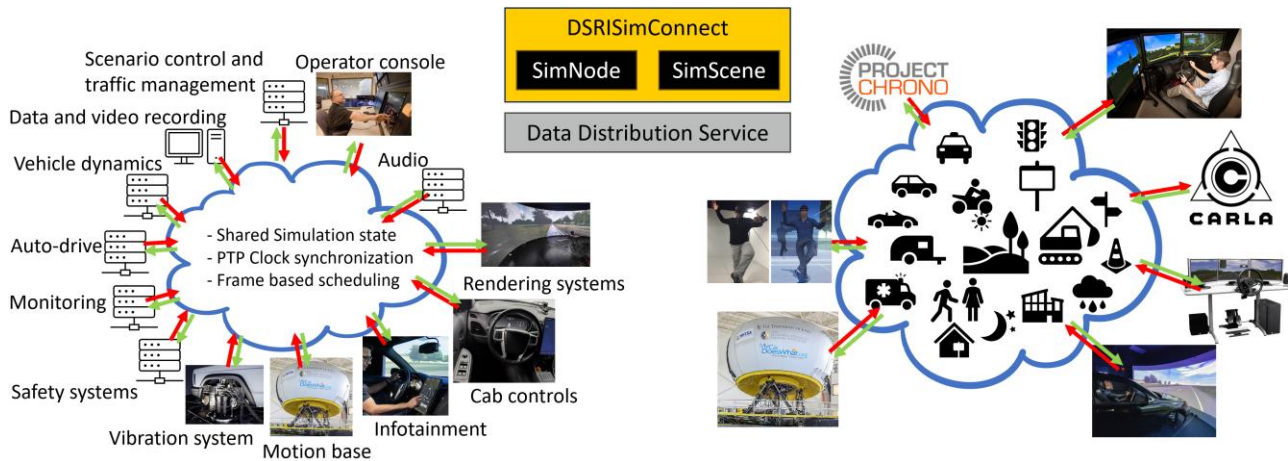


Figure 1 *DSRISimConnect* provides a distributed real-time simulation framework that coordinates modular components around shared state and scene information. *SimNode* (left) handles access to shared state information and coordinated execution of time-critical tasks, while *SimScene* (right) enables multiple simulators or software components to synchronize dynamic scene objects within a shared virtual environment. The framework builds on the Data Distribution Service (DDS) to provide high-performance, scalable data exchange.

To address these challenges, we developed *DSRISimConnect*, a distributed simulation framework intended as a practical foundation for modernizing the NADS-1, NADS-2, and miniSim™ simulators at the Driving Safety Research Institute (DSRI) (Schwarz, et al., 2023). Its goal is to enable incremental adaptation and rapid experimentation within operational simulators while preserving predictable real-time behavior.

The remainder of this paper presents an overview of *DSRISimConnect* before describing its core components in more detail. This is followed by a staged migration from the legacy shared-memory network architecture and performance results from the deployed system. Finally, representative deployment examples and co-simulation integrations are presented.

DSRISimConnect Overview

DSRISimConnect provides a unified framework for integrating, extending, and coordinating distributed subsystems in real-time driving simulators. Subsystem interaction is organized around shared simulation state and scene information, enabling new functionality and experiment-specific extensions to be introduced quickly without disrupting existing components or predictable real-time behavior.

At the communication level, *DSRISimConnect* employs a data-centric communication model in which shared simulation state and dynamic scene entities can be defined, modified, and extended, even at runtime. Extensions are scoped to participating subsystems, allowing components that

do not depend on additional information to continue operating unchanged.

For coordinated execution, the framework provides a global simulation time base that enables predictable, frame-based scheduling across subsystems and machines. Execution timing, data time-stamping, and synchronization are handled consistently, enabling shared state and scene information to be exchanged and synchronized reliably across participating components without reliance on proprietary synchronization hardware.

Together, these capabilities form a lightweight, framework that supports incremental modernization of existing DSRI simulators and a wide range of deployment scenarios, including distributed subsystem execution, distributed rendering, multi-simulator experiments, and real-time co-simulation with external simulation engines.

DSRISimConnect is implemented primarily in C++ to align with existing development environments and performance requirements, with minimal dependencies limited to DDS (RTI Connex and Fast DDS supported), and optional Python and Node.js bindings that expose selected functionality for low-frequency control and rapid prototyping, while time-critical execution remains in native code.

Communication Model and Runtime Extensibility

DSRISimConnect employs a data-centric communication model in which simulator subsystems interact by declaring and consuming shared simulation state information rather than

exchanging point-to-point messages. This approach decouples subsystem implementation from communication topology and allows simulator functionality to evolve without requiring invasive interface changes.

Communication is realized using the Data Distribution Service (DDS) (Object Management Group (OMG), 2015), which provides decentralized publish–subscribe data distribution over standard networking hardware. *DSRISimConnect* maps application-level shared state information and update semantics to DDS topics with appropriate quality-of-service policies, allowing shared state to be distributed predictably without requiring subsystem-specific middleware configuration. This design also provides a natural foundation for interoperability with other DDS-based systems.

Runtime-Extensible Data Structures

To support incremental evolving of shared simulation state information, *DSRISimConnect* allows the structure of communicated data to be defined and extended at runtime. Shared simulation state is described using *Variables*, which specify the structure and organization of data fields, and *Values*, which represent the corresponding runtime content. Both structure definitions and data updates are communicated through the framework, allowing participating subsystems to synchronize on shared layouts agreed on during initialization.

Data structures may be extended dynamically by adding new *Variables*, either globally or for individual object instances. Subsystems that do not depend on these extensions can continue operating unchanged, while participating components can make immediate use of the additional information.

For example, a shared vehicle representation may define a common set of *Variables* for pose and control state. Individual vehicle instances can be augmented with additional attributes—such as experiment-specific annotations—without modifying the base definition or affecting subsystems that require only the standard attributes.

Data Synchronization Semantics

Shared simulation data in *DSRISimConnect* are classified according to their synchronization semantics:

- *Static* data are immutable and communicated once (e.g., identifiers or configuration parameters).
- *Periodic* data are updated every simulation frame and prioritize low latency over guaranteed delivery, making them suitable for continuously changing state.
- *Conditional* data are transmitted only when values change and require reliable delivery, typically for discrete states or events.

The distinction between periodic and conditional data reflects a deliberate trade-off between low-latency, frame-synchronous updates and reliable transmission of semantically significant changes, allowing real-time subsystems to prioritize timely execution while ensuring correctness of critical events.

SimNode

In *DSRISimConnect*, a *SimNode* represents a simulator subsystem integrated with the framework and participating in coordinated execution. Each *SimNode* encapsulates the execution logic of a single subsystem—such as vehicle dynamics, motion cueing, rendering, audio, or scenario control—and connects it to the framework’s services for shared state access, timing, and lifecycle coordination.

SimNodes operate on shared simulation state as defined by the communication model described in the previous section. Subsystems declare which shared state information they provide and consume, while the framework manages distribution and synchronization. This allows *SimNode* implementations to focus on subsystem logic, with communication and data consistency handled by *DSRISimConnect*.

Global Time and Frame-Based Scheduling

To enable coordinated real-time execution across multiple *SimNodes*, *DSRISimConnect* employs Frame-Based Scheduling (FBS) driven by a shared global simulation time. All *SimNodes* advance in lockstep using fixed-duration simulation frames, ensuring temporally consistent execution across distributed subsystems.

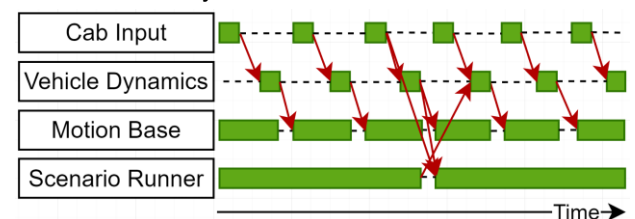


Figure 2: Frame-based scheduling in *DSRISimConnect*, showing time-synchronized execution windows across distributed *SimNodes* and data exchange aligned to the global simulation clock.

Accurate time coordination across machines is essential for FBS, and *DSRISimConnect* therefore provides a dedicated global simulation time service that synchronizes participating *SimNodes* using a software-based mechanism based on the Precision Time Protocol (IEEE1588, 2019). Nodes exchange time information, estimate network latency, and continuously adjust local clocks to a common reference, providing a consistent notion of simulation

time across machines over standard networking hardware.

Lifecycle and Simulator State and Management

SimNodes participate in simulator-level lifecycle and simulator state management.

The lifecycle management governs simulator independent initialization, configuration, activation, deactivation, and shutdown of *SimNodes*.

The configurable simulator state management controls experiment stages, including simulator- and hardware-specific preparation and shutdown sequences. *SimNodes* not involved in a particular stage simply acknowledge completion, allowing heterogeneous subsystems to participate without enforcing uniform behavior.

SimScene

SimScene provides an abstraction for representing and synchronizing dynamic entities in the simulated environment—such as vehicles, pedestrians, traffic signals, and other scene elements—across distributed simulators and co-simulation components.

Scene entities are represented as *SceneObjects* composed of property groups, whose values are described using the same runtime-extensible *Variables* and synchronization semantics introduced in the communication model. This allows scene-level actor sets and attributes to evolve, which is difficult to achieve with fixed data layouts that assume static data structures and actor sets.

Multiple participants can share objects they control while receiving synchronized representations of objects controlled by others. Scene updates are time-stamped allowing for time synchronization and dead-reckoning between participants.

Together, these mechanisms provide a shared temporally consistent representation of the simulated environment, forming the basis for distributed simulation and real-time co-simulation scenarios discussed in later sections.

Migration from SCRAMNet to DSRISimConnect

The NADS-1 driving simulator at DSRI is a long-running, operational research facility that historically relied on a SCRAMNet Network-shared-memory architecture (Curtiss-Wright Corporation, 2015) to achieve low latency and predictable real-time behavior between subsystems. While traditional shared-memory architectures excel at low-latency exchange of predefined control data,

they become increasingly restrictive when dynamic, heterogeneous, or experiment-specific data structures must be introduced incrementally.

Modernization therefore required a migration approach that preserved real-time guarantees and operational availability while allowing incremental adaptation. A full replacement of the legacy architecture all at once was neither feasible nor desirable due to ongoing research operation and safety-critical hardware dependencies.

DSRISimConnect was deployed alongside the existing SCRAMNet infrastructure, enabling staged migration at the subsystem level. A temporary bridge allowed legacy shared-memory data to be mapped to the framework's shared state model, allowing legacy and migrated subsystems to coexist during the transition.

Through this process, the majority of in-house developed simulator subsystems (approximately 80%) were migrated to operate as *SimNodes* using *DSRISimConnect*. SCRAMNet hardware remains in use for the motion base and its associated safety systems developed by external contractors. The resulting hybrid architecture preserves real-time and safety characteristics while enabling distributed deployment, runtime-extensible shared state, and integration of new components using standard networking hardware.

Technical Validation and Performance

To ensure that *DSRISimConnect* meets the operational demands of full-scale distributed research simulators, clock synchronization accuracy and shared-state data transfer latency were evaluated, as these metrics are critical for frame-based scheduling.

Experiments were performed on the NADS-1 simulator, which was migrated from a SCRAMNet-based shared-memory architecture, at several stages of the migration process. The fully migrated system consisted of 24 *SimNodes*, running in a distributed system with 18 machines connected via standard Gigabit Ethernet. Main Components are illustrated in Fig. 1. Most *SimNodes* participating in frame-based scheduling operate at 240Hz, while selected subsystems operate at 120Hz or 60Hz. Using *SCRAMNet* communication, with its known sub-microsecond latency, as a reference, data were recorded during simulations and analyzed.

Clock synchronization remained within <20 μ s, using the software-based time service. Periodic shared state updates were delivered with typical latencies of ~500 μ s, and throughput for large-scale scene updates - 1000 moving vehicles - was ~200 μ s. These measurements represent end-to-end

application-layer timings, including *DSRISimConnect*, DDS middleware, OS socket handling, network drivers, and physical network transmission.

These results demonstrate that predictable real-time behavior can be achieved using commodity Gigabit Ethernet, while simultaneously enabling greater architectural flexibility, support for co-simulation, and simplified subsystem maintenance.

Application and Deployment Examples

DSRISimConnect has been deployed in a range of simulator configurations and co-simulation environments, demonstrating its applicability to heterogeneous, real-time simulation workflows and its suitability for incremental modernization of operational systems.

Distributed Rendering with Unreal Engine

DSRISimConnect was used to integrate a distributed rendering system based on the Unreal Engine into the driving simulator. Using shared scene state provided by *SimScene*, rendering components receive time-stamped updates of dynamic scene entities and simulator state, enabling consistent visualization across multiple rendering nodes. This replaced custom TCP/UDP communication solutions and simplified the integration of additional views and display configurations without requiring changes to other simulator subsystems.



Figure 4: NADS-2 simulator with rendering for front and back views, side mirrors and backup camera used in multi driver interaction study alongside NADS-1 simulator.

Multi-Simulator Driver Interaction

To support studies involving interaction between multiple drivers, *DSRISimConnect* was used to couple independent driving simulators into a shared scenario. One simulator was a full-scale distributed installation (NADS-1), while the second was a single-PC based simulator (miniSim™). Each simulator contributed the actors it controlled and received synchronized representations of actors controlled by

other participants via *SimScene*. This ensured a consistent view of all actors across systems, enabling multi-driver experiments without centralized control logic or simulator-specific synchronization mechanisms.

Real-time Co-Simulations

As representative examples of integration with external simulation engines, we realized real-time co-simulations of the NADS-1 driving simulator with Chrono and Carla using *DSRISimConnect*.



Figure 5: Co-simulation involving simulator cab, motion and visualization system (DSRI), simulated traffic and LIDAR sensor data (Chrono), object detection and LIDAR data visualized with Foxglove (ROS2).

To enable a study involving a human-operated vehicle with LIDAR based obstacle avoidance, the NADS-1 simulator was coupled with the Chrono simulation engine (Project Chrono Development Team, 2026; Zhou, et al., 2023) and ROS2-based components. This setup, developed in cooperation with the University of Wisconsin-Madison, formed a virtual environment with simulated LIDAR sensor data, allowing object detection and avoidance algorithms to be integrated and evaluated under real-time, human-in-the-loop conditions. *SimScene* provided a shared, time-synchronized scene representation between the simulators.



Figure 6: Co-simulation involving NADS-1 simulator cab, motion, vehicle dynamics and 360-degree visualization system alongside Carla.

In a separate deployment, *DSRISimConnect* was used to integrate the Carla simulator (Dosovitskiy, et al., 2017) and ScenarioRunner into DSRI's driving simulator. Dynamic actor types supported by Carla—including vehicles, pedestrians, and traffic lights—were synchronized using *SimScene*. Vehicle control

and actor state were exchanged bidirectionally, allowing responsibility for scene entities to be distributed between both simulators.

DSRISimConnect's frame-based scheduling was used to advance Carla in lockstep with the rest of the system whenever Carla and ScenarioRunner were able to maintain real-time execution. Overall real-time behavior in co-simulation was bounded by the size and complexity of the scenario executed in Carla. In scenarios where occasional frame delays occurred, *DSRISimConnect* preserved temporal consistency across the remaining simulator subsystems.

This integration allowed Carla's sensor suite and traffic scenarios originally developed for autonomous-driving research to be introduced incrementally into ongoing simulator studies without invasive modifications to existing simulator subsystems.

All application scenarios evaluated in this work were deployed within a single facility; extending this approach to a large geographically distributed simulator networks would require additional interest-management mechanisms, which were not necessary for the scenarios considered here.

Conclusions and implications

DSRISimConnect provides a practical foundation for building and evolving human-in-the-loop research driving simulators while preserving predictable real-time behavior. Deployment in DSRI simulators demonstrates that distributed execution and runtime-extensible shared state can be introduced incrementally, without disrupting ongoing research or safety-critical subsystems, and that the same framework can serve as a clean baseline for new simulator installations.

For simulator operators and research teams at DSRI, the framework reduces the cost and risk of introducing new functionality. By decoupling subsystem integration from fixed data layouts or message-driven interfaces, *DSRISimConnect* simplifies the incorporation of dynamic scene actors, experiment-specific data, and external simulation tools, enabling new research workflows without global interface redesign or system downtime.

Beyond specific use cases, the framework's state-centric and runtime-extensible design provides a degree of future-proofing for driving simulation research. It enables existing functionality and scene actors to be extended, and new sensors, algorithms, and co-simulation partners to be integrated, without introducing new communication protocols or redefining static data structures. This allows simulators to adapt to evolving research questions and technologies without requiring architectural redesign.

From an industrial and applied-research perspective, the framework supports both hybrid architectures—where certified and legacy components coexist with newly developed subsystems—and green-field simulator designs that avoid proprietary infrastructure from the outset. While *DSRISimConnect* is currently only employed in DSRI simulators, we are open to making it available to partners based on interest and funding.

Overall, *DSRISimConnect* supports a wide range of current and future driving simulation research and provides a scalable, experience-driven foundation for sustaining and extending human-in-the-loop simulator facilities in evolving research and application environments.

Acknowledgements

This work was supported by in part by US DOT NHTSA contract # 693JJ922E00063/693JJ922F00118N and NSF Grant # 2209794.

References

- Curtiss-Wright Corporation, 2015. SCRAMNet Network: Shared-Memory Communication Architecture, s.l.: Curtiss-Wright.
- Dosovitskiy, A. et al., 2017. CARLA: An Open Urban Driving Simulator. s.l., Proceedings of the 1st Annual Conference on Robot Learning.
- IEEE1588, 2019. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems (IEEE 1588-2019), s.l.: IEEE.
- Macenski, S. et al., 2022. Robot Operating System 2: Design, architecture, and uses in the wild. Science Robotics, 7(66), p. eabm6074.
- Object Management Group (OMG), 2015. Data Distribution Service (DDS) Specification, Version 1.4, s.l.: OMG formal/15-04-10.
- Project Chrono Development Team, 2026. Project Chrono: An Open-Source Multi-Physics Simulation Engine. [Online] Available at: <https://projectchrono.org/>
- Schwarz, C. et al., 2023. The long and winding road: 25 years of the national advanced driving simulator.. IEEE computer graphics and applications, 43(4), pp. 121-128.
- Zhou, Z. et al., 2023. A Chrono-based framework for large-scale traffic simulation with human in the loop. Lisbon, Portugal, ECCOMAS Thematic Conf. Multibody Dynamics.

A HiL-Driven Brake Pedal Feedback Architecture for Driver-in-the-Loop Simulation

Raphael Groß², Anton Tworek¹, Bernhard Riegert¹, Peter E. Pfeffer²

(1) MdynamiX AG, Hessesstr. 89, 80797 Munich, e-mail: {anton.tworek, bernhard.riegert}@mdynamix.de

(2) Munich University of Applied Sciences, Lothstr. 64, 80797 Munich, e-mail: {raphael.gross, peter.pfeffer}@hm.edu

Abstract – The increasing demands of modern vehicle development require earlier and more integrated validation methods. Driving simulators have become essential for front-loading test activities, yet realistic brake pedal feedback has remained a limitation. While steering feel can already be reproduced convincingly through remote Hardware-in-the-Loop (HiL) integration, braking systems introduce additional complexity due to hydraulic nonlinearities, ABS/ESC dynamics, and the strong influence of brake pedal feel on driver perception. This paper presents the MXbrakeLink, a low-latency haptic coupling that connects a complete brake system on a MXbrakeHiL with the MXbrakePedal inside a Driver-in-the-Loop (DiL) driving simulator. The system enables drivers to experience authentic brake pedal behaviour, including force–travel characteristics, ABS pulsation, and nonlinear hydraulic effects, while the physical brake hardware remains stationary in the laboratory. This minimizes added driving simulator mass, maintains mechanical simplicity, and ensures high physical fidelity. An alternative mode allows software-generated brake pedal forces when independent HiL and DiL operation is required. The paper describes the architecture of the MXbrakeHiL, MXbrakePedal, and MXbrakeLink, evaluates their performance with emphasis on bandwidth and end-to-end latency, and demonstrates how the system supports a closed workflow between HiL and DiL. Finally, modularity and scalability are discussed, highlighting applicability across hydraulic and brake-by-wire systems, driving simulator classes, and real-time environments.

Keywords: Hardware-in-the-Loop (HiL), Driver-in-the-Loop (DiL), ESC/ ABS, Brake Systems, Brake Pedal Feel

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

Automotive development faces two opposing needs: ever-increasing software content, and ever-shorter time-to-market. As a result, validation is being shifted earlier in the V-process, with Hardware-in-the-Loop (HiL) and Driver-in-the-Loop (DiL) methods now complementing, and in many cases front-loading, activities that historically depended on scarce prototype vehicles. In braking, this shift is particularly impactful because functions such as ABS, ESC, and Brake-by-Wire (BBW) are both safety-critical and brand-defining through their influence on brake feel. HiL offers a reproducible, safe, and instrumented environment to develop and benchmark such controllers, even when no representative vehicle is available, while DiL provides the human-in-the-loop context needed to evaluate perception, HMI, and drivability under controlled conditions. (Friggs & Ewbank, 2022)

However, DiL acceptance is reliant on haptic credibility. For brake pedals, this credibility is determined by three factors:

- (i) a realistic force–travel characteristic across the entire operating range,
- (ii) a dynamic response that reproduces fast events such as ABS pressure modulation, or booster interventions, and
- (iii) low end-to-end latency from the driver's input to the perceived brake pedal reaction.

Driving simulators can already synthesize vehicle motion and visual cues with high fidelity, and the remaining gap is to bring the authentic brake system feedback, originating from real hydraulic or electromechanical hardware, into the driver's foot. (Ansible Motion, 2015)

This paper addresses that gap by presenting the MXbrakeLink, a low-latency linkage that leaves the complete brake hardware on a MXbrakeHiL in the lab, while transmitting its force feedback to the driving simulator brake pedal. The approach transfers a pattern that is already established for

steering (MXsteerLink (Hendewerk, et al., 2026 (in press))) to braking: the driver experiences the physical brake system's response in the driving simulator, yet the motion platform remains minimally loaded because the heavy hardware stays off-board. The system architecture (MXbrakeHiL, MXbrakePedal, MXbrakeLink) is described, the performance is quantified with an emphasis on latency and bandwidth, and workflow and use cases are demonstrated that couple subjective driver assessment with objective HiL data early in development. Finally, modularity and scalability are discussed, showing how the concept extends from conventional hydraulic systems to BBW configurations, and different environments.

Role of HiL and DiL in Brake System Development

Brake system development poses unique challenges due to the combination of safety-critical functions, nonlinear hydraulic dynamics, strong human-machine interaction requirements, and the need to balance objective control performance with subjective brake pedal feel. Within modern development processes, Hardware-in-the-Loop (HiL) and Driver-in-the-Loop (DiL) environments form complementary pillars that enable fast, safe, and repeatable progress throughout the V-model. Together, they allow engineers to explore a much broader space of scenarios than would be feasible with prototype vehicles alone.

Role of HiL in Brake System Development

Hardware-in-the-Loop (HiL) provides a controlled, real-time environment where vehicle dynamics models, traffic scenarios, and environmental conditions interact directly with real brake hardware, including the brake pedal, ABS/ESC ECU, brake lines, and calipers. This setup enables extensive testing that would be costly, time-intensive, or unsafe in prototype vehicles. HiL is well suited for automated parameter sweeps, regression testing, calibration variant evaluation, and robustness analysis. Critical manoeuvres such as panic braking, lane changes during ABS activity, μ -split braking, and rapidly changing road surfaces can be repeated consistently, and HiL supports fault injection for validating functional safety concepts, diagnostic behaviour, and degraded system modes such as pressure build-up irregularities. (Park, et al., 2025)

A major advantage of braking HiL systems is the availability of real hydraulic behaviour and original hardware feedback, which cannot be fully reproduced by simulation models. While simulations approximate hydraulic compliance, friction, hysteresis, and valve dynamics, they cannot capture

the nonlinearities and component tolerances present in actual systems. HiL therefore provides unmatched physical fidelity, ensuring that pressure build-up curves, ABS modulation profiles, and dynamic hysteresis correspond to the real system. This accuracy is essential for validating ABS slip control, ESC yaw stabilisation, brake-by-wire blending, energy recuperation strategies, booster control, and fallback modes, enabling safe and repeatable controller development and calibration.

Role of DiL in Brake System Development

Driver-in-the-Loop (DiL) complements HiL by capturing human perception and subjective evaluation in real time. DiL is ideal for assessing brake pedal feel, braking comfort, controllability, perceived ABS behaviour, and driver confidence during demanding manoeuvres. Dangerous or rare events, such as high-speed ABS braking or μ -split corner entry, can be tested safely and consistently. Rapid adjustment of brake pedal force-travel characteristics, booster maps, or blending parameters allows efficient exploration of subjective preferences before returning to HiL or SiL for objective refinement. DiL is particularly valuable for evaluating modulation capability, perceived stability during ABS operation, brake pedal feel consistency, and brand-specific brake feel targets, which are aspects that cannot be obtained from HiL or simulation alone (Isacco, et al., 2023).

Complementary Use of HiL and DiL

HiL and DiL are not alternatives but mutually reinforcing tools. HiL ensures objective accuracy, physical realism, and hardware-correct behaviour, while DiL adds perception, comfort, HMI understanding, and driver acceptance. Insights from DiL tests often flow back into SiL and HiL to refine models, retune controllers, or adjust brake pedal characteristics. Conversely, HiL results ensure that DiL scenarios are grounded in accurate braking behaviour. Their combination provides a comprehensive development workflow suited to the complexity of modern brake systems.

MXbrakeHiL and MXbrakeLink Architecture

MXbrakeHiL Architecture

The MXbrakeHiL is a fully integrated brake system Hardware-in-the-Loop environment designed to reproduce real brake behaviour with high physical fidelity. It combines real hydraulic or electromechanical brake hardware with a real-time simulation platform, enabling the brake system to interact with a virtual vehicle model as it would in an actual vehicle. At its core is the complete brake

hardware chain, including the master cylinder, brake booster, hydraulic lines, modulators, valves, sensors, and brake calipers or representative corner modules. Using the unmodified physical system allows the MXbrakeHiL to capture nonlinearities, hysteresis, valve dynamics, and pressure build-up rates that simulation alone cannot reproduce, ensuring accurate evaluation of ABS, ESC, brake-by-wire, and blending strategies. (Tworek, et al., 2022)

A real-time computing platform exchanges high-bandwidth sensor and actuator signals with the hardware, providing a deterministic loop for fast brake control algorithms. The possibility to integrate fault injection and configurable manoeuvre libraries support large-scale testing, creating a realistic and flexible platform for objective brake system validation and controller development.

MXbrakePedal Architecture

The MXbrakePedal is a high-performance haptic actuator designed to reproduce realistic brake pedal behaviour in Driver-in-the-Loop and HiL-coupled simulation environments. It applies the principles of dynamic, low-latency force rendering to the braking domain. The actuator connects directly to the vehicle's original brake pedal by replacing the master cylinder, ensuring that all forces and displacements act through authentic brake pedal geometry.

At its core is a high-power spindle actuator capable of generating more than 6 kN of force, covering the full range of pedal forces encountered in passenger cars, high-performance vehicles, and commercial platforms. Its high bandwidth and complex control architecture enable the reproduction of fast transient events such as ABS-induced pulsation, booster interventions, and sudden stiffness changes. This makes the system suitable for studies on brake pedal feel, modulation capability, and driver response in demanding manoeuvres.

The MXbrakePedal operates in two modes. In standalone mode, it generates force–travel curves using configurable linear, progressive, or OEM-specific maps. In external force-input mode, it receives commands directly from a simulation model that is executed on the real-time system, enabling customer-specific characteristics different to the general characteristic curves.

High sampling-rate force sensing, position feedback, and real-time control loops ensure stability, low latency, and a repeatable brake pedal feel, creating a robust platform for accurate and physically convincing brake pedal haptics.

MXbrakeLink Architecture

The MXbrakeLink establishes a real-time connection between the MXbrakePedal in the driving simulator and the brake hardware on the MXbrakeHiL. As shown in Figure 1, the driver's pedal input is

captured at the MXbrakePedal and transmitted digitally to the HiL, where the real brake booster, hydraulic lines, valves, modulators, and brake cylinders respond exactly as they would in a vehicle. The resulting brake pedal forces are sent back to the driving simulator, allowing the driver to experience the authentic braking behaviour of the physical system. The hardware remains stationary in the laboratory while its brake pedal feel and haptic behaviour is transferred seamlessly to the driver.

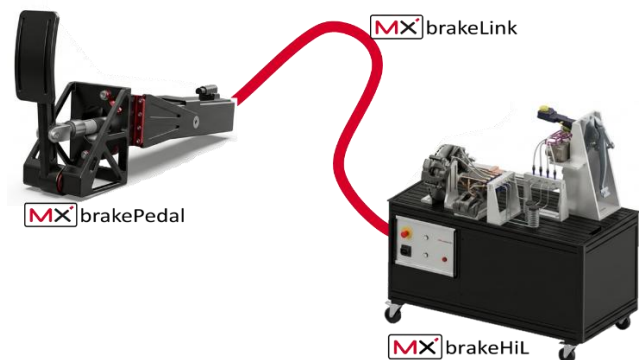


Figure 1 illustrates the MXbrakeLink setup with its subsystems MXbrakeHiL and MXbrakePedal. The subsystems are only connected through power cables and a complex control architecture.

By coupling the driving simulator directly to the MXbrakeHiL, the system reproduces characteristics that standard brake pedal feel models often lack. This gives the driver access to the original brake pedal feel and therefore enables better driving simulator usage. The driving simulator effectively becomes an extension of the HiL, enabling subjective feedback on real hardware responses without relocating the physical brake system.

To ensure convincing brake pedal feel, the MXbrakeLink is designed with the same mentioned low end-to-end latency targets.

Another advantage is consistency across test environments. Because all setups use the same MXbrakePedal and MXbrakeHiL, results remain comparable whether testing occurs on a static driving simulator, a motion platform, or at the MXbrakeHiL without a driver. Calibration and pedal maps transfer directly, reducing redundant work and the need for additional prototype vehicles.

By combining physical accuracy with immersive interaction, MXbrakeLink effectively closes the loop between objective HiL development and subjective DiL assessment.

Performance Evaluation of MXbrakeLink

Necessity of good Performance

The performance of the MXbrakeLink is essential for delivering a convincing and development-relevant brake pedal experience in the driving simulator. Because the system forms the real-time link between the driver's input and the physical brake hardware on the MXbrakeHiL, delays and inaccuracies immediately affect the perceived braking quality. Brake pedal feel is highly sensitive; variations in stiffness, force build-up, timing, or pulsation directly influence controllability, driver confidence, and comfort. The MXbrakeLink must therefore reproduce real hardware behaviour with high fidelity.

Performance is especially critical during fast dynamic events. These occur within milliseconds, and insufficient bandwidth or latency limits the driving simulator's usefulness for brake specific use-cases.

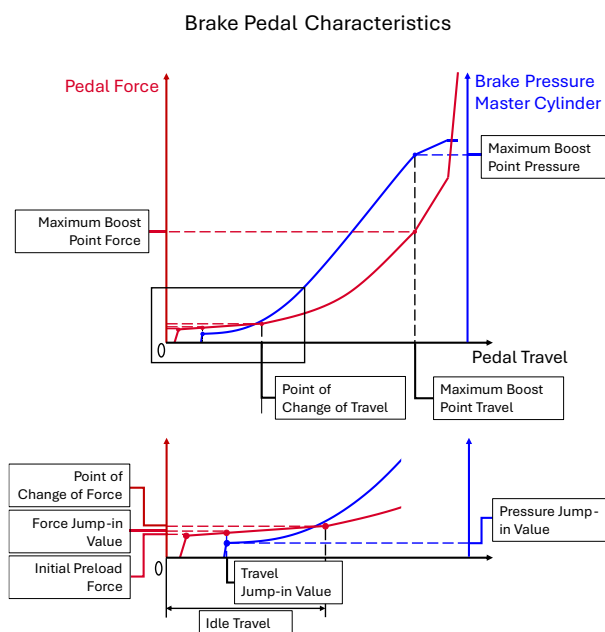


Figure 2 shows the basic characteristics of the brake pedal (based on (Trutschel, 2007) (Breuer & Bill, 2017))

When evaluating brake pedal feel, most relevant characteristic points of the force-travel curve are concentrated within the first third of the pedal stroke, as illustrated in Figure 2. Because this range also corresponds to normal day-to-day driving with decelerations below 2 m/s^2 (Camargo, et al., 2015), an accurate representation of this part of the pedal travel is particularly important. Accurate reproduction of these small variations is essential for realistic pedal feel and for assessing calibration differences during development.

Accuracy also ensures that DiL results correlate meaningfully with HiL data. Accurate and timely force

reproduction preserves continuity across development stages, making the driving simulator an effective extension of the MXbrakeHiL and enabling reliable, repeatable evaluations in early project phases.

Comparison of Performance parameters of MXbrakeLink

The performance of the MXbrakeLink can be assessed by analysing key signals, primarily brake pedal position and brake pedal force. The system achieves a position latency of less than 1 ms, with a maximum deviation of under 0.1 mm between the MXbrakePedal and MXbrakeHiL positions. For brake pedal force, the end-to-end latency is typically below 7 ms, and the force deviation remains generally within 30 N. Slightly larger instantaneous deviations may occur during abrupt driver inputs, but the system maintains high fidelity and responsiveness across the full operating range, enabling a realistic and consistent brake pedal feel in the driving simulator.

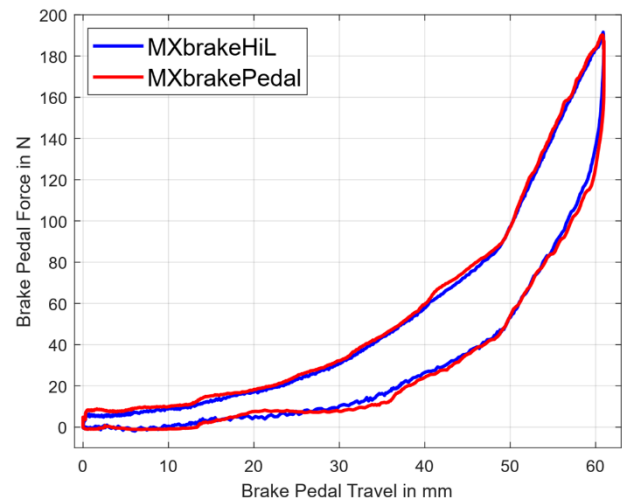


Figure 3 shows a comparison of MXbrakeHiL force (blue) and MXbrakePedal force (red) while operating in Link Mode with brake pedal velocities up to 65 mm/s while pushing and 100 mm/s while releasing

Figure 3 compares the pedal characteristics of the MXbrakeHiL and MXbrakePedal operating in Link mode. The MXbrakePedal signal (red) is manually applied by the driver in the driving simulator. The previously stated accuracy targets, agree with the evaluation of professional test drivers. Brake pedal force deviations remain within approximately 5 N. Achieving this level of precision in Link mode ensures that the driver perceives the same characteristic sensations that originate from the brake system under test on the HiL.

Use Cases and Workflow

The MXbrakeLink enables a continuous workflow that moves seamlessly between HiL-based objective testing and DiL-based subjective evaluation.

Development begins on the MXbrakeHiL, where engineers validate brake control functions, pedal characteristics, and system behaviour under controlled, repeatable conditions. With real brake hardware, authentic hydraulic behaviour, and full system observability, the MXbrakeHiL provides an ideal environment for calibration, functional safety testing, and robustness evaluations. New control strategies, such as brake blending logic, booster control, or ABS parameter changes, are verified to ensure stability, correct functionality, and realistic hardware interaction.

Once the system performs as intended, testing transitions to the DiL driving simulator. Through the MXbrakeLink, the real brake system response is transmitted to the MXbrakePedal, enabling drivers to experience authentic brake pedal feel and system dynamics. This stage is particularly valuable for assessing modulation capability, brake pedal feel, driver confidence, and the subjective impact of calibration adjustments. The driving simulator also allows rapid testing of brake pedal characteristics, road conditions, and controller variants without hardware modifications. Critical manoeuvres such as μ -split braking or high-speed ABS events can be performed safely and repeatedly.

A central element of this workflow is consistent evaluation across both environments. The MXbrake ecosystem integrates MXeval, an automated KPI-based analysis tool, and MXevalApp, a structured subjective driver-feedback Android application, ensuring that objective and subjective assessments follow the same definitions and evaluation criteria. MXeval enables systematic KPI-based analysis of brake performance on the MXbrakeHiL, including pedal force progression, pressure build-up dynamics, ABS signatures, and stability metrics. These KPIs provide a robust baseline for comparing calibration variants and verifying controller updates.

In the DiL phase, MXevalApp allows drivers to provide ratings for standardised questions like brake pedal feel, controllability, modulation capability, and perceived stability or custom inputs. Because both tools use aligned data structures, results from HiL and DiL can be directly correlated. This enables engineers to understand how objective system behaviour influences subjective driver perception, identify inconsistencies between measured and perceived behaviour, and refine calibration in a targeted manner.

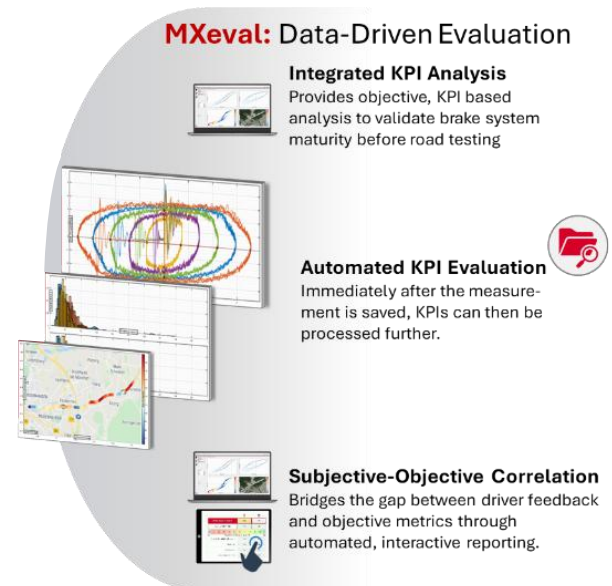


Figure 4 shows important aspects of the objective evaluation software MXeval and the subjective assessment software MXevalApp which are both integrated used together with the MXbrakeLink

The combined use of MXeval and MXevalApp increases traceability, improves comparability across development stages, and significantly strengthens the overall HiL ↔ DiL workflow. Insights gained during DiL testing often lead to refinements in controller parameters, pedal maps, or blending strategies. These adjustments are then re-validated on the MXbrakeHiL, forming a closed HiL ↔ DiL loop that aligns subjective experience with objective system behaviour and enables efficient, coherent brake system development.

Modularity and Scalability

A key strength of the MXbrake ecosystem is its modular and scalable architecture, which enables the same core components to be applied across a wide range of brake systems, driving simulator types, and real-time environments. The MXbrakeHiL, MXbrakePedal, and MXbrakeLink function as interoperable building blocks that can be combined or reconfigured without hardware redesign, allowing the system to be adapted easily to different development stages and vehicle architectures.

Modularity begins at the MXbrakeHiL, which can integrate various brake system types, including conventional hydraulic brakes, electromechanical boosters, and brake-by-wire systems. Components such as calipers, modulators, master cylinders, valves, and BBW actuators can be exchanged or combined depending on the application. This ensures long-term relevance across multiple platforms and generations.



Figure 5 shows the different systems of the MXbrake environment with the MXbrakeLink, a combination of the standalone capable systems MXbrakeHiL and MXbrakePedal

Scalability extends across simulation environments. The MXbrakeLink and MXbrakePedal operate consistently on fixed-base driving simulators, dynamic motion platforms, and laboratory workstations. Because the same pedal actuator, coupling logic, and signal-processing chain are used throughout, results remain comparable and fragmentation between setups is avoided.

The system is also independent of specific real-time frameworks, supporting platforms from dSPACE, VI-Grade, Vector, IPG and other EtherCAT-based environments. As a result, the MXbrake platform remains flexible and future-proof: brake hardware can evolve, driving simulator configurations can change, and real-time systems can be updated while maintaining a coherent and efficient workflow across all phases of brake system development.

Conclusion

The MXbrakeLink concept reduces a long-standing gap in brake system development by merging the physical accuracy of Hardware-in-the-Loop environments with the perceptual benefits of Driver-in-the-Loop simulation. By coupling the real brake hardware on the MXbrakeHiL with the MXbrakePedal in the driving simulator, developers and test drivers can experience authentic hydraulic behaviour, nonlinear system effects, and dynamic brake events without relying solely on simulation models or prototype vehicles. The system's extremely low latency ensures that fast transients and subtle brake pedal feel characteristics are reproduced accurately, enabling realistic driver interaction and subjective evaluations that are comparable to the real vehicle.

The performance analysis shows that the transmitted force, velocity, and position signals match the MXbrakeHiL reference with exceptional precision, demonstrating that the MXbrakeLink is capable of replicating real-world brake behaviour

within the driving simulator environment. This capability allows the DiL platform to act as a natural extension of the MXbrakeHiL, supporting rapid calibration cycles, earlier validation phases, and more coherent development workflows.

Reference

- Ansible Motion, 2015. *Ansible Motion*. [Online] Available at: <https://www.ansiblemotion.com/automotive-driver-in-the-loop-simulation-articles/test-handling-effects-on-automobile-drivers> [Accessed on 19.03.2026].
- Breuer, B. & Bill, K. H., 2017. *Bremsenhandbuch*. s.l.:Springer Fachmedien Wiesbaden GmbH.
- Camargo, E. et al., 2015. *Pedal Feeling Study Basis on Customer Braking Behavior*, Sao Paulo: s.n.
- Friggs, A. & Ewbank, D., 2022. Optimizing the Vehicle Development Process by Combining Driving Simulators and Virtual Test Driving. In: P. E. Pfeffer (ed.) *12th International Munich Chassis Symposium 2021. Proceedings*. 12th International Munich Chassis Symposium 2021 Hrsg. s.l.:Springer Vieweg, Berlin, Heidelberg, pp. 61-72.
- Hendewerk, J., Jörg, K., Thaler, K. & Pfeffer, P. E., 2026 (in press). Enhancing Steering Feel in Driving Simulators via Remote Hardware in the Loop Integration. In: P. E. Pfeffer (ed.) *16th International Munich Chassis Symposium 2025 (IMCS 2025)*. s.l.:Springer Vieweg, Berlin, Heidelberg.
- Isacco, L., Mura, F. L. & Giberti, H., 2023. Brake System Model for a Haptic Brake Pedal Control. In: *5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. s.l.:IEEE, pp. 01-05.
- Park, S. M. et al., 2025. Implementation of a Real-Time ESC Simulation Environment. In: P. E. Pfeffer (es.) *15th International Munich Chassis Symposium 2024. IMCS 2024. Proceedings*. s.l.:Springer Vieweg, Berlin, Heidelberg, pp. 59-76.
- Trutschel, R., 2007. Analytische und experimentelle Untersuchung der Mensch-Maschine-Schnittstellen von Pkw-Bremsanlagen. In: Universitätsverlag Ilmenau (ed.) *Kraftfahrzeugtechnische Berichte*. Ilmenau: Augsburg, Klaus.
- Tworek, A., Thaler, K., Radspieler, M. & Pfeffer, P., 2022. An Innovative HiL Test Environment with a Fully Integrated Brake System and Automated Brake Performance Evaluation. In: P. E. Pfeffer (ed.) *12th International Munich Chassis Symposium 2021. Proceedings*. s.l.:Springer Vieweg, Berlin, Heidelberg, pp. 341-357.

SENSORS IN DRIVING SIMULATION

Active Infrared with SE-RAY-IR & SCANeR Studio

Florian Faucher¹, Antoine Boudet¹, Thomas Nguyen²

(1) OKTAL Synthetic Environment, 11 avenue du Lac - 31320 Vigoulet-Auzil - France, e-mail: {ffaucher, aboudet}@oktal-se.fr

(2) AVSimulation, 1 Cours de l'Île Seguin - 92100 Boulogne-Billancourt - France, e-mail: thomas.nguyen@avsimulation.fr

Abstract -

This paper introduces a prototype interface enabling cosimulation between an automotive simulation platform (SCANeR Studio) and a spectral EO/IR rendering engine (SE-RAY-IR). The workflow combines dynamic driving scenarios with spectral and temporal modeling of laser illumination and materials to generate meaningful data for active infrared imaging. A user-oriented setup process is described, from scenario definition to imaging system configuration, including emission profile, range gating, and material properties. The interface synchronizes the synthetic environment with the scene dynamics through complementary APIs. Results demonstrate the ability to support fine-tuning of imaging system configurations and the analysis of active sensing disturbances on detection. The framework is illustrated through a simple road scenario for initial evaluation of active IR concepts and highlights advantages over general-purpose rendering engines.

Keywords: Active Infrared, Laser beam, Electro-optic, Spectral imagery, Synthetic environment

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

Infrared (IR) and electro-optical (EO) imaging systems are increasingly considered for automotive perception in low-light and night-time conditions. In particular, the extension of the visible spectrum up to near-infrared (NIR), combined with active illumination system (LED or laser sources), improve detection of vulnerable road users and low-contrast objects. However, accurately reproduce such a technology in a simulation tool requires to model both passive and active radiometric contributions in a physically consistent manner.

While general-purpose rendering engines enable the rapid generation of high-resolution, photorealistic images that are visually compelling to the human eye, they are not intended to model spectral information and variations of the radiometric quantities in the visible and infrared domain.

This paper presents a prototype of a cosimulation workflow between two software solutions: SCANeR Studio for scenario creation and vehicle dynamics, and SE-RAY-IR which is a physics-based spectral rendering engine dedicated to electro-optical (EO) and infrared (IR) domain. The user workflow is illustrated through a simple road scenario on which a laser illuminator mounted on a vehicle interacts with the synthetic environment, and a colocated IR imaging system captures the resulting spectral and radiometric response of the virtual scene.

The results presented in this paper rely on a simplified

pinhole camera and do not aim at reproducing the specific behavior of a real instrument. Instead, the objective is to show how interfacing both softwares enables to produce meaningful synthetic data for automotive perception oriented studies.

Product Solution

The prototype introduced below is the outcome of an industrial partnership between AV Simulation (SCANeR Studio) and Oktal Synthetic Environment (SE-WORKBENCH-EO). Both are "sister companies", with a common SOGECLAIR shareholder which strengthens and perpetuates the partnership.

SCANeR Studio

SCANeR Studio is a driving simulation software designed to configure and execute complex traffic scenarios using a comprehensive 3D asset library and dedicated scenario design tools. It enables efficient creation of road infrastructures, while mobile entities are driven by dedicated models defining their behavior and dynamics, resulting in realistic interactions, including night-time scenarios involving vulnerable road users (eg. figure 1).

The simulation engine produces physically consistent vehicle trajectories, enabling fine-tuning of the whole imaging system configuration, including the position-



Figure 1: SCANer Studio night scenario

ing and orientation as well as motion induced effects studies.

SE-RAY-IR

SE-RAY-IR (from SE-WORKBENCH-EO) is the ray-tracing engine of OKTAL-SE that computes high realistic spectral images both in visible and the infrared spectrum (from 0.4 microns to 15 microns) issued from complex scenarios including 3D terrain, 3D targets, atmospheric conditions, sensors models and associated trajectories

The underlying model relies on thermal emission definition of textured materials, bidirectional reflectance distribution function (HSTG, Li-Torrance) as well as directional emissivity. It also accurately takes into account of the atmospheric propagation, diffuse reflection of the environment (sky, ground), light sources emissions and multiple specular reflections.

Active Spectral IR Rendering Principle

This feature extends SE-RAY-IR engine by introducing active illumination through laser sources. Each laser emitter is characterized by its beam type (e.g., uniform, Gaussian, or user-defined pattern), temporal properties (pulse duration and repetition), spectral profile, and emitted energy.

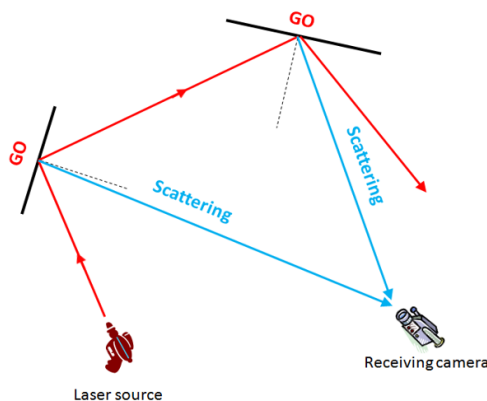


Figure 2: Illumination principle

On the sensor side, the acquisition process is mod-

eled by configuring the temporal behavior of an opto-electronic sensor shutter. This includes parameters such as activation delay, integration time, and an optional temporal response profile, allowing the simulation of time-dependent signal acquisition.

Light propagation is explicitly accounted for by considering the time-of-flight of optical paths, computed from their geometric length and the speed of light. This enables consistent modeling of the interaction between emitted and received signals in active illumination scenarios.

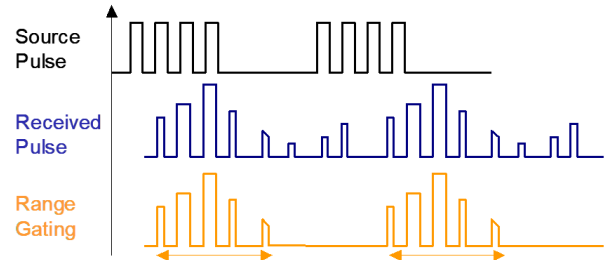


Figure 3: Range gating principle

As illustrated in figure 2, light is emitted from the laser source and propagates through the scene before being reflected toward the sensor, where it is temporally integrated according to the sensor shutter profile.

By controlling the sensor shutter timing, it is possible to select the portion of light that has traveled over a specific duration, corresponding to a given propagation distance. This principle, commonly referred to as range gating, enables the enhancement and detection of objects by filtering out unwanted radiometric contributions, as illustrated in Fig. 3. The emitted pulse is temporally shifted by its propagation through the scene and integrated according to the sensor shutter profile.

At the focal plane, the total measured radiance is expressed as the sum of a passive component L_{out}^P and an active L_{out}^A component :

$$L_{out}(M, \vec{\omega}_{out}) = L_{out}^P(M, \vec{\omega}_{out}) + L_{out}^A(M, \vec{\omega}_{out}) \quad (1)$$

The passive contribution, associated with ambient illumination, is assumed to be invariant over the integration period and is therefore independent of the temporal gating.

$$L_{out}^A(M, \vec{\omega}_{out}) = \frac{\int_{\Delta t} L_{out}^A(M, \vec{\omega}_{out}, t) \cdot \Omega(t) dt}{\int_{\Delta t} \Omega(t) dt} \quad (2)$$

In contrast, the active contribution depends on the temporal interaction between the emitted pulse and the sensor shutter. To ensure consistency between both components, the active signal is normalized by the integral of the shutter opening profile $\Omega(t)$, allowing it to be expressed as an equivalent radiance and combined with the passive contribution.

Interface prototype

SCANer Studio relies on a model-based architecture with low-level APIs, enabling user specific implementation of interface models to connect external applications.

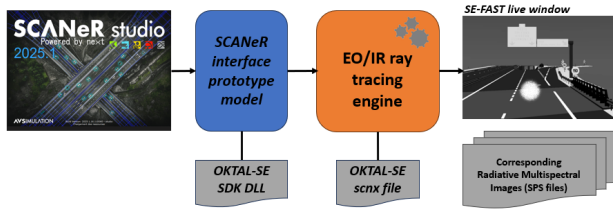


Figure 4: Simulation workflow

The EO/IR rendering engine SE-RAY-IR also exposes a meaningful C-based API, enabling external control of the synthetic environment. The dynamic scene elements, including mobile entities, sensor position and orientation, and illumination parameters, are updated at each simulation step. Thanks to both APIs, a dedicated interface (blue box on Fig. 4) was easily developed to exchange scenario data and synchronize SCANer Studio with the rendering engine.

The simulation execution is driven by SCANer Studio, as illustrated in Fig. 4. During scenario execution, a rendering window provides visual feedback in real time (or offline), while corresponding spectral radiance outputs files are generated for post-processing and analysis using dedicated tools from the SE-WORKBENCH-EO.

Methodology

This section adopts the perspective of a user setting up an active IR simulation study from scratch to perform an initial analysis. It covers the definition of the road scenario, the assignment of material properties, and the configuration of thermal, atmospheric, and imaging system parameters.

Road scenario preparation

The scenario consists of a single ego vehicle equipped with both an IR imaging system and a laser illuminator mounted on the top of it. The ego vehicle evolves along a simple road environment designed to highlight the interaction between scene geometry, sensor configuration, and active illumination.

A speed bump is introduced to generate realistic motion perturbation, enabling the analysis of sensor behavior under typical road conditions including vibrations and orientation changes. Several panels are placed along the road and serve as controlled targets within the scene. These elements are assigned distinct material properties.

Figure 5 shows a snapshot of the scenario during execution, with a visualization window positioned at the location of the future IR camera, displayed using a standard non-spectral OpenSceneGraph rendering.

Synthetic materials preparation

An external extension provided by OKTAL-SE enables the assignment of pre-filled physical material properties to 3D assets directly from SCANer Studio (see Fig. 6). First-order physical information can be applied on textures or geometry nodes

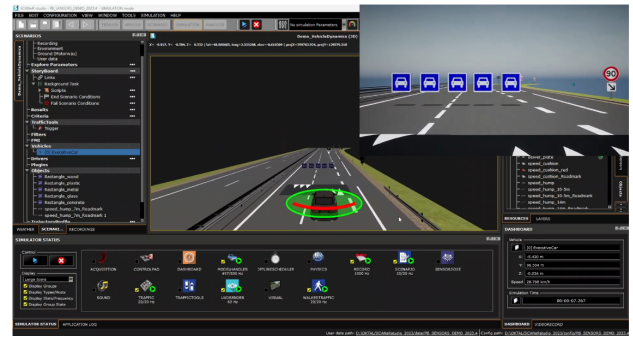


Figure 5: Scenario preparation

to support preliminary EO/IR simulations (more advanced material modeling can be performed with SE-WORKBENCH-EO environment).

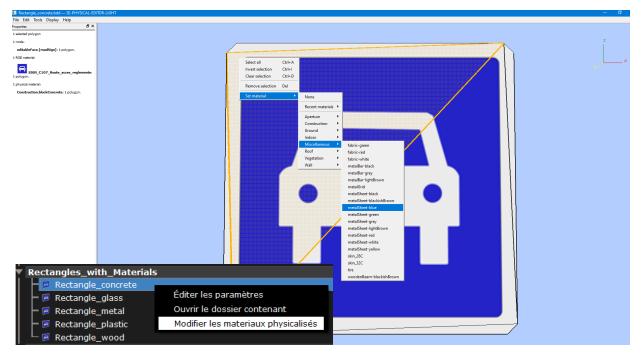


Figure 6: Scenario preparation

The five panels introduced in the previous section correspond to five distinct physical materials: concrete, glass, metal, plastic, and wood.

Each material is assigned consistent optronic properties that are described using BRDF models capturing emissivity as well as diffuse and specular reflectance as functions of wavelength. Thermal properties must be specified as well, including conductivity, density, convection, and initial temperature. Combined with the radiative transfer computation, described in the next section, these material properties are then required to pre-compute the temporal evolution of the scene temperatures.

Atmospheric and thermal radiative transfer

Realistic EO/IR simulation requires modeling both the atmospheric conditions and the temporal evolution of scene temperatures. Atmospheric conditions are first defined to compute the illumination of the scene at each time step. The resulting radiance and irradiance history, including shadowing effects, is then used to estimate the thermal state of the scene. Within the SE Workbench, these two successive pre-computation steps are performed using dedicated tools: SE-ATMOSPHERE for atmospheric modeling and SE-THERMAL for thermal compilation. As illustrated in Fig. 7, the resulting radiative and thermal quantities (tables of temperatures) eventually provide the physical inputs required for the EO/IR rendering process.

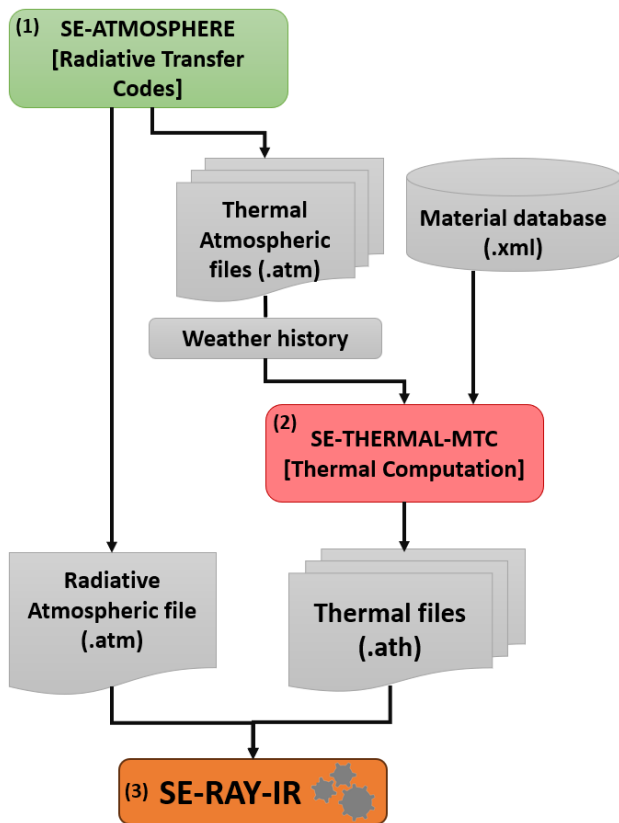


Figure 7: Radiative and thermal computation process

Atmospheric model

SE-ATMOSPHERE exposes a comprehensive graphical user interface and is coupled with several radiative transfer codes, including MODTRAN (more information available on *Modtran website* n.d.). The simulation environment is defined through a set of global parameters, including the date, geographical location, atmospheric model, atmospheric profile (e.g., tropical, midlatitude, summer, or winter), ground altitude, and wind speed. Time-dependent weather conditions, such as haze, clouds, rain, wind, and visibility, together with the spectral and line-of-sight (LOS) sampling, are also configured through the interface (Fig. 8).

Observer latitude (°)	43.6	Longitude (°)	4												
Ground altitude (m)	0			04:30	19:12										
Local date	2012/07/28														
Time zone	UTC+0200 Coordinated Universal Time			12:42	22:49										
No aster	Local time	Profile name	Pressure (hPa)	Temperature (°C)	Humidity (%)	Extinction (km ⁻¹)	Rain rate (mm/h)	Wind speed (m/s)	Wind direction (°)	Emissivity	Cloud	Haze	Cloud height (m)	Cloud thickness (m)	Visibility range (km)
★ E1 -0.07332 °	03:00	Midlatitude_summer	1013	286.15	66.10	0.17	0	0	0	0.9	Cumulus	Rain(VIS>5km)	1000	1500	5
★ A2 -57.0203 °	04:00	Midlatitude_summer	1013	286.15	66.10	0.17	0	0	0	0.9	Cumulus	Rain(VIS>5km)	1000	1500	5
★ E2 25.004 °	07:00	Midlatitude_summer	1013	296.15	66.10	0.17	0	0	0	0.9	Cumulus	Rain(VIS>5km)	1000	1500	5
★ A2 -87.1166 °	07:00	Midlatitude_summer	1013	296.15	66.10	0.17	0	0	0	0.9	Cumulus	Rain(VIS>5km)	1000	1500	5
★ E3 56.4956 °	10:00	Midlatitude_summer	1013	296.15	66.10	0.17	0	0	0	0.9	Cumulus	Rain(VIS>5km)	1000	1500	5
★ A2 -127.651 °	10:00	Midlatitude_summer	1013	296.15	66.10	0.17	0	0	0	0.9	Cumulus	Rain(VIS>5km)	1000	1500	5
★ E3 33.0013 °	16:00	Midlatitude_summer	1013	295.15	66.10	0.17	0	0	0	0.9	Cumulus	Rain(VIS>5km)	1000	1500	5
★ A2 -265.689 °	16:00	Midlatitude_summer	1013	295.15	66.10	0.17	0	0	0	0.9	Cumulus	Rain(VIS>5km)	1000	1500	5
★ E1 -48.0155 °	20:00	Midlatitude_summer	1013	291.15	66.10	0.17	0	0	0	0.9	Cumulus	Rain(VIS>5km)	1000	1500	5
★ A2 -306.342 °	20:00	Midlatitude_summer	1013	291.15	66.10	0.17	0	0	0	0.9	Cumulus	Rain(VIS>5km)	1000	1500	5

Figure 8: Atmospheric model properties

Based on these inputs, the selected radiative transfer solver computes the atmospheric quantities required by both the thermal solver and the ray-tracing engine. The generated outputs include :

- spectral solar and lunar irradiance ($W m^{-2} m^{-1}$)

- atmospheric spectral transmittance (dimensionless)
- and sky spectral radiance ($W m^{-2} sr^{-1} m^{-1}$)

These quantities are tabulated over wavelength and, where applicable, over multiple lines of sight (LOS) defined by their altitude, range, elevation, and azimuth angles. The results are stored in multidimensional lookup tables whose size depends on the selected sampling parameters. Depending on the configuration, generating these tables can require significant computation time.

Thermal influences

The SE-THERMAL solver computes the temporal evolution of the temperature of each scene material from the atmospheric data and the associated material properties. The atmospheric files provide the time history of the incident solar and sky irradiance, while the material description specifies the optical and thermal characteristics of each surface.

For each material, the heat transfer equation is numerically solved using the accumulated radiative fluxes received from the sun and the sky for every illumination direction, defined by its azimuth and elevation angles with respect to the surface normal, and for each spectral band. The computed temperature histories are then exported as lookup tables and used during rendering to retrieve the surface temperature at any simulation time.

Each surface layer is characterized by its average visible absorption coefficient, spectral emissivity, and natural convection coefficient. For materials with a finite or semi-infinite thickness, the thermal model additionally accounts for the thermal conductivity, specific heat capacity, and density of the material.

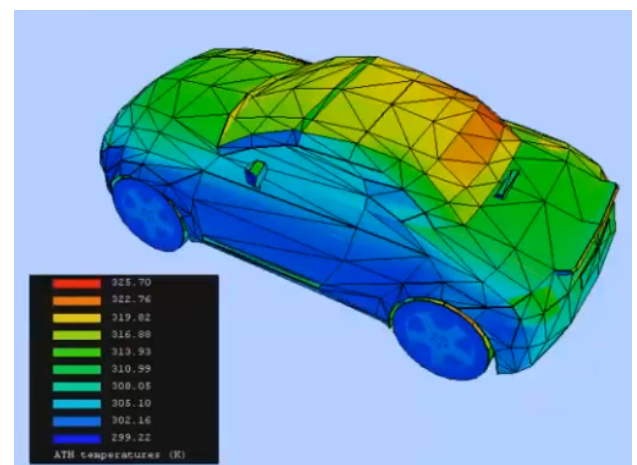


Figure 9: Example of thermal affectionation on a car

The generated temperature maps can be visualized for validation purposes. Figure 9 presents an example of the computed surface temperatures for a stationary vehicle.

Imaging system configuration

The imaging system configuration is defined using an external context creation tool (SE-SCENARIO from SE-WORKBENCH), where all parameters related to

the sensor, the laser source, and the ray tracing process are specified.

This tool generates a .scnx file, illustrated in Fig. 4, which is loaded by the interface module during simulation.

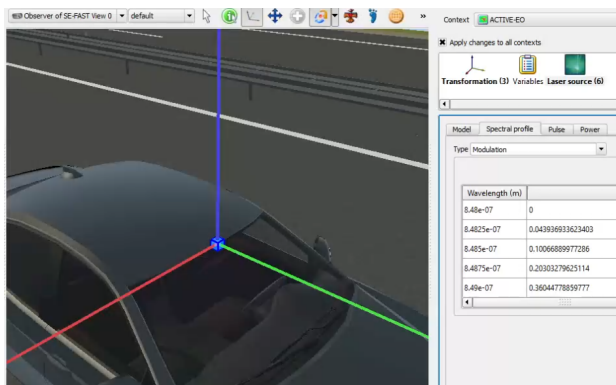


Figure 10: Context settings

As shown in Fig. 10, the laser source and the camera are colocated with identical position and orientation to simplify the configuration, although bistatic configurations are also supported.

The laser is first defined with a Gaussian beam pattern and has a spectral profile ranging from 848 to 852 nm, as illustrated in Fig. 10. Its emission power and pulse duration are then specified.

The camera is configured with a spectral band ranging from 700 to 900 nm, covering the end of the visible spectrum and the beginning of the near-infrared domain, and discretized into a set of sampled wavelengths. Additional parameters such as sensitivity thresholds, aperture, resolution, and depth of field are also defined.

Finally, the sensor shutter is configured by defining its opening duration and temporal profile, synchronized with the laser emission reference time.

Results and discussion

This section presents typical execution results and provides a brief analysis to illustrate its capabilities.

Simulation Execution

Once the configuration is completed, the simulation is launched from SCANer Studio. As shown in Fig. 11, a rendering window displays the IR camera view of the scene during the execution. The projection of the laser illumination on the road surface is clearly visible in the rendered output. Output files are generated during the simulation, with their number determined by the user-defined call frequency configured in SCANer Studio. Their size depends on the level of detail specified in the context parameters.

Spectral image analysis

As shown in Fig. 12, a zoom on the panels reveals two distinct grayscale levels resulting from the different physical properties of the assigned materials.

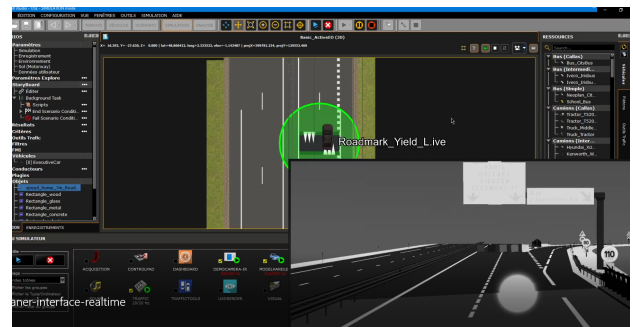


Figure 11: Sensors processing stages

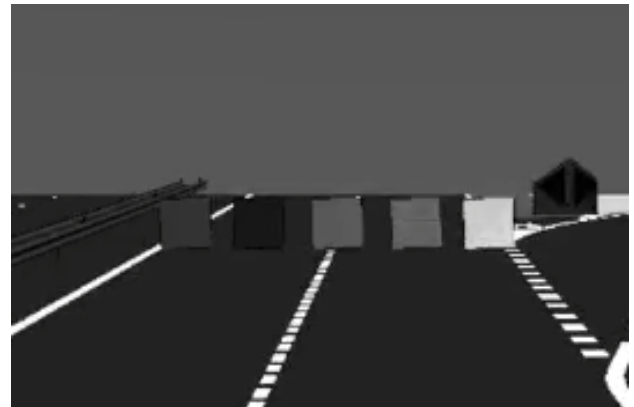


Figure 12: Different shades of panels

For a more in-depth analysis, the generated SPS files can be examined using a dedicated external tool. Figure 13 shows the spectral radiance distribution of a selected pixel in the resulting image can be observed at the pre-sensor plane. The pixel is expressed in radiance integrated over the sub-wavelength band ($W.m^{-2}.sr^{-1}$). The contribution of the laser is clearly visible, matching the user-defined spectral profile within the expected wavelength band and attenuated according to its propagation through the scene.

An additional effect, not discussed previously, is the modeling of speckle patterns induced by coherent illumination. When a coherent laser illuminates a rough surface, the reflected polarized light scattered by multiple micro-scale structures produces an interference pattern known as objective speckle. This speckle pattern is independent of the imaging system and is defined at the focal plane, which corresponds to the level at which images are computed here. Figure 14 illustrates an example of a simulated speckle pattern.

The model is user-configurable and accounts for parameters such as the image size at the focal plane and the laser beam aperture. More information on the computation principle of speckle patterns can be found in Duncan and Kirkpatrick, 2008, as this aspect is beyond the scope of the present work.

Perspectives and limitations

For clarity purposes, the coexistence of two physically based rendering engines has not been discussed so far: a fast rasterization-based engine (SE-

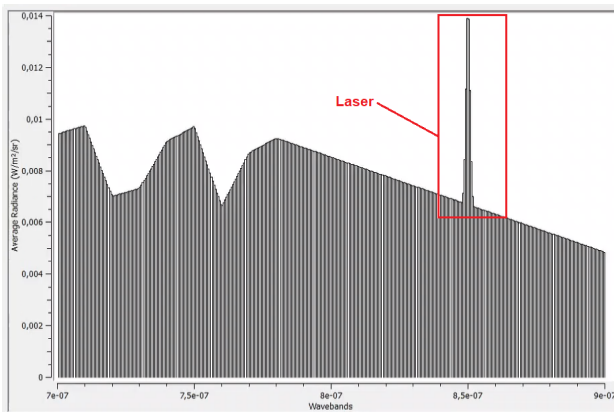


Figure 13: Radiance spectral distribution

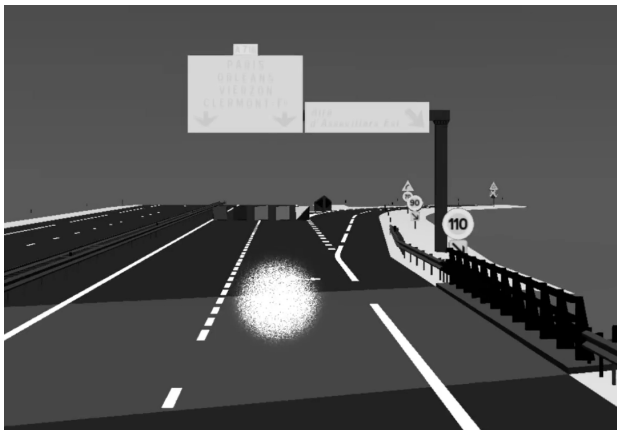


Figure 14: Speckle pattern

FAST-IR) and a slower, high-fidelity ray-tracing engine (SE-RAY-IR).

The rasterization-based engine provides high performance but exhibits limitations, such as support for bi-static configuration, whereas the ray-tracing engine enables more accurate and physically consistent simulations at a higher computational cost. Significant efforts have been made to bridge these two engines, enabling dynamic switching between them while sharing scene data and material resources.

The results presented here are intentionally limited to pre-sensor image generation. The proposed workflow is primarily intended for early design studies, such as sensor positioning and system architecture evaluation, rather than SIL or HIL validation. Advanced sensor modeling, including instrument effects such as the modulation transfer function (MTF), is nevertheless available through a dedicated library, but is outside of the scope of this first prototype.

Conclusion

This paper introduced a prototype interface enabling the cosimulation of SCANer Studio with the physically based spectral EO/IR rendering engine developed by OKTAL-SE. The proposed workflow generates physically consistent spectral data for active imaging in realistic driving scenarios, including vehicle dynamics.

The framework enables early studies of sensor positioning and highlight the potential for studying the impact on detection performance. Combined with the extensive SCANer asset library, it allows the rapid creation of a wide variety of complex driving scenarios, including challenging nighttime situations.

Compared with conventional rendering engines generating high resolution 8-bit RGB images, the proposed approach provides high dynamic range spectral outputs better suited to perception algorithms, where radiometric fidelity may be more valuable than image resolution for early detection under degraded conditions.

Finally, the framework supports both visible and infrared spectral bands within the same simulation environment and relies on a mature and industrially validated software product (Cathala and Barbé, 2018).

References

- Cathala, T. and Barbé, S., 2018. VALIDATION OF SE-WORKBENCH-EO IN THE VISIBLE SPECTRAL BAND. In: Available at <https://www.oktal-se.fr/download/14/electro-optic-infrared/4585/validation-of-se-workbench-eo-in-the-visible-spectral-band.pdf>.
- Duncan, D. D. and Kirkpatrick, S. J., 2008. Algorithms for simulation of speckle (laser and otherwise). In: V. V. Tuchin and L. V. Wang eds. *Complex Dynamics and Fluctuations in Biomedical Photonics V*. Vol. 6855. Proceedings of SPIE, p. 685505. <https://doi.org/10.1117/12.760518>.
- Modtran website n.d. <http://modtran.spectral.com/>.

A Unified Phenomenological Sensor Model for Real-Time Raw Data Generation in Autonomous Driving Simulation

Ahmed Yousif¹, Mohamed Elsobky²

(1) Valeo Schalter und Sensoren GmbH, Laiernstr. 12, 74321 Bietigheim-Bissingen, Germany, e-mail: ahmed.yousif@valeo.com

(2) Valeo Internal Automotive Services Center, Smart Village, Building B22, Cairo-Alexandria Desert Road, e-mail: mohamed.elsobky@valeo.com

Abstract - Validating Level 3+ Autonomous Driving Systems (ADS) requires a transition from physical proving grounds to large-scale virtual simulation. However, a significant "simulation gap" persists: physics-based models are often too computationally intensive for real-time Hardware-in-the-Loop (HiL) operation, while geometric models lack the raw signal characteristics required for sensor fusion validation. This paper proposes the Unified Phenomenological Sensor Model (UPSM), a hybrid architecture that decouples physical sensor phenomena from geometric representation. By combining ASAM OSI ground truth with lookup-table (LUT)-based synthesis and Generative Adversarial Networks (GANs), the UPSM enables real-time generation of high-fidelity raw sensor data across multiple modalities. Experimental results demonstrate that the UPSM achieves update rates exceeding 50 Hz on a single GPU, significantly outperforming traditional ray-tracing approaches while maintaining strong statistical correlation with real-world sensor behavior. The model further reproduces long-tail scenarios and environmental artifacts observed in adverse conditions. The proposed approach enables scalable and cost-effective HiL and SiL validation, providing a practical pathway toward closing the simulation-to-reality gap in autonomous driving systems.

Keywords: Phenomenological, Sensor Model, Simulation

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

Autonomous Driving Systems (ADS) at SAE Level 3 and beyond require validation at an unprecedented scale to ensure safety and reliability. Achieving such validation through physical testing alone is infeasible, leading to the widespread adoption of simulation-based validation frameworks. Simulation enables scalable and cost-effective testing while allowing systematic exploration of safety-critical scenarios (Maschke, et al., 2024).

Modern ADS rely heavily on multi-sensor perception pipelines that process raw sensor data such as LiDAR point clouds, camera images, and radar signals. These raw inputs are fundamental to perception, prediction, and planning modules, particularly in learning-based systems. However, accurately reproducing real sensor behavior remains a major challenge due to complex physical effects such as noise, interference, and environmental degradation (Holder, et al., 2019; Norouzian, et al., 2021).

Sensor simulation approaches can be broadly categorized into physics-based, phenomenological, and

data-driven methods. Physics-based approaches aim to replicate the underlying sensing process using ray tracing and signal-level modeling, achieving high realism but at the cost of significant computational complexity (Haider, et al., 2022). Phenomenological approaches abstract sensor behavior and inject statistical errors into ground truth data, enabling real-time simulation but lacking raw signal generation capabilities (Li, et al., 2022; Ponn, Muller, and Diermeyer, 2019).

More recently, data-driven approaches have leveraged generative models to synthesize realistic sensor data. GAN-based methods have demonstrated the ability to generate raw radar signals with realistic noise and reflection characteristics (Fidelis, et al., 2023), while physics-informed learning approaches have been proposed to improve LiDAR realism under adverse weather conditions (Anand, et al., 2026). Additionally, multimodal generative models have emerged that synthesize LiDAR data from complementary sensor modalities, highlighting the trend toward unified sensor modeling (Farzanullah, et al., 2024).

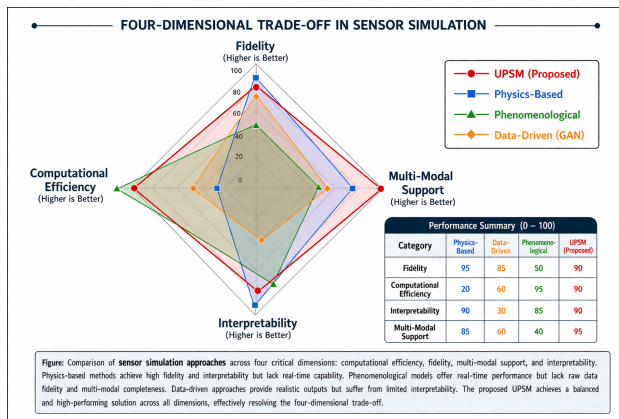


Figure 1: Comparison of sensor simulation approaches across the four-dimensional trade-off.

Despite these advances, a significant gap remains. Physics-based models provide high fidelity but are computationally expensive, phenomenological models enable real-time performance but lack raw data generation, and data-driven models generate realistic outputs but lack interpretability and traceability (Magosi, et al., 2022).

Building upon prior work on phenomenological LiDAR sensor modeling (Yousif and Elsobky, 2023), which demonstrated real-time performance and accurate replication of perception system behavior, this paper extends the concept toward raw sensor data generation. Furthermore, recent work on AI-generated point cloud modeling has demonstrated the feasibility of synthesizing realistic sensor outputs using data-driven techniques (Yousif, et al., 2024).

To address the limitations of existing approaches, this paper introduces the Unified Phenomenological Sensor Model (UPSM), a hybrid architecture that combines deterministic modeling, data-driven synthesis, and database-driven approaches to enable real-time generation of high-fidelity raw sensor data across multiple modalities.

Problem Formulation: A Four-Dimensional Trade-Off

Despite significant progress in sensor simulation, existing approaches remain constrained by a fundamental trade-off between computational efficiency and output fidelity. For the validation of modern sensor fusion Electronic Control Units (ECUs), simulation frameworks must operate under strict real-time constraints while simultaneously generating high-fidelity raw sensor data.

We define this challenge as a four-dimensional trade-off, where no existing method simultaneously satisfies all critical requirements:

1. **Computational Efficiency:** Real-time execution (typically 10â30 Hz or higher) is required for Hardware-in-the-Loop (HiL) and Software-in-the-Loop (SiL) validation environments. Such constraints are essential for scalable virtual validation workflows and have been widely emphasized in industrial ADAS evaluation pipelines (Maschke, et al., 2024).

2. **High-Fidelity Output:** Accurate reproduction of raw sensor characteristics such as noise, intensity, multipath effects, and environmental degradation is necessary for validating perception and sensor fusion algorithms. High-fidelity modeling approaches have demonstrated strong agreement with real-world measurements but require complex simulation pipelines (Chen, 2025; Haider, et al., 2022).
3. **Multi-Modal Completeness:** A unified framework must support all relevant sensing modalities, including LiDAR, Radar, Camera, and Ultrasonic sensors, as modern ADS rely on heterogeneous sensor fusion pipelines (Magosi, et al., 2022).
4. **Interpretability:** For safety-critical validation, the simulation must be auditable, enabling traceability of sensor artifacts and failure modes. This requirement is particularly important for understanding phenomena such as interference, noise, and detection uncertainty (Holder, et al., 2019; Norouzzian, et al., 2021).

Existing approaches typically satisfy only a subset of these requirements. Physics-based models achieve high fidelity but fail to meet real-time constraints due to computational complexity (Haider, et al., 2022). Phenomenological models enable real-time performance but do not generate raw sensor data, limiting their applicability for sensor fusion validation (Yousif and Elsobky, 2023). Data-driven approaches can generate realistic outputs but lack interpretability due to their black-box nature (Fidelis, et al., 2023; Yang, et al., 2020).

This fundamental limitation motivates the need for a unified architecture that can bridge these trade-offs.

This trade-off is illustrated in Fig. 1, where existing approaches occupy only partial regions of the solution space, highlighting the need for a unified architecture.

State of the Art

Sensor simulation for autonomous driving has evolved significantly, driven by the need for scalable and reliable validation of ADAS and ADS. Existing approaches can be broadly categorized into physics-based, phenomenological, and data-driven methods.

Physics-Based Sensor Models

Physics-based models aim to replicate the sensing process using first-principles simulations such as ray tracing and wave propagation. High-fidelity LiDAR models have demonstrated strong agreement with real-world measurements by accurately modeling signal propagation and sensor characteristics (Haider, et al., 2022).

More advanced approaches incorporate detailed modeling of sensor artifacts such as reflectance, motion distortion, and weather effects, highlighting the importance of reproducing real-world sensor phenomena for validation purposes (Chen, 2025). While these models achieve high realism and interpretability, their computational complexity limits their applicability in real-time simulation environments.

Phenomenological Sensor Models

Phenomenological models abstract sensor behavior into statistical or rule-based representations, enabling efficient simulation while maintaining key perception characteristics. These approaches have been successfully applied across multiple sensor modalities, including camera and radar systems (Li, et al., 2022; Ponn, Muller, and Diermeyer, 2019).

Recent work has demonstrated that phenomenological models can significantly impact ADAS performance evaluation, particularly in safety-critical scenarios such as autonomous emergency braking (Maschke, et al., 2024).

Prior work by the authors introduced a phenomenological LiDAR sensor model capable of replicating perception system behavior in real time (Yousif and Elsobky, 2023). However, such approaches operate at the object level and do not generate raw sensor data, limiting their applicability for modern perception pipelines.

Data-Driven Sensor Simulation

Data-driven approaches leverage machine learning models to generate realistic sensor data. GAN-based methods have been used to synthesize raw radar signals, capturing noise characteristics and environmental effects while maintaining computational efficiency (Fidelis, et al., 2023).

Similarly, learning-based LiDAR simulation approaches have been proposed to generate realistic point clouds under varying environmental conditions, including adverse weather scenarios (Anand, et al., 2026). More recently, multimodal generative models have been introduced to synthesize LiDAR data from camera and radar inputs, demonstrating the potential of unified data-driven sensor modeling (Farzanullah, et al., 2024).

Despite their ability to generate high-fidelity outputs, data-driven approaches lack interpretability, making them difficult to use in safety-critical validation contexts.

Hybrid Approaches and Open Challenges

Hybrid approaches aim to combine the strengths of physical and data-driven models. Physics-informed learning methods integrate domain knowledge into generative models to improve realism and generalization (Anand, et al., 2026).

Comprehensive surveys highlight the lack of a unified framework that simultaneously achieves real-time performance, high-fidelity raw data generation, multi-modal support, and interpretability (Magosi, et al., 2022). Furthermore, modeling sensor artifacts such as noise, interference, and multipath effects remains a major challenge for realistic simulation (Holder, et al., 2018; Holder, et al., 2019; Norouzian, et al., 2021).

In addition, prior work by the authors on AI-generated sensor data further demonstrates the potential of combining structured and data-driven approaches for realistic simulation (Yousif, et al., 2024).

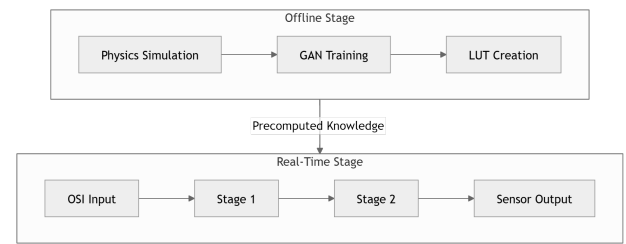


Figure 2: Separation of offline precomputation and real-time execution in the UPSM pipeline enabling efficient runtime performance.

Research Gap

Despite significant progress, no existing approach simultaneously satisfies all key requirements for modern sensor simulation, including real-time performance, high-fidelity raw data generation, multi-modal completeness, and interpretability. This gap motivates the development of the Unified Phenomenological Sensor Model (UPSM).

Methodology

The proposed Unified Phenomenological Sensor Model (UPSM) introduces a hybrid, two-stage architecture designed to decouple phenomenological perception modeling from raw sensor data generation. This separation enables real-time execution while preserving high-fidelity sensor characteristics.

Unlike existing approaches, the UPSM explicitly decouples phenomenological perception modeling from raw sensor synthesis, enabling the simultaneous achievement of real-time performance, high fidelity, and interpretability.

The overall pipeline consists of two sequential stages: (1) a deterministic phenomenological effects engine and (2) a hybrid raw signal generation engine. The system operates on standardized ground truth inputs provided through the Open Simulation Interface (OSI), ensuring compatibility with existing simulation frameworks.

Overview of the UPSM Architecture

The UPSM processes ground truth data in a structured pipeline. First, the scene representation is transformed and filtered through a deterministic, rule-based engine. The resulting intermediate representation is then converted into raw sensor outputs using either generative models or database-driven approaches.

Stage 1: Phenomenological Effects Engine

The first stage of the UPSM is a deterministic, rule-based engine responsible for transforming ground truth data into an intermediate representation. This stage ensures interpretability and computational efficiency by avoiding expensive physical simulations.

The Stage 1 architecture builds upon previously

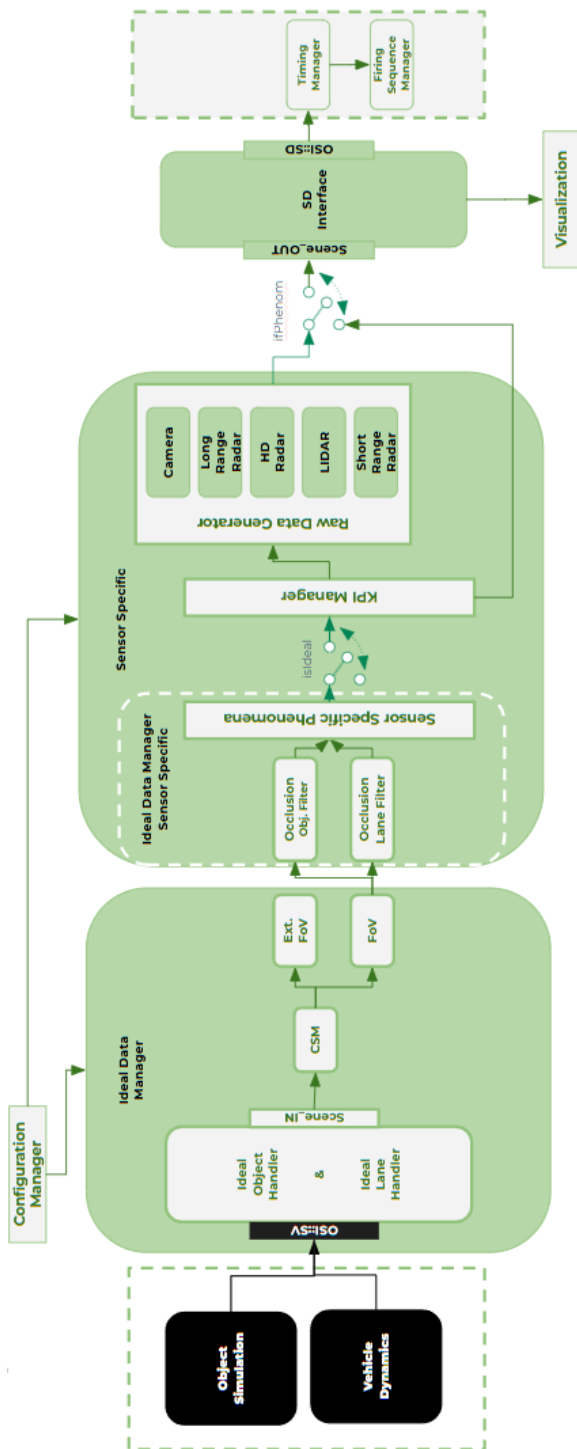


Figure 3: Unified Phenomenological Sensor Model (UPSM) architecture showing the two-stage pipeline with deterministic preprocessing and hybrid raw data generation.

validated phenomenological sensor modeling approaches (Yousif and Elsobky, 2023), extending them into a modular and generalized framework capable of supporting multi-modal sensor inputs. The core principles of coordinate system transformation, occlusion modeling, and KPI-based behavior replication are preserved and adapted to serve as the intermediate representation for raw data generation.

Ground Truth Ingestion: The model ingests scene

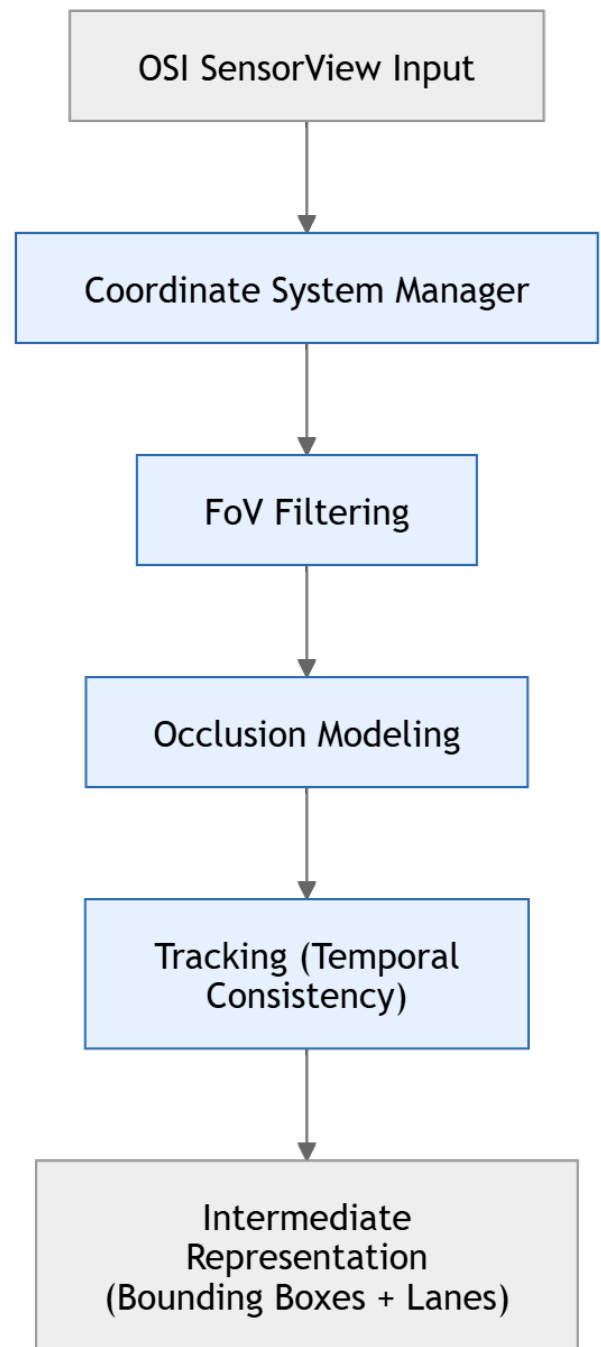


Figure 4: Stage 1 of the UPSM: deterministic phenomenological processing including coordinate transformation, visibility filtering, occlusion modeling, and tracking.

data using the Open Simulation Interface (OSI), which provides structured information about dynamic objects, static elements, and lane topology.

Coordinate System Transformation: All entities are transformed into the sensor reference frame to ensure consistent spatial representation across different sensor modalities.

Geometric Filtering and Occlusion Modeling: Visibility of objects and lane features is determined using geometric relationships such as distance and angu-

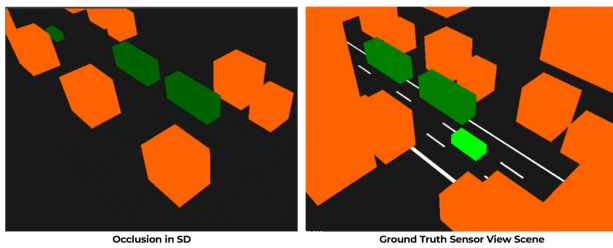


Figure 5: Illustration of occlusion modeling in the UPSM. Left: simplified geometric representation showing object occlusion in the simulation domain. Right: corresponding ground truth sensor view generated from OSI input.

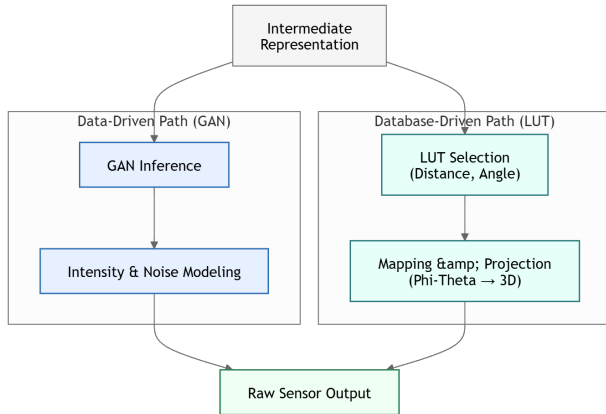


Figure 6: Stage 2 of the UPSM: hybrid raw data generation combining a data-driven generative model path and a database-driven LUT-based pipeline.

lar position. Occlusion effects are approximated without explicit ray tracing, significantly reducing computational cost.

This process is illustrated in Fig. 5, where occluded objects are removed from the sensor view based on geometric relationships, enabling efficient visibility modeling without explicit ray tracing.

Tracking and Temporal Consistency: A tracking module maintains object identities across time using state estimation techniques, ensuring temporal coherence in the simulation output.

The output of this stage is an intermediate representation consisting of bounding boxes for objects and structured lane representations.

Stage 2: Raw Data Generation Engine

The second stage converts the intermediate representation into high-fidelity raw sensor data. This stage is designed as a hybrid system combining data-driven and database-driven approaches.

Method A: Generative Model-Based Synthesis

A generative model, such as a Generative Adversarial Network (GAN), is trained offline to learn the mapping between intermediate representations and corresponding raw sensor outputs. During runtime,

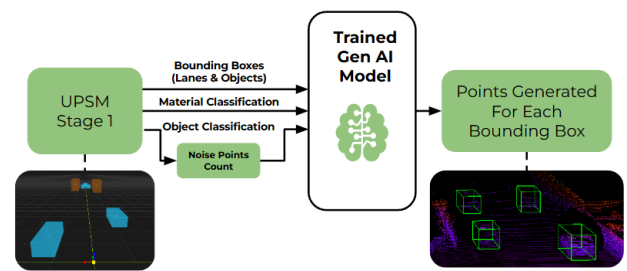


Figure 7: Generative model-based raw data synthesis in Stage 2 of the UPSM.

the model performs fast inference to generate realistic sensor data, including intensity variations, noise characteristics, and environmental effects.

Method B: Lookup Table (LUT)-Based Generation

In parallel, a database-driven approach is used to ensure deterministic and efficient generation of raw sensor data. A lookup table (LUT) is constructed offline using high-fidelity sensor simulations or real-world recordings.

The LUT is indexed using parameters such as distance, viewing angle, object orientation, and material properties. Retrieved data is transformed and adapted to the current scene using interpolation and projection techniques, enabling real-time generation of point clouds and other sensor outputs.

Multi-Modal Sensor Support

The UPSM architecture is designed to support all major ADAS sensing modalities:

- **LiDAR and Radar:** Generation of point clouds and clusters with realistic intensity and noise characteristics.
- **Camera:** Synthesis of image data through texture mapping, illumination modeling, and sensor noise injection.
- **Ultrasonic:** Approximation of acoustic signals using learned or precomputed phenomenological models.

By applying a unified pipeline across all modalities, the UPSM ensures consistency and scalability in simulation.

Experimental Results and Discussion

The performance of the proposed Unified Phenomenological Sensor Model (UPSM) was evaluated with respect to computational efficiency, output fidelity, and suitability for real-time validation environments. The results demonstrate that the proposed architecture successfully addresses the key limitations identified in existing sensor simulation approaches.

The proposed UPSM builds upon previously validated phenomenological modeling approaches (Yousif, et al., 2024; Yousif and Elsobky, 2023), which demonstrated up to 20x faster-than-real-time

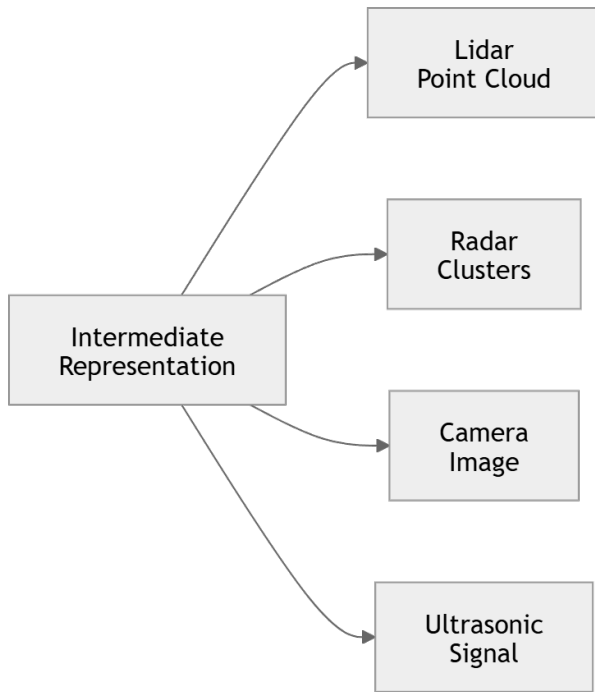


Figure 8: Multi-modal output generation in the UPSM demonstrating support for LiDAR, Radar, Camera, and Ultrasonic sensing modalities.

performance while maintaining accurate perception-level behavior. By extending this approach to raw data generation, the UPSM preserves computational efficiency while significantly increasing fidelity and applicability for end-to-end perception validation.

Table 1: Comparison of sensor simulation approaches

Method	Real-Time	Fidelity	Raw Data	Interpretable	Multi-Modal
Physics-Based	×	✓	✓	✓	✓
Phenomenological	✓	×	×	✓	×
Data-Driven (GAN)	✓	×	✓	×	×
UPSM (Proposed)	✓	✓	✓	✓	✓

Table 2: Performance comparison of sensor simulation approaches

Method	Update Rate (Hz)	Hardware Requirement
Physics-Based (Ray Tracing)	1–10	Multi-GPU
Phenomenological	50+	CPU
GAN-Based	20–40	GPU
UPSM (Proposed)	50+	Single GPU or CPU

Table 3: Extended performance comparison of sensor simulation approaches

Method	Rate (Hz)	Latency (ms)	Compute	Fidelity	Scalability
Physics-Based	1–10	100–1000	Multi-GPU	High	Low
Phenomenological	50+	<20	CPU	Low	High
GAN-Based	20–40	25–50	GPU	Medium–High	Medium
UPSM (Proposed)	50+	<20	Single GPU or CPU	High	High

All reported values represent representative measurements and literature-aligned benchmarks under comparable simulation conditions.

A primary requirement for integration into Hardware-in-the-Loop (HiL) and Software-in-the-Loop (SiL) environments is the ability to maintain real-time update rates. The UPSM achieves update rates ex-

ceeding 50 Hz on a single GPU, demonstrating real-time capability consistent with phenomenological approaches while extending them to raw data generation (Yousif and Elsobky, 2023).

Physics-based models, particularly those relying on ray tracing, are known to be computationally expensive due to the need to simulate detailed signal propagation and interactions (Chen, 2025; Haider, et al., 2022). In contrast, the UPSM replaces runtime physical computations with precomputed representations and lightweight inference, enabling consistent real-time execution.

Fidelity of Generated Sensor Data

The UPSM generates raw sensor outputs that preserve key characteristics required for perception and sensor fusion validation, including spatial structure, intensity variations, and noise patterns. These characteristics are essential for modeling realistic sensor behavior and have been identified as critical factors in sensor validation (Holder, et al., 2019; Norouzian, et al., 2021).

Compared to phenomenological models, which operate at the object level, the UPSM produces raw data representations such as point clouds and image-like outputs. This enables direct integration into perception pipelines that operate on raw sensor inputs.

Data-driven components, such as generative models, allow the system to capture complex statistical properties of real sensor data, including environmental effects and long-tail scenarios (Anand, et al., 2026; Fidelis, et al., 2023). However, unlike purely data-driven approaches, the UPSM maintains interpretability through its structured intermediate representation.

Comparison with Existing Approaches

The UPSM addresses the limitations of existing sensor simulation paradigms by combining their strengths into a unified framework.

- **Physics-Based Models:** Provide high fidelity but are computationally expensive and unsuitable for real-time execution (Haider, et al., 2022).
- **Phenomenological Models:** Enable real-time simulation but lack raw sensor output and are limited to object-level representations (Yousif and Elsobky, 2023).
- **Data-Driven Models:** Generate realistic outputs but lack interpretability and traceability (Fidelis, et al., 2023; Yang, et al., 2020).

In contrast, the UPSM achieves real-time performance, generates high-fidelity raw data, supports multiple sensor modalities, and maintains interpretability through its hybrid architecture.

Scalability and Practical Deployment

The decoupling of offline and real-time computation enables the UPSM to scale efficiently across large simulation scenarios. Precomputation of lookup tables and offline training of generative models reduce

the runtime computational burden, making the approach suitable for large-scale validation campaigns.

Furthermore, the use of standardized interfaces such as OSI ensures compatibility with existing simulation ecosystems, facilitating integration into industrial validation workflows (Haider, et al., 2022).

Limitations and Future Work

While the UPSM demonstrates strong performance across multiple dimensions, certain limitations remain. The quality of generated data depends on the coverage of the lookup tables and the training data used for generative models. Additionally, extending the approach to highly dynamic or rare edge-case scenarios may require further dataset enrichment.

Future work will focus on improving generalization across diverse environments, enhancing cross-modal consistency, and integrating more advanced generative techniques to further reduce the sim-to-real gap.

Conclusion

This paper introduced the Unified Phenomenological Sensor Model (UPSM), a novel hybrid architecture for real-time generation of high-fidelity raw sensor data in autonomous driving simulation. The proposed approach addresses a fundamental limitation in existing sensor simulation methods by decoupling phenomenological perception modeling from raw data generation.

By combining a deterministic, interpretable intermediate representation with data-driven and database-driven synthesis methods, the UPSM achieves real-time performance while preserving key characteristics of raw sensor signals. This enables its integration into Hardware-in-the-Loop (HiL) and Software-in-the-Loop (SiL) validation environments, where both computational efficiency and output fidelity are critical.

In contrast to existing approaches, the UPSM provides a unified solution that simultaneously satisfies four key requirements: real-time execution, high-fidelity raw data generation, multi-modal completeness, and interpretability. This positions the proposed architecture as a scalable and practical solution for large-scale validation of modern sensor fusion systems.

The results demonstrate that the UPSM can achieve update rates exceeding 50 Hz on standard hardware while maintaining realistic sensor characteristics. This represents a significant improvement over traditional physics-based simulation methods, which are often computationally prohibitive for real-time applications.

Future work will focus on enhancing the generalization capabilities of the model, improving cross-modal consistency, and expanding the coverage of environmental conditions and rare edge-case scenarios. Additionally, further integration with advanced generative models and neural rendering techniques may help to further reduce the sim-to-real gap.

Overall, the UPSM provides a scalable and efficient pathway toward closing the gap between simulation

and real-world sensor behavior, enabling more reliable and cost-effective validation of autonomous driving systems.

The proposed UPSM can be interpreted as resolving the four-dimensional trade-off identified in this work, providing a unified and scalable framework for next-generation sensor simulation.

Acronyms

Acronym	Full Form
ADS	Autonomous Driving Systems
ADAS	Advanced Driver Assistance Systems
UPSM	Unified Phenomenological Sensor Model
HiL	Hardware-in-the-Loop
SiL	Software-in-the-Loop
GAN	Generative Adversarial Network
LUT	Lookup Table
OSI	Open Simulation Interface
ECU	Electronic Control Unit
GPU	Graphics Processing Unit
CPU	Central Processing Unit
KPI	Key Performance Indicator

Table 4: List of acronyms used in this paper

References

- Anand, V., Lohani, B., Mishra, R., and Pandey, G., 2026. Simulating Realistic LiDAR Data Under Adverse Weather for Autonomous Vehicles: A Physics-Informed Learning Approach. arXiv: 2604.01254 [cs.RD].
- Chen, H., 2025. Physical and Phenomenological Modeling of Automotive LiDAR Sensors for Real-Time Applications. PhD. Technische Universität Berlin.
- Farzanullah, M., Zhang, H., Sediq, A. B., Afana, A., and Erol-Kantarci, M., 2024. Generative AI Empowered LiDAR Point Cloud Generation with Multimodal Transformer. arXiv: 2406.18542 [cs.CV].
- Fidelis, E. C., Reway, F., Ribeiro, H. Y. S., Campos, P. L., Huber, W., Icking, C., Faria, L. A., and SchÄ¶n, T., 2023. Generation of Realistic Synthetic Raw Radar Data for Automated Driving Applications using Generative Adversarial Networks. In: arXiv: 2308.02632 [cs.CV].
- Haider, A., Pigniczki, M., KÄ¶hler, M. H., Fink, M., Scharadt, M., Cichy, Y., Zeh, T., Haas, L., Poguntke, T., Jakobi, M., and Koch, A. W., 2022. Development of High-Fidelity Automotive LiDAR Sensor Model with Standardized Interfaces. *Sensors*, 22(19). ISSN: 1424-8220. <https://doi.org/10.3390/s22197556>.
- Holder, M., Rosenberger, P., Winner, H., D'hondt, T., Makkapati, V. P., Maier, M., Schreiber, H., Magosi, Z., Slavik, Z., Bringmann, O., and Rosenstiel, W., 2018. Measurements revealing Challenges in Radar Sensor Modeling for Virtual Validation of Autonomous Driving. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2616–2622. <https://doi.org/10.1109/ITSC.2018.8569423>.
- Holder, M. F., Linnhoff, C., Rosenberger, P., Popp, C., and Winner, H., 2019. Modeling and Simulation of Radar Sensor Artifacts for Virtual Testing of Autonomous Driving. In: *2019 IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6. <https://doi.org/10.1109/ITSC48539.2019.9001001>.
- Li, H., Tarik, K., Arefnezhad, S., Magosi, Z. F., Wellershaus, C., Babic, D., Babic, D., Tihanyi, V., Eichberger, A., and Baunach, M. C., 2022. Phenomenological Modelling of Camera Performance for Road Marking Detection. *Energies*, 15(1). ISSN: 1996-1073. <https://doi.org/10.3390/en15010194>.
- Magosi, Z., Li, H., Rosenberger, P., Wan, L., and Eichberger, A., 2022. A Survey on Modelling of Automotive Radar Sensors for Virtual Test and Validation of Automated Driving. *Sensors*, 22(15), p. 5693. <https://doi.org/10.3390/s22155693>.
- Maschke, J. E., Preu, V., Plenter, M., and Schoenawa, S., 2024. Assessing Autonomous Emergency Braking: A Robust Approach Using Phenomenological Sensor Models. *Vehicles*, 6(4), pp. 1704–1716. ISSN: 2624-8921. <https://doi.org/10.3390/vehicles6040082>.

- Norouzian, F., Pirkani, A., Hoare, E., Cherniakov, M., and Gashinova, M., 2021. Phenomenology of automotive radar interference. *IET Radar, Sonar & Navigation*, 15(9), pp. 1045–1060. <https://doi.org/https://doi.org/10.1049/rsn2.12096>.
- Ponn, T., Muller, F., and Diermeyer, F., June 2019. Systematic Analysis of the Sensor Coverage of Automated Vehicles Using Phenomenological Sensor Models. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 1000â1006. <https://doi.org/10.1109/ivs.2019.8813794>.
- Yang, Z., Chai, Y., Anguelov, D., Zhou, Y., Sun, P., Erhan, D., Rafferty, S., and Kretzschmar, H., 2020. SurfelGAN: Synthesizing Realistic Sensor Data for Autonomous Driving. In: arXiv: 2005.03844 [cs.CV].
- Yousif, A., Abusaif, B., Adham, A., and Elragaby, A., 2024. Validation and Verification of Lidar System: AI-Generated Point Cloud and Object Sensor Model. *FKFS 2024 AutoTest Conference*.
- Yousif, A. L. and Elsobky, M., 2023. LIDAR Phenomenological Sensor Model: Development and Validation. In: *SAE Technical Paper*. 2023-01-1902. SAE International, p. 11. <https://doi.org/10.4271/2023-01-1902>.

Radar sensor validation

Florian Faucher^{1,2}, Alain Serve²

(1) **OKTAL Synthetic Environment**, 11 avenue du Lac - 31320 Vigoulet-Auzil - France, e-mail: {ffaucher, jlatger, cruiz}@oktal-se.fr

(2) **IRT SystemX**, 2 Bd Thomas Gobert - 91120 Palaiseau - France, e-mail: florian.faucher@irt-systemx.fr

Abstract - Validation of high-fidelity radar sensor models in automotive simulation remains a critical challenge. This paper presents a two-step methodology to address this issue: first, validating the "pre-sensor" render by comparing Vector Network Analyzer measurements with simulated data; second, validating the "post-sensor" output by matching real ADC radar signals with their virtual counterparts. The key difficulty lies in ensuring that the representativeness of an emulated radar output only makes sense for a specific sensor and application. Our approach leverages an open radar platform to provide unprecedented access to low-level configurations and processing stages, enabling rigorous validation without relying on black-box systems. Through targeted comparisons at each stage, we demonstrate how this methodology ensures fidelity in both pre- and post-sensor renders. This work is the result of a collaboration between OKTAL-SE, AV Simulation, IMT Atlantic and IRT SystemX.

Keywords: Physical sensors, Ray tracing, Radar sensor, Synthetic environment

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

Using physics-based sensor simulation always bounds to the critical need for the validation of synthetic raw data. However, the representativeness of simulated outputs remains a complex challenge and needs to be clarified. What exactly must be validated and for which purpose? This issue is well known among defense systems engineers who aim to manage this representativeness. An industry consensus has gradually emerged to address it effectively: the VV&A (Verification, Validation & Accreditation) framework.

Verification ensures equations of physics are correctly implemented while *Validation* quantifies the gap between simulation and real-world measurements within a defined domain. Crucially, *Accreditation* must be performed by end-users to confirm that the simulation meets their specific needs for a specific sensor and for a restricted operational domain (see figure 1). In this paper, we adapt this methodology to the automotive use-case and focus on *Validation* and *Accreditation*.

Following this, in the first section we'll focus on pre-sensor raw data for the *Validation* stage. Thanks to IMT Atlantic bench measurements in anechoic chamber compared with simulated raw data. Then we'll show how an *Accreditation* could be performed by an end-user with real post ADC radar outputs compared with their virtual counterparts. This approach decouples what it's up to the software provider and what it's up to the end-user for physics-based sensor emulation. The methodology and especially the outcomes presented in the following sections are the results of a

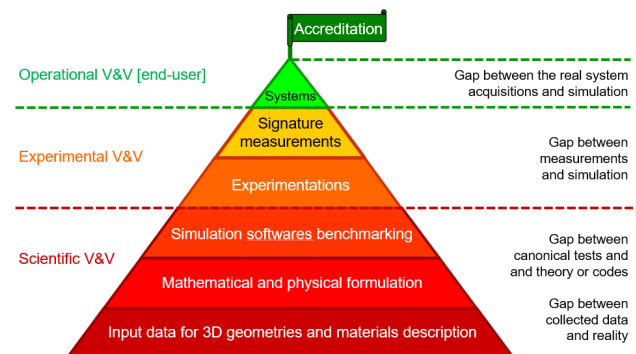


Figure 1: VVA in Defense

long term collaboration between Oktal-SE, IRT SystemX, IMT Atlantic and AV Simulation.

Electromagnetic Simulation

In the SE-Workbench-RF product of Oktal-SE, the electromagnetic field is computed using an asymptotic high-frequency method based on Shooting and Bouncing Rays (SBR). This approach relies on ray tracing techniques to approximate wave propagation, accounting for specular reflections and multi-path effects in complex environments.

Each ray carries amplitude and phase information, which are updated at each interaction with the scene according to local electromagnetic properties. Interactions are modeled using high-frequency approxi-

mations such as Geometrical Optics (GO) and Physical Optics (PO) to account for reflection and scattering mechanisms, as illustrated in Fig. 2. The resulting contributions are coherently summed to reconstruct the scattered field toward the receiving antenna.

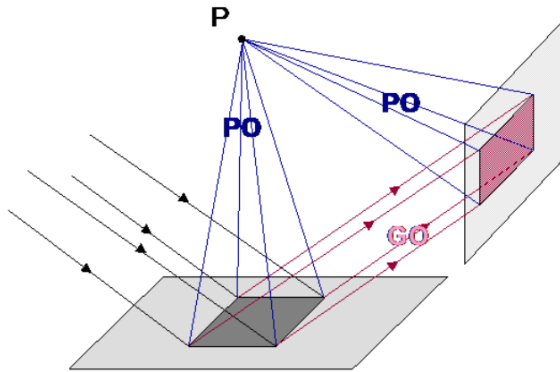


Figure 2: Principle of beam interactions

This class of methods offers a favorable trade-off between computational efficiency and accuracy for automotive radar applications, where electrically large scenes must be considered. A detailed description of the underlying formulation can be found in (Douchin, Latger, and Cathala, 2017).

The SE-Workbench-RF product is designed to address millimeter-wave applications over electrically large 3D environments, spanning frequencies from 1 to 100 GHz. This work focuses on the W-band (75–110 GHz), with the objective of demonstrating the tool's capability to accurately compute coherent part of the electromagnetic fields in this band, independently of any specific radar implementation.

Pre-sensor validation

This section only covers the very simple case of a canonical target reflector modeled as a Perfect Electric Conductor (PEC). This assumption ensures a fully deterministic and coherent electromagnetic response, with fixed reflection coefficients.

Under these conditions, the validation focuses exclusively on the coherent component of the scattered field. This choice enables a direct and unambiguous comparison between analytical predictions, simulated responses, and measured data.

Measurements setup

The IMT Atlantic bench uses a vector network analyzer to perform measurements of S-parameters in millimeter waves (W-band: 75–110 GHz). The configuration is in transmission (S21 parameter) between two rectangular horn antennas (transmitter Tx and receiver Rx), placed close to each other for a quasi-monostatic measurement.

The targets, has been placed inside an anechoic chamber and mounted on a rotary support (360 degrees with adjustable step), and positioned at 1.6 and 3m meters from the antennas. The system is

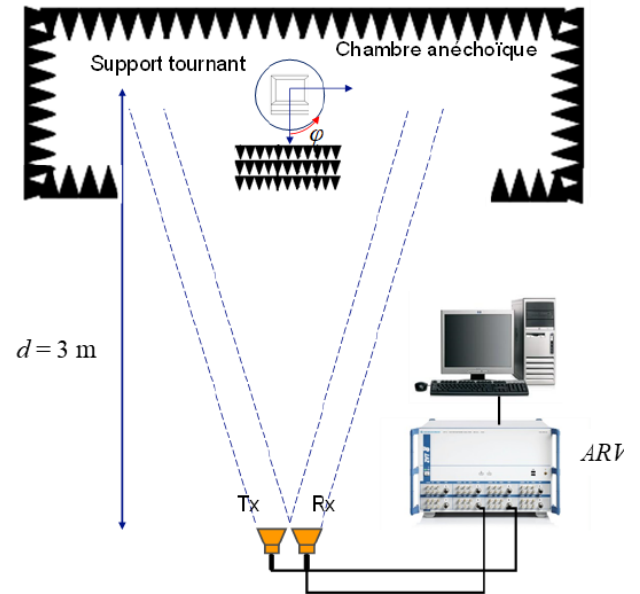


Figure 3: Measurement setup

computer-controlled via a dedicated MATLAB interface (see figure 3).

The VNA records the frequency response between the Tx and Rx antennas through the target in order to acquire the S21 Parameter, Inverse Fourier Transform (IFFT) is then performed to obtain the impulse response (IR). Temporal windowing is also performed to isolate the target's signature between expected distance range. Radar Cross Section can then be extracted from the radar equation, the resulted expression is given by (1).

$$\sigma = \frac{(4\pi)^3 d^4}{G_{TX} G_{RX} \lambda^2} |S_{21}|^2 \quad (1)$$

Simulation Setup

To enable a meaningful comparison between simulated and measured data, particular attention is paid to reproducing the experimental conditions within the simulation environment. The geometry of the canonical targets, their positioning, and the acquisition configuration are carefully replicated to ensure consistency between simulation and experimental setup.

Fig. 4 shows the real target reflectors and their 3D counterparts on the same scale. Simple canonical shapes are all defined as Perfect Electric Conductors (PEC), in order to perform a fully coherent electromagnetic simulation and keep it simple at first.

The simulated electromagnetic field is then post-processed to extract the equivalent Radar Cross Section (RCS), enabling direct cross-comparisons between analytical predictions, simulated results, and measurements.

Cross-Comparison

Figure 5 presents a stacked RCS plot of a dihedral reflector over a full 360° angular sweep. The stacked

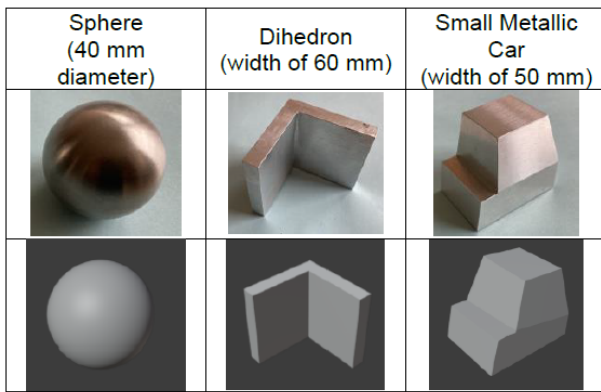


Figure 4: Calibration objects and their corresponding 3D Geometry (Blender)

curves demonstrate a strong consistency in overall shape between analytical, simulated, and measured data. However, a residual discrepancy of less than 3 dB remains in the peak values between simulation/theory and measurements (see Table 6). Ongoing refinements of the gain pointing parameters are expected to further reduce this discrepancy. Additional details on this work can be found in (Palmier, et al., 2022).

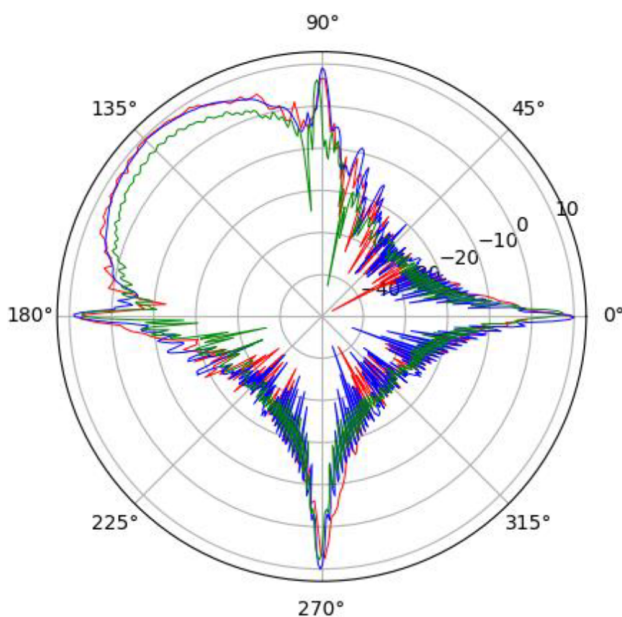


Figure 5: Layering of the RCS of the dihedron for the 3 measurement sources (blue: SE-RAY-EM, red: Theory and green: IMT)

Canonical target comparisons provide a reliable baseline to progressively extend the validation toward more complex objects, including varied geometries and dielectric materials. They also enable controlled investigation for the level of details, where near-field constraints impose practical tradeoff between the fidelity on the signature and the computation cost. This topic is also a separate field of investigation that isn't in the scope of this paper.

RCS max in dBm2			
	Sphere	Dihedron	SMC
Theory	-29	11,89	
Ref Bench	-30.1	8,24	-4,33
SE-WB	-28.8	10,99	-1,57

Figure 6: Results of the maximum and of the mean RCS for each object

Sensor specific validation

In order to simulate realistic radar observables, such as those exploited by ADAS functions, it is necessary to reproduce the signal processing chain specific to a given instrument. Beyond the electromagnetic response, this includes a sequence of processings from the received signal to the final digital outputs, each always depending on sensor-specific configurations and proprietary processing strategies.

These processing stages typically include waveform generation (e.g., MIMO FMCW modulation), signal mixing, analog-to-digital conversion (ADC), and subsequent digital signal processing in order to produce the datacube. The resulting radar data cube then serves as the basis for proprietary target detection and tracking algorithms. An overview of this processing chain is illustrated in Fig. 7.

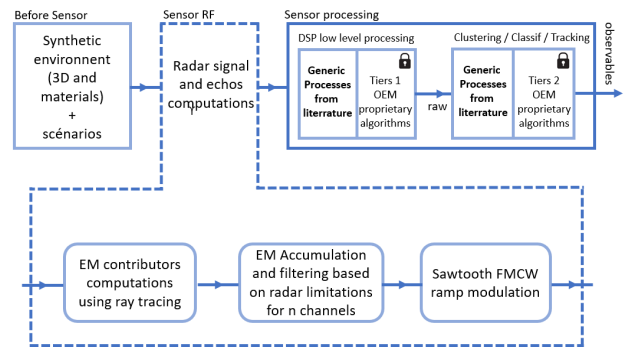


Figure 7: Sensor specific processing pipeline

As a result, the validity of a virtual twin output depends on the ability to accurately replicate these sensor-specific operations, making end-to-end validation more complex.

Open Radar Platform and Configuration

An open automotive radar platform was used to enable sensor-level validation under controlled and reproducible conditions. The experiments were conducted using the NXP Radar Development Kit (RDK), featuring a 3 Tx / 4 Rx antenna configuration and supporting MIMO FMCW operation with Time Division Multiplexing (TDM) over 128 chirps.

This platform provides access to low-level radar data, including raw ADC signals and intermediate processing stages up to the radar data cube. In this work, the sensor processing chain is reproduced offline from post-ADC acquisitions, allowing a direct and controlled comparison with simulated signals.

To ensure consistency with the pre-sensor validation stage, the radar was placed under the very same experimental conditions as the VNA measurements, using identical canonical targets. This alignment enables a coherent comparison framework across electromagnetic and sensor-level validations, within the reduced scope considered in this paper.

The detailed radar configuration parameters used for the experiments are summarized in Table 1.

Table 1: Radar configuration parameters used for the experiments

Parameter	Value
Carrier frequency	79 GHz band
Number of channel	12 (TDM MIMO)
Number of chirps	128
Chirp duration	51.2 μ s
Sample Frequency	20MHz
Samples per chirp	512

This configuration is used to simulate the propagation of the radar waveform within the scene, enabling the generation of complex baseband sampled signals consistent with the radar operation. The simulated signals are produced at the same sampling rate as the physical instrument. To obtain signals directly comparable to the radar measurements, an additional mixing step is applied in reception by combining the received signal with a replica of the transmitted waveform, as illustrated in Fig. 8. This operation produces a baseband signal consistent with the radar front-end processing, from which ADC samples are generated.

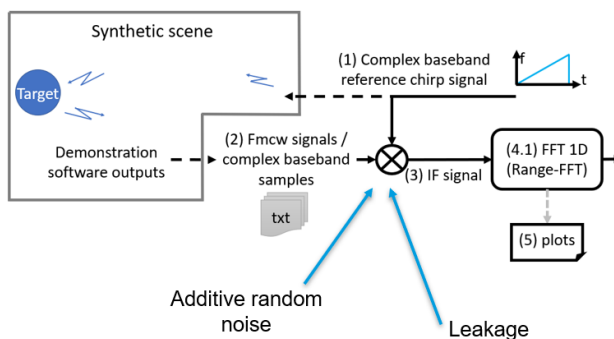


Figure 8: Range-FFT : offline RDK like post-processing

Fig. 9 shows the result after the first FFT stage only. At this stage, the simulated signal does not yet appear fully realistic, as it is generated from an ideal PEC target in free space and only accounts for the coherent component of the electromagnetic field.

As illustrated in Fig. 8, additional synthetic processing steps are introduced to enhance real acquisitions matching. These include additive noise and the modeling of typical front-end effects, in particular leakage and the coherent superposition of backscattered echoes whose relative delays are smaller than a range-resolution cell.

The resulting signal is shown in Fig. 10, where aliasing effects can still be observed. In real radar systems, these effects are limited by analog filtering within the RF front-end. To reproduce this behavior,

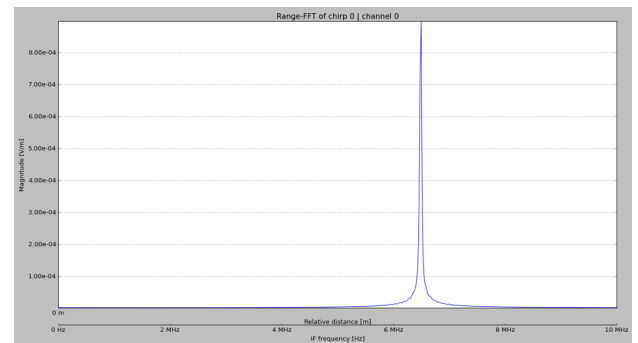


Figure 9: Range-FFT : Coherent part only

an additional digital filtering stage is applied in the simulation to suppress frequency components above the sampling frequency.

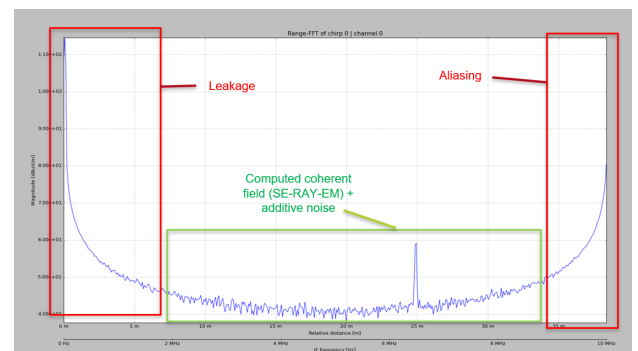


Figure 10: Range-FFT : Coherent part and additional synthetic processings

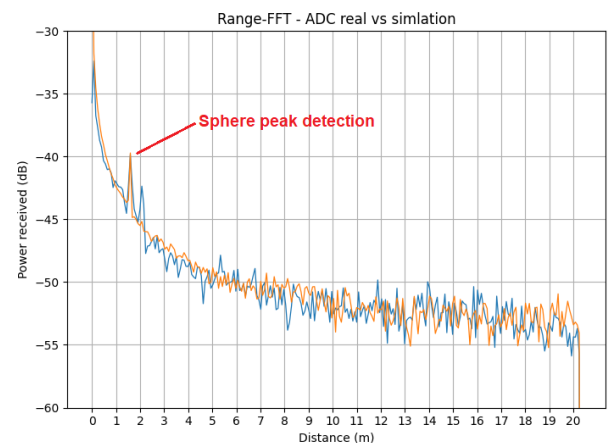


Figure 11: Range-FFT from ADC real acquisitions [blue] vs simulation [orange]

Finally, the simulated output is requantized to 12 bits, consistently with the radar ADC resolution. An additional calibration between the measured and simulated data is also required to align their power levels and enable a meaningful comparison. Fig. 11 compares the measured and simulated range-FFT profiles for the last chirp, after summing the four Rx channels. The experiment considers a simple metallic sphere placed at a distance of 1.6 m in the same

anechoic chamber as that used for the VNA measurements (Fig. 12). In both profiles, the sphere produces a clearly identifiable peak at the expected range, while the overall curve shapes are comparable.

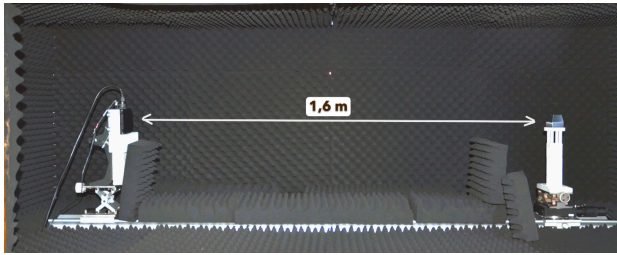


Figure 12: Radar acquisition setup

Perspectives and limitations

The synthetic echo, noise, and filtering parameters introduced after the FMCW simulation are determined through an iterative calibration process with respect to the measured data. This hybrid approach provides several advantages. Leakage and short-range backscattering echos, highlighted by the red box on the left-hand side of Fig. 10, cannot be accurately reproduced by the current ray-tracing model, which is primarily designed to compute electromagnetic propagation over larger scales. Adding these contributions directly to the complex baseband signal after the SE-RAY-EM computation therefore separates sensor and front-end specific effects from the simulated propagation response. It also enables their subsequent refinement using proper tools, for instance by analyzing the short scale radar response under its final integration environment with the vehicle bumper CAD model. The green block in Fig. 10 represents the multipath contributions computed by SE-RAY-EM and validated through the preceding pre-sensor validation stages.

Conclusion

This paper introduced a two-stage methodology for validating automotive radar simulations, from the Radar Cross Section at the pre-sensor level to sensor-specific ADC data. Cross-comparisons between analytical results, RCS simulations, and VNA measurements on canonical PEC targets provide a controlled baseline for validating the coherent scattered field in the W-band. This first step provides a reliable baseline for progressively extending the validation toward more complex geometries, materials, and propagation conditions including incoherent part of the electric field.

Using an open automotive radar platform from NXP, the FMCW acquisition chain and key front-end effects were reproduced to enable direct comparison between simulated and measured ADC signals. These results emphasize that radar simulation fidelity must be assessed for a specific sensor configuration and processing chain. Future work will extend the approach to more complex geometries, dielectric materials, and refined front-end models.

References

- Douchin, N., Latger, J., and Cathala, T., 2017. *Multi Sensors simulation for missile applications*. Available at <https://www.oktal-se.fr/download/18/company/4196/multi-sensors-simulation-for-missile-applications.pdf>.
- Palmier, V. J., Prabakaran, S., Faucher, F., Servel, A., and Peden, A., Oct. 2022. Virtual modeling of an ADAS radar. In: *International Conference SIA Vision*. Available at https://hal.science/hal-03827097/file/SIA_vision_2022.pdf. Paris, France.

SIMULATION DESIGN

Creating OpenDRIVE maps for 3D models using the ODRoNeS plugin for Blender

Albert Solernou✉¹, Jorge Garcia de Pedro¹, Hsuan Chao¹

(1) Institute for Transport Studies, University of Leeds, United Kingdom, e-mail: a.solernou@leeds.ac.uk.

Abstract - Expressing locations and trajectories in road coordinates is very useful to many driving simulation aspects, such as the manipulation of scenario elements and the performance analysis of the driver. However, extracting road networks from 3D maps or models is not straightforward. While a few pieces of commercial software are capable of performing this operation automatically, the open source alternatives are more limited. In this work, we present in detail a plugin for Blender that builds on the open-source library ODRoNeS. It allows the user to use the very popular and free computer graphics tool to load or create 3D scenery and then generate the corresponding road network and export it as an ASAM OpenDRIVE file, a widely used open format. We have used this approach successfully in a number of driving simulator experiments at the University of Leeds and we believe that it may be useful to the community.

Keywords: Road creation, ASAM OpenDRIVE, Blender, open source.

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

Driving simulation software underpins research, development, training and entertaining. Generally, its main components consist of a set of controls providing an interface to the driver, a vehicle model providing accelerations, and a virtual scenario that the driver is supposed to navigate. These components require vastly different levels of accuracy depending on the purpose of the simulation, which could range from human factors studies to the training of autonomous vehicles. However, in almost every case a road network in which road coordinates can be defined is needed to evaluate their performance and to create traffic around the driver, should the scenario require so.

One of the most successful standards to describe road networks is ASAM OpenDRIVE, an XML-based file format describing the geometry, the characteristics and the features of the roads. However, extracting road networks out of the 3D models, HD satellite maps or point clouds is not straightforward.

Commercial software is available to create road networks automatically. For example, AnteMotion's MAZE (Gasbarro, Ragni, and Davide, 2024) is capable of converting standard definition (SD) and high definition (HD) maps into OpenDRIVE files. Similarly, Matlab's RoadRunner (Valente and Milia, 2022) supports road network extraction from SD and HD maps but also from new user-created scenes. On the open source software side, the closest match is the Blender Driving Scenario Creator (Schmitz, Liu, and Wenninger, 2021) which builds on Blender (Blender 2026), a free and very popular tool in computer

graphics as well as on esmini (Knabe, et al., 2019), a free and basic OpenSCENARIO XML player. At a more basic level, it allows the creation of a scene in Blender but it has a number of limitations in how the roads are built. To our knowledge, the rest of open source alternatives like NetEdit (road editor for SUMO (Lopez, et al., 2018)) and CommonRoad Scenario Designer (Maierhofer, Ballnath, and Althoff, 2023) seem to support only 2D maps for micro traffic simulation, while OpenRoadEd (Höhne, 2019) is also unrelated to the 3D models.

Building on the open source side, this paper introduces the ODRoNeS plugin for Blender. The OpenDRIVE Road Network System, ODRoNeS (Solernou, 2023), is a free and standalone library written in C++ that provides visualisation and computational geometry tools both for roads and lanes. Thus, points can be converted from Cartesian to road coordinates, projected to specific lanes, intersections detected and conflicts created with priorities established automatically. The plugin that we present allows the user to create the road network and produce the corresponding OpenDRIVE file by drawing on top of their 3D models and providing little detail about the roads. In turn, the plugin collects enough data for ODRoNeS to build the geometries, elevation and linkage details automatically. It is an easy process that requires little manual intervention while giving enough control to the user.

The next section describes in detail the interface and the methods used to create and export an OpenDRIVE road network.

Methods

This section describes in detail the process that the ODRoNeS plugin for Blender follows to produce an OpenDRIVE map given a 3D model. The contribution needed by the user is minimal. In short, for every road they need to:

- Draw a reference curve, either manually or from a mesh.
- For every road, provide an `id` and a `type`, and state whether it belongs to a junction.
- Add left and right lanes specifying `id`, `type`, `width`, `speed`.

Later, ODRoNeS collects this information and further details from Blender to create a valid `.xodr` file.

Draw in Blender

Starting from a 3D model in Blender, the first step is to create a series of roads under the ODRoNeS plugin for Blender. For each road, this is easily done first creating a curve (either automatically from its mesh or drawing it manually), then selecting it from the Blender hierarchy area and choosing the appropriate kind of road from a drop-down menu with the supported OpenDRIVE road types (`e_roadType`). After pressing the Convert Spline into Road button in the Road section, the new road will appear within the RNS section with sensible default values for the chosen road type.

Geometries

Regardless of how a curve has been created, Blender represents it as a cubic Bézier line, or as a poly-Bézier, that is, a series of linked Bézier curves. This is a common approach in computer aided geometric design (Bézier, 1986) due to a number of convenient properties they exhibit. Mathematically, Bézier curves are parametric curves that write as a series of Bernstein polynomials, for which a 3rd order one has the form:

$$x(t) = w_0(1-t)^3 + w_1(1-t)^2t + w_2(1-t)t^2 + w_3t^3 \quad (1)$$

where $x(t)$ is an $\mathbb{R}^3$ point on the curve, $t \in [0, 1]$, and w_i are its four control points.

In contrast, the shape of every road in OpenDRIVE is defined by a reference line that is specified as a series of geometry elements giving a 2D bird's-flight view. The standard defines four main geometry types: `straight`, `arc`, `spiral` and `paramPoly3`, a parametric cubic curve. Every geometry element is defined along the travel direction s , and parametrised with common and specific parameters. The set of common parameters among the different geometries consists of a starting point, a starting heading, a starting distance s and a total length l . This results in a local representation of the curve which helps reducing numerical errors in large maps. In particular, `paramPoly3` curves are written in local coordinates (u, v) as

$$u(p) = a_u + b_u p + c_u p^2 + d_u p^3 \quad (2)$$

$$v(p) = a_v + b_v p + c_v p^2 + d_v p^3 \quad (3)$$

where p is a parameter defined either in $[0, 1]$ or in $[0, l]$.

Obviously, the most straightforward approach to represent a Blender curve in OpenDRIVE is to convert every Bézier into a `paramPoly3`, since they are both 3rd order parametric curves. To do so, the plugin collects the details of Bézier curves through its Python API, and ODRoNeS parametrises the corresponding `paramPoly3` curves by translating and rotating the set of Bézier control points into the local space and rearranging the different factors of the polynomial.

When possible, a Blender curve can be simplified into a `straight` or an `arc`. Identifying a straight segment represented as a Bézier is straightforward: all the control points fall onto the same line. Finding out when to swap an `arc` is trickier since Bézier curves and arcs have a different mathematical nature. However, if the curve has been intentionally been drawn as an arc in Blender, it will be created as a series of short Bézier curves that result in a visually similar curve. This requires the start and end points of every Bézier $(x_1, \dots, x_n)$, to be on the same arc. Using this series of points, ODRoNeS creates a series of arcs defined with sets of three points (x_1, x_i, x_n) , where $i \in (2, n-1)$. If all the resulting arcs have a similar enough curvature, then ODRoNeS can approximate the arc with x_1, x_n and the initial tangent taken as the average of all the $(n-2)$ arcs considered.

Elevation

OpenDRIVE captures road elevation through four complementary methods. First, the `elevation` itself describes the height along the reference line as a series of `elevation` elements:

$$e(\Delta s) = a + b\Delta s + c\Delta s^2 + d\Delta s^3 \quad (4)$$

where $\Delta s = s - s_o$, s goes from 0 to the end of the road, and s_o is the distance down the reference line where this `elevation` element starts.

This polynomial form relates directly to the z coordinate of the 3D Bézier curves in Blender. Hence, every `elevation` element can be parametrised just by rearranging the parameters and later normalising them by the length of the curve on the xy -plane ($b = b'/l$, $c = c'/l^2$, $d = d'/l^3$, where l is the length of the road).

The other three methods in the standard provide alternative descriptions of the lateral profile of the road. First, `superelevation` describes the tilt angle along the travel direction s as a series cubic polynomials, just like `elevation`. Second, `shape` elements specify the road shape as a series of 3rd order polynomials in the t direction (perpendicular to the travel direction) to be interpolated linearly in the s direction, and finally `cross section surface` further enhances shapes by defining different strips and allowing cubic polynomials to define the change of their coefficients along s .

Blender provides the torsion angle at the end points of every curve, its documentation stating that the torsion along the curve itself can be interpolated linearly. This relates directly to the OpenDRIVE `superelevation` and thus ODRoNeS parametrises such elements, providing detail on the resulting maps. On the contrary, the other two methods describing the road shape in the t direction cannot be parametrised so directly and the ODRoNeS plugin for Blender does not support them.

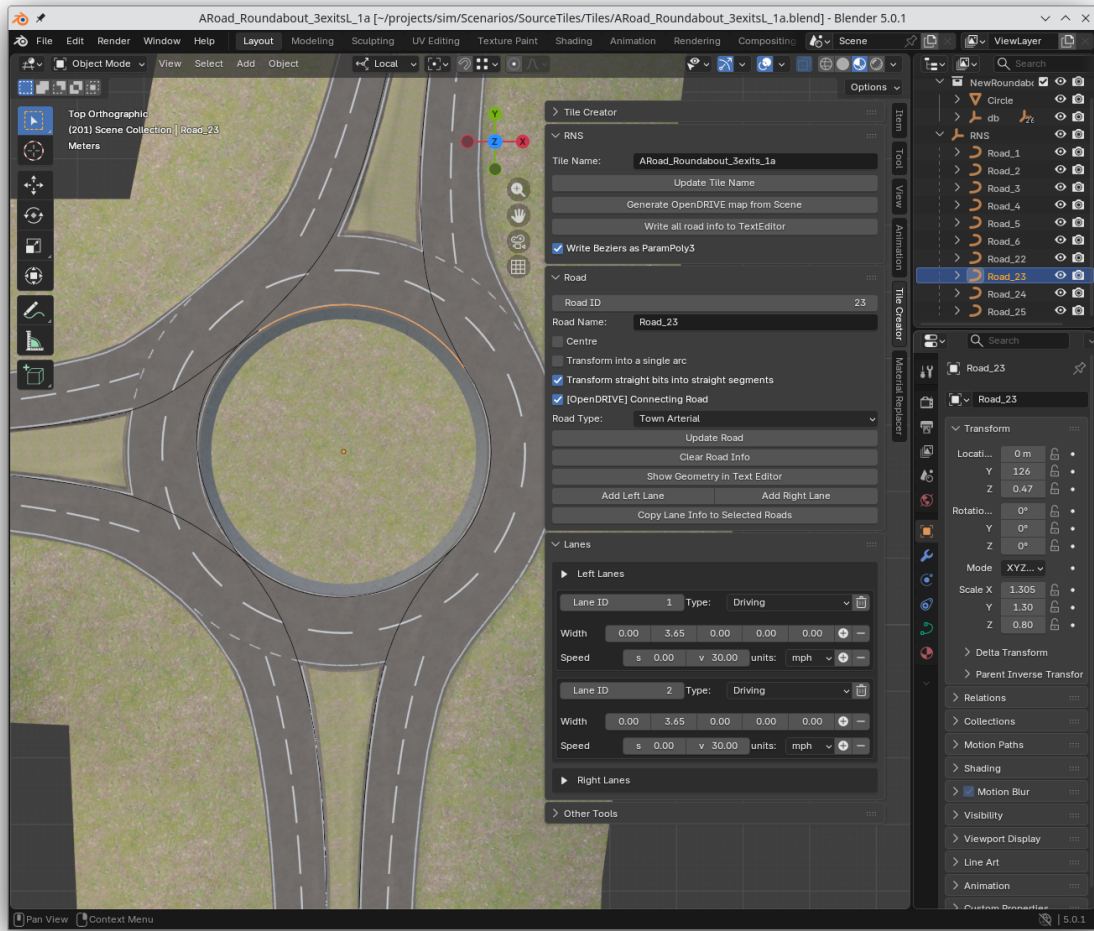


Figure 1: The ODRoNeS plugin for Blender interface in Blender showing a roundabout for which all reference lines are drawn in black but one that is selected and highlighted in orange. The Tile Creator area shows the details of the road, and two left lanes (on a left-side-driving map) defined with constant width and speed.

Roads and Lanes

In order to write an OpenDRIVE file, the ODRoNeS plugin for Blender also needs to know the details of the lanes on both sides of the reference curve. More precisely, the user needs to add left and right lanes, their width, speed limit (and its units) and type.

OpenDRIVE defines width through a series of `width` elements, each defining five parameters a , b , c , d and $sOffset$ so that the width at a certain distance down the road s is calculated as

$$w(\Delta s) = a + b\Delta s + c\Delta s^2 + d\Delta s^3 \quad (5)$$

in which $\Delta s = s - s_{offset}$, and s goes from zero to the end of the road. Similarly, the standard states that multiple speed elements can be defined, with different limits changing at specific s distances.

This is supported by ODRoNeS and its Blender plugin as shown in Fig.1, which shows a screenshot of the interface. The reference lanes are drawn in black, one of them highlighted in orange, for which the interface shows how two left lanes (this is a left-side-driving map) are defined with constant length and constant speed.

Linkage and Junctions

At both road and lane levels, the OpenDRIVE standard demands the maps to make explicit every link, that is, to always specify the next and previous lane and road. When the linkage presents any complexity, for example at forks or roundabouts, it requires a detailed description using a junction element.

Given a set of roads and lanes in Blender, ODRoNeS can do the linkage automatically taking no input from the user. The approach uses only geometric considerations and is described in the following paragraphs.

In a preliminary stage, ODRoNeS sets port and starboard links to every lane on every road. Later, taking the driving side as specified by the OpenDRIVE map, ODRoNeS sets the direction for every lane on two-way roads, again using geometric considerations. With that, ODRoNeS links lanes in a four-stage process.

First, it links lanes on two-way roads, that is, for every pair of two-way roads and for every pair of lanes l_i and l_j onto these roads, consider two lanes linked either if l_i finishes where l_j starts or l_i starts where l_j finishes. That works because ODRoNeS has al-

ready established the way of travel for lanes on two-way roads.

Second, it links lanes on one-way roads to lanes on two-way roads if any of their ends are close enough and their tangents at that point match. This disregards pairs of lanes touching at an angle, which should be unrelated. When lanes are linked, it also flips those on the one-way road if the link was origin-to-origin or destination-to-destination, thus correcting their way of travel.

Thirdly, ODRoNeS considers linking lanes on every pair of one-way roads using the same close-edges and tangents conditions. Finally, when all the one-way road pairs have been considered, ODRoNeS adjusts the direction of the rest of lanes on every one-way roads, by recursively checking whether they link to a two-way or to a one-way road with a known direction of travel.

While this is enough to provide predecessor and successor details to lanes and roads, and ODRoNeS does not need more to load an OpenDRIVE file, the standard requires a junction when there is more than one successor or predecessor. Currently, the ODRoNeS plugin for Blender solves this problem by requiring the user to mark each road belonging to a junction as a `connectingRoad`.

Discussion

We have been using maps made with the plugin previously described in a series of driving simulator experiments, both to create traffic along the roads and to characterise the driving performance of the participants. These maps were created out of 3D models in which the roads had dedicated meshes, and this allowed us to automatically create the reference line by converting each road mesh into a Bézier curve, thus resulting in high quality maps. Instead, if the road on the 3D model is not an independent object with a separated mesh the automatic road-to-curve conversion is not possible. That would happen, for example, on a model created with 3D scene reconstruction methods, as they create a single non-editable object. Previous NeRF-based methods had intrinsic difficulties with mesh editability and although new-generation Gaussian splatting methods may be able to solve this challenge, further research is needed (Chen and Wang, 2026). When roads are not independent 3D models, the user needs to manually draw the reference lines onto the road which could become a tedious process for large areas.

Visualising the resulting OpenDRIVE lanes in Blender is a useful feature that is currently missing. It is important both to ensure that they match the 3D models, and to check that they flow nicely. Because the lanes are extruded from the reference line, discontinuities will appear if the tangents at the end points of the reference lines do not match, even when they do touch. Visualising the lanes is quite easy to implement, and it is one of our next milestones to improve usability.

Another limitation regards marking “connecting roads” manually to create common junctions correctly. This is not a significant burden, but it requires understanding OpenDRIVE junctions. Future versions of the plugin will automate this step using geometric

considerations. Indeed, ODRoNeS can already detect overlapping roads which could be used to identify common and virtual junctions, and it could detect direct junctions based on the number of successor and predecessor roads.

Finally, automatic parameterisation of shapes and strips to take into account the lateral shape of the road is not straightforward. While road camber is automatically parameterised from the torsion of the mesh, automatic handling of the more sophisticated surface shape remains out of scope for now.

Conclusions

This paper introduces a new plugin for Blender capable of extracting a road network from a 3D scene when loaded in the popular computer graphics tool Blender and of saving it in the OpenDRIVE standard format. The paper exposes the mathematical details of how geometry, elevation and linkage are calculated and optimised from the data exposed by the Python interface to Blender. The plugin builds on the free software library ODRoNeS, a standalone C++ tool meant for simulation providing computational geometry both for roads and lanes. We expect that the work is useful to other small groups working in driving simulation and related fields and to the open source community.

References

- Bézier, P., 1986. *The mathematical basis of the UNIURF CAD system*. Butterworth-Heinemann.
- Blender 2026. <https://www.blender.org>. [Accessed Apr. 14, 2026].
- Chen, G. and Wang, W., 2026. A survey on 3d gaussian splatting. *ACM Computing Surveys*, 58(12), pp. 1–39.
- Gasbarro, L., Ragni, M., and Davide, V., 2024. AnteMotion’s solution for Automatic creation of simulation environments for ADAS and AV. In: *Product Solution Book of the Driving Simulation Conference 2024*. Driving Simulation Association. Strasbourg, France, pp. 35–38.
- Höhne, H., 2019. *OpenDRIVE Road Editor*. <https://github.com/fhwedel-hoe/OpenRoadEd>. [Accessed Apr. 14, 2026].
- Knabe, E., Guldogus, M., Lundell, S., and Han, Z., 2019. *Environment Simulator Minimalistic (esmini)*. <https://github.com/esmini/esmini>. [Accessed Apr. 14, 2026].
- Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wiessner, E., 2018. Microscopic Traffic Simulation using SUMO. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2575–2582. <https://doi.org/10.1109/ITSC.2018.8569938>.
- Maierhofer, S., Ballnath, Y., and Althoff, M., 2023. Map Verification and Repairing Using Formalized Map Specifications. In: *2023 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1277–1284.
- Schmitz, J., Liu, Y., and Wenninger, J., 2021. *Blender driving scenario creator add-on*. <https://github.com/johschmitz/blender-driving-scenario-creator>. [Accessed Apr. 14, 2026].
- Solernou, A., 2023. *ODRoNeS (OpenDRIVE Road Network System)*. <https://github.com/ITSSimulators/ODRoNeS>. [Accessed Apr. 23, 2025].
- Valente, M. and Milia, L., 2022. Roadrunner: Scenario Design for Driving Simulation. In: *Product Solution Book of the Driving Simulation Conference 2022*. Driving Simulation Association. Strasbourg, France, pp. 9–14.

A Low-Cost Driving Simulator Platform for Integrated Vehicle Comparison, HIL testing and Public Engagement

Stefan Haehnlein¹, Judith Frueh¹, Mirko Ronecker¹, Christian Kloeffer¹

(1) Offenburg University of Applied Sciences, Badstraße 24, 77652 Offenburg, Germany
stefan.haehnlein@hs-offenburg.de

Abstract – This paper presents a low-cost driving simulator platform designed for real-time vehicle comparison, optional powertrain test bench integration and public engagement. The system combines a commercial off-the-shelf (COTS) hardware setup (approx. € 4,000) with a simulation environment, enabling a unified platform that bridges vehicle simulation, hardware-in-the-loop (HIL) testing and science communication. The platform allows parallel evaluation of battery-electric vehicles (BEV) and internal-combustion-engine vehicles (ICE) under identical user-driven conditions. Simulated driving scenarios can be optionally coupled with a physical electric powertrain testbench via real-time UDP communication. Gamification elements including a user interface with leaderboard and direct comparison between BEV and ICE supports intuitive understanding of energy consumption, efficiency and operating costs. The system has been deployed at six public events, reaching approx. 200 participants. Short interaction cycles and immediate feedback encourage repeated use and initialize discussion between users and scientific staff. While the current evaluation is based on qualitative observations and basic metrics, the results demonstrate the feasibility of combining COTS, vehicle simulation, HIL testing and science communication. Future work will focus on structured user studies and quantitative validation.

Keywords: Driving Simulator, Hardware-in-the-Loop (HIL), Electric Powertrain, Low-Cost Simulation, Science Communication

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

Driving simulators have become valuable tools in both research and public outreach, as they provide safe, repeatable, and intuitive representations of vehicle dynamics. Recent work in electrified powertrain testing has shown that real-time co-simulation enables synchronised coupling of test benches, allowing realistic evaluation of electric powertrains under dynamically changing loads (Rautenberg et al., 2023). Likewise, emulation of on-road operating conditions on physical test benches enables reproduction of both typical and unexpected drive-cycle behaviours that are difficult to generate consistently in field tests (Fajri, et al., 2016). Parallel research on cost-efficient driving simulators has demonstrated that systems based on commercial COTS components can achieve sufficient fidelity for behavioural studies and educational use (Weinberg and Harsham, 2009). Within science communication, interactive and mobile installations are recognised as effective formats for engaging

diverse and underserved audiences and supporting informal learning processes (Humm and Schrögel, 2020). Building on these strands of research, this work presents a low-cost driving simulator platform that integrates comparative vehicle modelling and optional hardware coupling. The platform is intended to support technical demonstrations and to explore its potential as a tool for science communication and was developed as part of the move.mORE project. The move.mORE project funded by the Innovation University (IHS) promotes research-based transfer of ideas, knowledge and technology in the field of sustainable mobility and is jointly run by the universities of Karlsruhe and Offenburg.

System Architecture

Hardware Setup

In this work, low-cost is defined as a total hardware cost at approx. € 4,000. The proposed simulator is built as a mobile driving simulator without motion feedback, except steering wheel force feedback

(FF), using COTS components. Compared to high end research simulators with multi degree of freedom motion systems and costs above € 150,000, as well as mid-range commercial simulators (€ 10,000 - € 60,000) (De Frutos and Castro, 2021; Skogstad, M. R., & Sætren, G. B., 2021), the presented platform focuses on a cost efficient and mobile setup for flexible experimentation and public engagement. The system consists of a single sim-racing cockpit, a FF direct-drive steering wheel, pedals, a multi-monitor visualization setup and a desktop PC. The setup is described in detail in Table 1. The prices are updated in April 2026 with equivalent hardware components.

Table 1: Hardware Setup

Component	Description	Price (Excl. VAT)
Cockpit	Next Level Racing F-GT Elite Wheel Plate Edition with ERS3 Seat, Monitor Stand, Keyboard and Mouse Tray and Wheels	€ 1,425
Desktop PC	mATX Case, AMD Ryzen 7 7700, B850M Mainboard, Nvidia RTX 5060 Ti, 32 GB DDR5 RAM, 1 TB SSD, 750 W Power Supply, Win 11	€ 1,500
Sound System	Samsung HW-S810GD	€ 250
Monitors	3x 32" WQHD 100Hz Monitor (LG 32U631)	€ 435
Wheelbase and Pedals	Fanatec CSL DD QR2 Ready2Race Bundle (5 Nm)	€ 320
Accessories	Mouse, keyboard, cables, mounting hardware	€ 50
Optional TV for Public	55" TV with mobile stand (LG 55UR781C)	(€ 350)
Driving simulator without optional TV		€ 3,980

The overall setup including the optional monitor for public engagement can be seen in Figure 1.



Figure 1: COTS driving simulator with optional monitor

Software Setup

The software architecture of the proposed platform follows a modular co-simulation approach combining a real-time vehicle simulation environment with an open interface. The primary simulation environment is BeamNG.tech, which models vehicle dynamics using a soft-body (node-beam) approach. In contrast to conventional vehicle modelling approaches based on rigid body dynamics combined with explicit component models (e.g., tire models such as the Pacejka Magic Formula), forces and dynamic behavior in BeamNG.tech emerge from structural interactions, contact modeling and parameterized material properties. Consequently, effects such as tire compliance and mechanical friction are not explicitly parameterized but arise implicitly from system dynamics (BeamNG GmbH, 2025). Telemetry data from a BEV, including vehicle speed, motor rotational speed, motor torque, battery capacity and State-of-Charge (SOC) are transferred in real-time into a Matlab/Simulink environment. A minimum step size of 1 ms was achieved in the conducted tests (limited by BeamNg.tech interface). While this is sufficient for applications such as energy flow and power analyses, it is not suitable for studies requiring PWM related real-time behavior. The Matlab/Simulink co-simulation layer serves as the central processing unit of the platform. Two main functional paths are implemented. First, telemetry data can be transmitted via UDP to an external electric powertrain test bench, enabling real-time HIL operation. Second, the same data is used to compute a parallel comparison model of an ICE based on simplified energy flow and efficiency calculations. The architecture is designed to be modular and extensible.

While the current implementation uses BeamNG.tech with Matlab/Simulink, equivalent functionality can also be achieved using Python-based implementations. BeamNG provides an open Python API, enabling the integration of custom vehicle models, control algorithms, comparison models and UDP sockets implemented directly in Python. Similarly, BeamNG.tech can be replaced by other simulation environments or even low-cost games, provided that real-time telemetry access and sufficient co-simulation performance are available. For example, the driving simulator and the entire architecture have already been used in combination with the racing game Project Cars 3 where the telemetric data can be exchanged via a local host UDP connection. However, the lower costs come at the expense of deriving engine torque and fuel consumption from limited telemetry data (only speed over time, a vehicle model have to be implemented) rather than obtaining them directly from the simulator. Overall, the software architecture represents a trade off between physical modeling fidelity, real-time performance, and system accessibility.

The overall architecture is shown schematically in Figure 2.

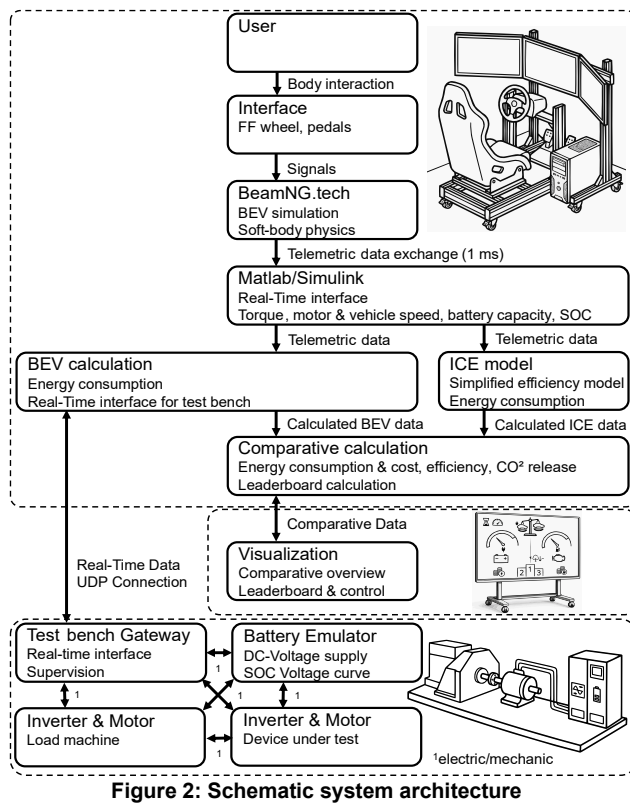


Figure 2: Schematic system architecture

Application Scenarios

Powertrain Test Bench Coupling

In the first scenario carried out, telemetry data from the simulated BEV is transmitted via UDP to an electric powertrain test bench for HIL operation. An implemented synchronization mechanism based on a continuously incrementing counter ensures real-time consistency and allows detection of communication errors for safety reasons. Also higher level protocols like the distributed co-simulation protocol are well suited for this purpose (Krammer et al., 2018). The simulated motor speed is mapped to the speed reference for the load machine, while

torque demands are translated into inverter current references based on Maximum Torque Per Ampere and field weakening strategies. Additionally, a battery emulator reproduces the DC-link voltage based on the simulated SOC, enabling realistic operating conditions. Thermal conditions can be varied using an external conditioning system. This setup allows real-time evaluation of control algorithms for electric powertrains under user driven, dynamically changing load conditions.

Real-Time Vehicle Comparison

In the second scenario, the telemetry data are used to compute a parallel comparison model of an ICE. Both vehicles follow identical user driven trajectories. Thereby the energy consumption of the BEV is derived directly from simulation data. The ICE model is intentionally simplified and based on energy flow calculations with representative drivetrain efficiency derived from literature (Ehsani et al., 2018). It is designed for comparative visualizations rather than high fidelity drivetrain analysis. Both results are visualized in a graphical user interface given in Figure 3, showing energy consumption, efficiency, CO₂ pollution and estimated energy costs. The leaderboard scoring system seen in Figure 2 on the right-hand side, combining travel time and energy consumption is integrated. This gamification element encourages repeated usage and exploration of different driving styles. This scenario demonstrates that intuitive real-time comparison between BEV and ICE can be achieved within a COTS simulation environment.

Public Deployment and Evaluation

The simulator platform was deployed at seven public events between 2024 and 2026 reaching approximately 250 participants. Basic interaction metrics were recorded in 2024 and 2025 (number of participants, approx. age, academic background and number of runs). The predefined test case the users

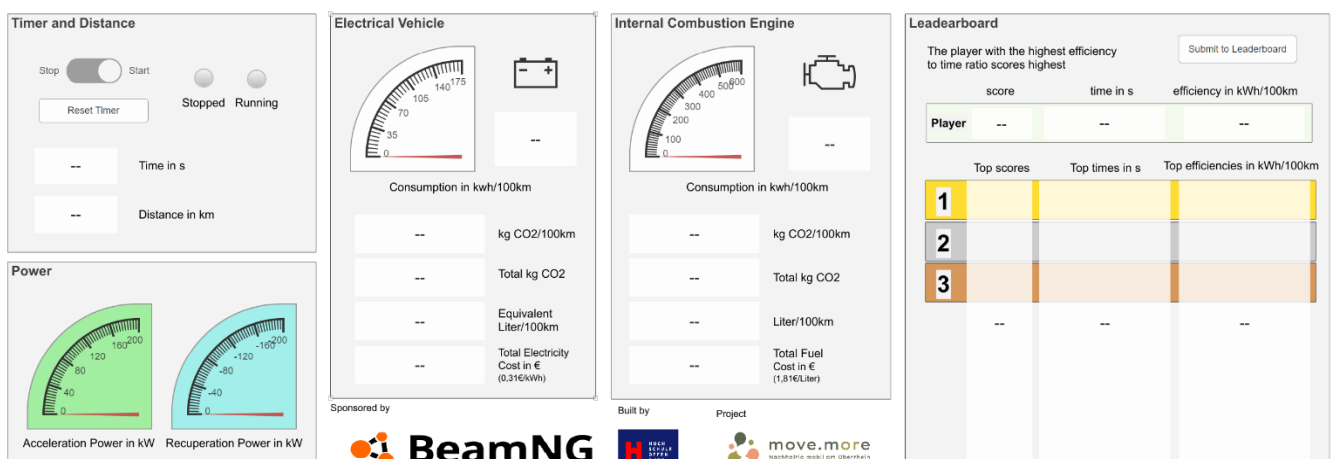


Figure 3: Graphical user interface with control elements showing real-time comparison metrics and leaderboard

should drive was designed with short interaction cycles of approx. 1 – 2 minutes per simulation run. This enabled a high throughput of users while encouraging repeated interaction. Observations across all deployments indicate that most users completed multiple runs, typically at least twice, in order to directly explore the influence of their behavior on energy consumption and efficiency metrics. This immediate feedback loop was identified as a key factor in user engagement. Following repeated interactions, users frequently remained at the simulator platform, discussing their results with the supervising scientific staff. These discussions often focused on driving behavior, energy consumption, differences between BEV and ICE and the implemented calculations.

Since 2026, deployments have been accompanied by questionnaires. Preliminary evaluations indicate that simulator use alone does not necessarily increase knowledge of electromobility, but it reliably attracts diverse audiences, initiates discussion, and facilitates contact with non academic groups that are otherwise difficult to reach. At one event, test bench coupling added visual and acoustic feedback from speed, load and inverter metrics, but no clear qualitative difference in engagement or discussions was observed. While no formal quantitative study was conducted for the earlier deployments, these observations provide initial insights into the systems potential for interactive science communication. Figure 4 shows the deployment of the driving simulator at a public event on the marketplace in Offenburg. Due to its compact size (3 m x 3 m minimum floor size) and high mobility, the simulator can be easily set up in outdoor environments.



Figure 4: Deployment during public event

Conclusion

This work presented a mobile low-cost driving simulator platform that integrates real-time vehicle comparison, HIL capabilities and public engagement within a unified system architecture. The results demonstrate that a low-cost simulator can provide sufficient real-time comparison between BEV and ICE under identical driving conditions. The integration with a physical test bench further

confirms the suitability for exploratory testing of electric drivetrain control strategies and also proved to be effective as a tool for science communication. Short interaction cycles, immediate feedback and comparative visualization enabled users to directly relate their driving behavior to energy consumption and efficiency. Repeated usage and subsequent discussion with supervising staff indicate that the system can successfully facilitate public engagement. The qualitative observations and basic interaction metrics and the first qualitative evaluation provide initial evidence for the potential of the platform to bridge the gap between technical research and public engagement. Future work will focus on further detailed quantitative evaluation of user interaction and learning outcome.

References

- De Frutos, S.H. and Castro, M., 2021. Assessing sim racing software for low-cost driving simulator to road geometric research. *Transportation Research Procedia*, 58, pp.575–582. <https://doi.org/10.1016/j.trpro.2021.11.076>.
- Ehsani, M., Ebrahimi, K., Longo, S. and Gao, Y., 2018. *Modern electric, hybrid electric, and fuel cell vehicles*. Third edition edn. Boca Raton, FL: CRC Press.
- Fajri, P., Prabhala, V.A.K. and Ferdowsi, M., 2016. Emulating On-Road Operating Conditions for Electric-Drive Propulsion Systems. *IEEE Transactions on Energy Conversion*, 31(1), pp.1–11. <https://doi.org/10.1109/TEC.2015.2481180>.
- Humm, C. and Schrögel, P., 2020. Science for All? Practical Recommendations on Reaching Underserved Audiences. *Frontiers in Communication*, 5, p.42. <https://doi.org/10.3389/fcomm.2020.00042>.
- Krammer, M., Benedikt, M., Blochwitz, T., Alekeish, K., Amringer, N., Kater, C., Materne, S., Ruvalcaba, R., Schuch, K., Zehetner, J., N., 2018. The Distributed Co-simulation Protocol for the integration of real-time systems and simulation environments. In: *Proceedings of the 50th Computer Simulation Conference*, SummerSim '18. San Diego, CA, USA: Society for Computer Simulation International.
- Rautenberg, P., Weber, P., Degel, J.P., Hähnlein, S., Gauterin, F., Koch, T., Doppelbauer, M. and Gohl, M., 2023. Electrified Powertrain Development: Distributed Co-Simulation Protocol Extension for Coupled Test Bench Operations. *Applied Sciences*, 13(4), p.2657. <https://doi.org/10.3390/app13042657>.
- Skogstad, M. R., & Sætren, G. B., 2021. Driving Factors for Driving Simulators – A Feasibility Study. [online] European Safety and Reliability Conference (ESREL) <<https://www.rpsonline.com.sg/proceedings/9789811820168/pdf/190.pdf>>.
- Weinberg, G. and Harsham, B., 2009. Developing a low-cost driving simulator for the evaluation of in-vehicle technologies. In: *Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. [online] AutomotiveUI '09: International Conference on Automotive User Interfaces and Interactive Vehicular Applications. Essen Germany: ACM. pp.51–54. <https://doi.org/10.1145/1620509.1620519>.

AI & MACHINE LEARNING FOR SCENARIO GENERATION

An AI-Based Validation Intelligence Layer for Regulation-Ready Autonomous Driving

Yves Peirsman¹, Stephen Lernout²

(1) Deontic BV, Leuven, Belgium; yves@deontic.ai (Co-Founder & CTO)
(2) Deontic BV, Leuven, Belgium, stephen@deontic.ai (Co-Founder & CEO)

Abstract – Autonomous driving validation must cover a large and continuously evolving space of traffic scenarios while preserving traceability from requirements and operational design domain assumptions to executable tests and audit-ready evidence. This paper presents Deontic, an AI-based validation intelligence layer that converts natural-language requirements, ODD constraints, and safety objectives into standards-based OpenSCENARIO and OpenDRIVE artifacts. The solution integrates with existing requirements, coding, simulation, and CI/CD environments through an API layer and MCP server. It supports base-scenario generation, parameter expansion, reusable scenario libraries, requirements linkage, and continuous homologation workflows that monitor change, assess impact, and prioritize regression validation. Internally reported benchmarks indicate a reduction from approximately 340 minutes of manual engineering work to a few minutes for a single base scenario, with corresponding cost reductions. The results suggest that domain-specific AI can scale scenario coverage and compliance operations without replacing engineering rigor.

Keywords: autonomous driving validation; scenario engineering; operational design domain; continuous homologation; generative AI.

Copyright: © 2026 by the authors.
Licensee Driving Simulation Association.

Introduction

Autonomous driving development depends not only on model performance but also on the ability to validate behavior across a large and evolving space of traffic scenarios. As regulatory requirements become more formalized, validation increasingly requires traceability, auditability, and repeatability. Validation and homologation are major bottlenecks because rules change, internal specifications drift, operational design domains expand, and the number of scenarios requiring generation and maintenance grows rapidly. In such an environment, manual methods struggle to scale.

The motivation for this work is tied to the changing regulatory landscape. The emerging UNECE automated driving system framework makes auditable, scenario-based validation central to approval. Autonomous driving companies, automotive OEMs, and Tier 1 suppliers therefore need to convert requirements, ODD assumptions, and safety logic into regulator-ready evidence. We propose Deontic as an answer to that need: a system that automates traceable scenario generation while fitting into existing engineering, simulation, and homologation workflows.

This work builds on several prior technical directions. First, the solution aligns with standards-based scenario modelling through ASAM OpenSCENARIO and OpenDRIVE. Second, it reflects the continuing importance of scenario-based validation in the era of end-to-end autonomy, where internal model behavior may be difficult to interpret directly. Third, it uses agent-oriented tooling, including the Model Context Protocol, to connect AI agents with requirements, simulation, and compliance systems. Deontic can therefore be understood as a convergence of simulation-based validation, requirements engineering, compliance monitoring, and generative AI for engineering workflows.

Product Solution

Deontic is not an autonomous driving model, but an enabling validation layer around autonomous driving development. For a non-specialized audience, the product can be described as an AI software agent that helps engineers generate and maintain the test scenarios needed to demonstrate that a vehicle behaves safely and in accordance with regulatory expectations. Instead of manually

writing every scenario, engineers can start from written requirements, ODD definitions, and constraints, which Deontic translates into structured, standards-based scenario artifacts. Core capabilities include natural-language conversion into OpenSCENARIO and OpenDRIVE, parametric expansion across variables such as weather, actors, and environment, and the creation of reusable scenario libraries with simulator integration. The platform is therefore useful both for generating individual scenarios and for creating broader coverage across a validation space.

A central proposition is that Deontic acts as the validation intelligence layer for end-to-end autonomy. While modern autonomy stacks may rely on vision-language models or end-to-end models to understand scenes and produce driving behavior, Deontic addresses a different but equally important challenge: how to prove, in a traceable and auditable way, that such systems are safe across relevant scenarios. It turns requirements and ODD assumptions into auditable scenario evidence and preserves regulatory auditability even when the internal workings of the driving model are opaque.

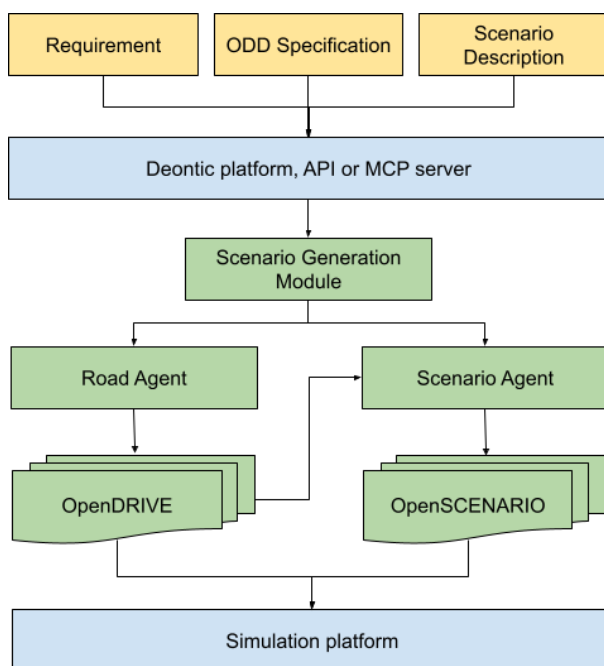


Figure 1: High-level architecture of Deontic's AI-driven scenario pipeline

Methodology

The methodology is based on integration into the existing working environment of automated driving

engineers. Rather than replacing engineering tool chains, Deontic is deployed where AD engineers already work, within coding environments and testing pipelines tied to requirements systems, simulation platforms, and CI/CD processes. This embedded approach supports industrial use rather than isolated experimentation.

From a software perspective, Deontic interacts with engineering environments such as Visual Studio Code and agent interfaces like Claude Cowork, while also exposing functionality through an API layer. The system connects requirements integration, simulation integration, and scenario-based testing workflows. Its output includes scenario generation, ODD coverage, requirements linkage, and homologation oversight, with compatibility across simulation and physical-AI model environments used in automotive engineering.

The architecture is further implemented through a Model Context Protocol design. An orchestrating agent plans tasks, calls tools, and retains context. An MCP server provides stable contracts for tool access. Deontic-specific tooling then interprets the submitted requirements, scenario descriptions and ODD constraints, and generates the requested scenarios in OpenDRIVE and OpenSCENARIO. It can also interface with requirements systems or simulation platforms where necessary. This modular design supports extensibility and reduces the need to rewrite prompts or manually bridge systems when a workflow changes.

In the main use case, a user begins with a requirement, an ODD slice, or a safety objective. Deontic interprets the requirement, derives scenario intent, creates a standards-compliant scenario, and expands it across relevant parameters for testing coverage. A second use case is continuous homologation, organized as a three-step loop: monitor regulatory, legislative, and internal changes; analyze their impact on features, target ODDs, and scenario sets; and close the gap by regenerating scenarios and prioritizing regression validation. This methodology addresses both initial validation and the longer lifecycle of maintaining compliance as systems and rules evolve.

Example Use Case

For example, suppose an engineer needs to verify and validate the behavior of an ADAS function like adaptive cruise control. With Deontic, he or she can submit a natural-language description of the requirement, together with a definition of the target operational design domain, which could state, for example, that the function should work on highways, at speeds below 130 kilometers per hour, in clear weather conditions. Deontic then uses this input to generate a range of base scenarios, with varying scenario participants (such as lead cars, motorbikes

and trucks), road networks (e.g. straight versus curving roads, with different numbers of lanes), traffic conditions (e.g. high versus low traffic), etc. For each scenario, it creates OpenDRIVE and OpenSCENARIO files, which can be imported into a wide range of simulation software, such as IPG's CarMaker, to run the full simulation. Moreover, each base scenario can be expanded to cover a wide variation in parameters, including the vehicle settings (e.g. the speeds of the scenario participants and the distance between them), the weather conditions (e.g. the position of the sun), etc. In the Deontic platform, all generated artifacts can be traced back easily to the ODD specification and requirements they were created from. In this way, engineers can achieve maximal ODD coverage ready for simulation in minimal time.

Results and Discussion

Internal measurements present concrete performance claims for the scenario-engineering workflow. For a single base scenario, the human-engineered V-model process is estimated at 340 minutes, or about 5.5 to 6 hours. The breakdown includes requirements and intent interpretation, scenario and interaction design, detailed parameterization, implementation in OpenDRIVE and OpenSCENARIO, verification and debugging, and traceability and documentation. This decomposition shows that the challenge extends beyond writing a scenario file to the full engineering lifecycle surrounding scenario creation.

By contrast, the Deontic AI agent is reported to complete the corresponding work in one to five minutes for a single base scenario. The internal cost comparison is about USD 580 for the manual engineering process versus about USD 30 for the Deontic-based process. Although these values require validation across broader customer deployments, the magnitude of the reported improvement supports the proposition that AI assistance can materially change the economics of validation.

Deontic's internal benchmarks also show that its custom AI agent for ASAM OpenSCENARIO and OpenDRIVE outperforms general-purpose foundation models on scenario-logic tasks. This suggests that the technical advantage does not come merely from using a large language model, but from combining domain-specific customization with standards awareness and engineering workflow integration. For regulated technical outputs such as simulation scenarios, industrial deployment requires consistency, structure, and compatibility with downstream toolchains.

The reported results imply several industrial benefits. First, reduced scenario-generation time enables broader coverage without proportionally increasing

engineering headcount. Second, tight linkage between requirements and generated scenarios improves traceability for audits and homologation review. Third, the continuous-homologation roadmap addresses the practical challenge of maintaining validation assets as regulations, product features, and assumptions change over time. Together, these features position Deontic not merely as a productivity tool, but as infrastructure for scalable compliance and validation operations.

Conclusions and Implications

We position Deontic as a response to a growing industrial need. As autonomous driving systems move toward deployment under stricter regulatory frameworks, validation and homologation become central barriers to scale. The traditional manual process for scenario engineering is slow, costly, and difficult to maintain under continuous change. Deontic addresses this problem through an AI-based validation intelligence layer that transforms requirements, ODD assumptions, and compliance constraints into auditable, simulator-ready scenarios.

The expected impact for users is substantial. Engineers gain faster scenario generation, reduced manual effort, and improved traceability between requirements and validation evidence. Validation and homologation teams gain stronger support for audit readiness and change management. OEMs, Tier 1 suppliers, and autonomous driving developers may benefit from lower validation costs, shorter turnaround times, and improved scalability as their scenario libraries and regulatory obligations grow.

For the broader industrial community, the value of this work lies in repositioning validation as a software-driven, continuously updated capability rather than a labor-intensive bottleneck. If the reported gains in time and cost are sustained in practice, platforms such as Deontic could become essential infrastructure for regulation-ready autonomy. More broadly, this work suggests a model for applying generative AI in safety-critical domains: not as a replacement for engineering rigor, but to scale standards-based, traceable, and auditable engineering processes.

POSTERS

Virtual Reality Driving Simulator for Education - Form Factor, Cybersickness and Perception

Helen Loeb¹, Ayagoz Smagulova², Matan Rosenberg², Eduardo de la Pena³, Oskars Sniegs⁴

(1) Jitsik, 276 Barwynne Rd Wynnewood PA 19096 USA: helen.loeb@jitsik.com

(2) Drexel University, 3141 Chestnut St, Philadelphia, PA 19104: {as5984, mlr432}@drexel.edu

(3) MetoDrive, Vasku iela 26, Rīga, Latvia: eduardo@metodrive.com

(4) VRCentrs, Vasku iela 26, Rīga, Latvia: oskars@VRCentrs.

Copyright: © 2026 by the authors.

Licensee Driving Simulation Association.

Introduction

Across the world, various countries have added driving simulator training and assessment to the licensing process. Singapore, India, France, the UK, Japan allow simulator training in some form. In the US, the *Ohio - Ready Set Drive* program, introduced in 2019, exposed all prospective new drivers to simulated driving scenarios prior to their in-car driving test (Shapiro, 2023).

Yet, despite these advances in deployment, driving simulators are not used to the best of their capacity. We explore with this work the pros and cons of Virtual and Mixed Reality which hold the potential of making driving simulators more immersive at a low cost.

Research Question

Our research questions are the perceived educative value of the simulator as expressed by the user, and the level of cybersickness experienced.

Methodology

For our experiments, two simulators were built: the MetoDrive Pro and MetoDrive Real.

MetoDrive Pro

The MetoDrive Pro is built around a Treq frame which offers robust support for all electronics, Fanatec pedals and steering wheel, and a real car seat. A Meta Quest 3 Virtual Reality Headset was chosen for display. Also a set of 3 monitors can be used instead of the VR headset.

MetoDrive Real

With the goal of maximizing immersion and replicating the feel of a car, an alternative setup was developed where a real vehicle was used for an in-car driving simulator. A Volkswagen Golf Mk4 was retrofitted into a Virtual Reality Driving Simulator. The steering column and pedals were physically removed from the actual car and replaced with the Fanatec racing system. This simulator is attractive as it brings the feel and look of a real vehicle, so crucial for novice users.



Figure 1: Standalone MetoDrive Pro



Figure 2: In-Car MetoDrive Real

Driving scenarios

The MetoDrive Pro and MetoDrive Real simulators ran the same software suite, developed with Unity 3D and Visual Studio C#. A curriculum of 14 driving levels feature challenging scenarios such as:

- Pedestrian crossing on "Zebra"
- Truck obscuring pedestrian crossing
- Old and slow person crossing the street
- Lane change with traffic
- Icy road
- Hidden car exiting a garage
- Left lane merge on one way road
- Left turn with pedestrian crossing
- Children running on both sides of the street

Human Research

In May 2025, a Romanian energy company (name undisclosed) set out to raise awareness of the dangers

of the road by exposing its employees to challenging scenarios on driving simulators. Our research team used this opportunity for a human research study. Data was collected from a pool of 418 employees who drive routinely for work. Participants experienced challenging scenarios on either the Metodrive Pro or the Metodrive Real. After the experiment, they were invited to rate their experience through a choice of 5 options: 1) Awesome, 2) Great, 3) Ok, 4) Bad, 5) Horrible. A total of 241 participants drove the MetoDrive Pro vs a total of 177 for Metodrive Real.

Results and Discussion

Raw data for the experiment is presented in Table 1. Out of the pool of 418 participants, 368 participants completed the experiment (88%) and 50 did not feel comfortable and withdrew from the study.

Table 1: Participants ratings

Rating	Standalone MetoDrive Pro	In-Car MetoDrive Real
Awesome	101	87
Great	59	38
Ok	27	18
Bad	2	0
Horrible	1	1
Missing data	21	13
Unfinished	30	20

Simulation sickness

The study of these participants, who typically have no experience with Virtual Reality, permits a good insight into the simulator sickness effect. Out of the 384 participants who completed the survey, 285 (74%) rated the experience as Awesome or Great while 54 participants did not complete the ride or rated the experience as bad or horrible, a rate of 14%.

Simulator Form Factor

The In Car MetoDrive Real provided an experience somewhat different from the Standalone MetoDrive Pro. The native car environment is instantly recognized by participants. Still, the limitation to a Virtual Reality display (no monitors back up), could have made participants uncomfortable.

Results showed that favorable impressions (Awesome or Great) for the Standalone simulator amount to 72% vs 76% for the In-Car Simulator. Unfavorable

impressions (Bad or Horrible) amount to 15% for the MetoDrive Pro vs 13% for MetoDrive Real.

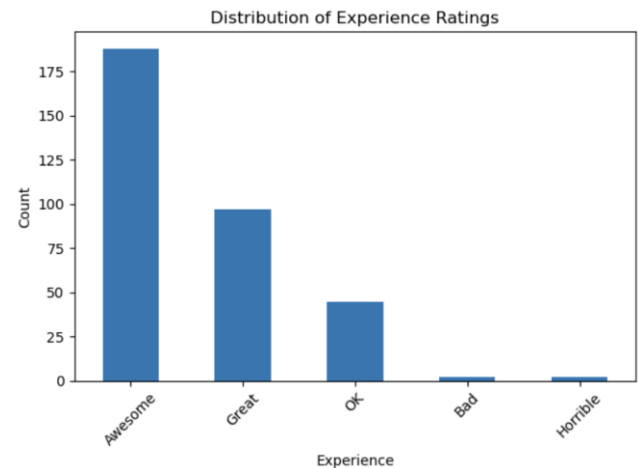


Figure 3: Distribution of Session Outcomes

When comparing the simulator form factor, we did not find any statistical difference for either cybersickness or overall impression. This result surprised us because the Standalone MetoDrive Pro and In-Car Metodrive Real have distinctive features. While we anticipated a strong interest by participants for the in-car simulator, we anticipated the VR only feature would translate into a high cybersickness rate and withdrawal from the study. This did not happen.

Conclusion

Driving simulation has come a long way since its early days. Virtual Reality immersion offers true potential to make it a valuable tool for driving education. Our Human study of 418 participants resulted in a 12% withdrawal rate for our population of professional adults. The majority of participants (72%) had a favorable impression of the simulation. Still, much needs to be done to make the tool ubiquitous for driving students around the world.

References

- Loeb, H., Leibowitz, C., Tehrani, R., & Hernandez, J., 2024. Driving Simulation for Driving Education: Can Mixed Reality Do It? *2024 Road Safety and Simulation Conference, University of Kentucky*.
- Mangharam, R., & Loeb, H., 2025. Driver Training for Shared Autonomy Systems Using Mixed Reality.
- Shapiro, N. 2023, Virtual driving test can predict risks of teen car accidents, *Forbes*.
- Walsh, E. A., Elliott, M. R., Cheng, S., Romer, D., Curry, A. E., Grethlein, D., & Winston, F. K. 2023. Driving skills at licensure and time to first crash. *Pediatrics*, 152(5), e2022060817.
- Wu, Z., Zhang, L., Hernandez, J., Leibowitz, C., Loeb, H., Dong, X., & Mangharam, R. 2024. Pipeline for Fast Digital Twin Development and Integration in Driving Simulation. *2024 Road Safety and Simulation Conference, University of Kentucky*

Transforming Simulation with Reality-Based Neural Rendering & World Models

Tolga Kart¹, Piotr Sokolski²

(1) Third Dimension AI, tolga.kart@thirddimension.ai

(2) Third Dimension AI, piotr.sokolski@thirddimension.ai

Abstract – Third Dimension AI is building the spatial generation layer for autonomy, robotics, and next-generation interactive content. Its flagship product, SuperSim, addresses the core bottlenecks of modern simulation: the sim-to-real gap, slow and costly environment creation, limited edge-case coverage, and fragmented toolchains, by starting from real sensor data rather than handcrafted assets. Leveraging state-of-the-art neural rendering 3D techniques, SuperSim transforms captured logs into dynamic, photorealistic simulation environments in hours, while AI-based world modeling enables scalable scenario generation with no artists required.

Keywords: Autonomous Vehicle Validation, Neural Simulation, Sim-to-Real Gap, World Models

Copyright: © 2026 by the authors.

Licensee Driving Simulation Association.

Introduction

The Simulation Challenge for Autonomous Systems

Simulation sits at the heart of modern autonomous vehicle (AV) development. It provides a controlled, repeatable environment for training perception and end-to-end models, validating planning and control software, and stress-testing behavior across the full distribution of real-world conditions. Yet despite its centrality to the development pipeline, simulation remains one of the industry's most persistent bottlenecks, constrained by four interconnected problems.

Sim-to-Real Gap

Traditional simulators rely on handcrafted or procedurally generated synthetic assets. The resulting environments are visually plausible but statistically diverge from real sensor data in ways that matter deeply for machine learning: lighting response, material reflectance, sensor noise characteristics, and the spatial statistics of objects and occlusions all differ from reality. Models trained or validated in these environments frequently fail to transfer to real-world deployments, a phenomenon known as the sim-to-real or domain gap, requiring expensive bridging techniques that partially negate the cost savings simulation is intended to provide.

Slow & Costly Simulation Creation

Building high-fidelity simulation environments manually is expensive, time-consuming, and difficult to scale. Creating a single production-quality scene typically requires weeks of effort from artists, simulation engineers, and domain experts. Maintaining those environments as sensor hardware evolves or as new operating domains are introduced multiplies the cost further. For smaller teams or programs under schedule pressure, this overhead is frequently prohibitive.

Limited Edge-Case Coverage

Safety standards and regulatory frameworks increasingly require AV developers to demonstrate robust performance not only in common conditions but across a vast long tail of rare, dangerous, and adversarial scenarios. These edge cases, unexpected obstacle behaviors, unusual weather events, sensor degradation modes, are by definition underrepresented in real-world data collection, and constructing them from scratch with artist-authored assets is both labor-intensive and geometrically limited in scope.

Fragmented Toolchains

Automotive development teams today navigate a patchwork of disconnected systems for data capture, 3D reconstruction, simulation rendering, and

validation reporting. Each handoff between tools introduces latency, data loss, and integration overhead. The absence of a unified pipeline from captured reality to validated simulation result forces teams to invest significant engineering effort in toolchain maintenance rather than core vehicle development work.

Product Solution: SuperSim

SuperSim is Third Dimension AI's neural simulation platform. It addresses all four of the challenges described above through a unified pipeline that begins with real sensor data and delivers simulation-ready, photorealistic environments in hours, without requiring artist input at any stage.

3D Computer Vision As The Foundation

SuperSim ingests raw sensor data and processes it through a state-of-the-art 3D reconstruction pipeline. At its core, SuperSim employs neural rendering techniques, a cutting-edge approach that represents scenes as collections of learnable 3D Gaussian primitives, enabling real-time, view-consistent rendering at photorealistic quality.

Unlike mesh-based representations, Gaussian Splatting faithfully captures continuous, view-dependent appearance effects, such as specular reflections, translucency, fine surface detail, and complex lighting interactions, which are critical for sensor realism. Because the environments are derived directly from real sensor captures, they inherit the photometric and geometric properties of the original scene, fundamentally reducing the domain gap at its source.



AI-Based World Modeling

Reconstruction alone produces a high-fidelity replica of a captured scene. SuperSim goes further by

layering AI-based world modeling on top of the reconstruction layer, enabling dynamic, generative extension of each captured environment. Specifically, SuperSim supports:

- 3D object insertion: placing new agents, obstacles, and infrastructure elements into the reconstructed scene with physically consistent appearance and occlusion
- Novel viewpoint synthesis: rendering the scene from sensor positions that were never physically captured, expanding the effective coverage of a single data collection run
- Environmental variation: modifying lighting conditions, weather, time of day, and surface properties to generate diverse operating conditions from a single baseline capture
- Agent behavior scripting: defining and parametrically varying the trajectories and actions of dynamic objects within the scene

Together, these capabilities transform SuperSim from a playback tool into a generative simulation platform: a single captured drive becomes the seed for thousands of distinct, simulation-ready scenarios.

Methodology

Step 1: Data Sharing & Initial Reconstruction

Customers share their sensor logs and captured data with Third Dimension AI through a secure transfer process. Third Dimension AI's engineering team processes this data through the SuperSim reconstruction pipeline and produces initial reconstructions for customer review. This review phase is collaborative: customers assess visual fidelity, geometric accuracy, and sensor-match characteristics, and provide feedback that is incorporated before the project scope is finalized.

Step 2: Project Scoping & API Setup

Once the customer has reviewed the initial reconstructions and agreed to proceed, Third Dimension AI works with the customer to scope the full project: defining the scenarios to be reconstructed, the world-modeling capabilities required, and the integration touchpoints with the customer's existing simulation infrastructure. A dedicated API is then provisioned, giving the customer programmatic access to rendered environments, novel viewpoints, dynamic scenario generation, and sensor output synthesis. The API is designed to integrate cleanly with existing simulation frameworks and data pipelines, positioning SuperSim as an additive high-

fidelity layer rather than a wholesale toolchain replacement.

Step 3: Ongoing Support

Third Dimension AI provides ongoing support once a SuperSim integration is live. This includes monitoring simulation quality, assisting with scenario configuration, and ensuring that SuperSim-enabled simulations continue to function correctly as the customer's sensor hardware, software stack, and data collection program evolve. The objective is a stable, continuously improving simulation capability that scales alongside the customer's development program.

Results and Discussion

SuperSim delivers a fundamentally different simulation capability profile compared to legacy solutions. The following outcomes have been observed across customer integrations.

Failure Debugging in Exact Context

When an autonomous system encounters an unexpected failure in the field, understanding the root cause requires the ability to re-run the precise conditions of that incident with full observability. SuperSim enables teams to reconstruct the exact scene of a real-world event (sensor geometry, environmental conditions, agent configurations, and more) and replay it with parametric variation of individual factors to isolate causation. This closes a critical gap in the post-incident analysis workflow that generic synthetic environments cannot address.

Regression Testing with Behavioral Variation

SuperSim's dynamic object placement and agent behavior scripting capabilities support systematic regression testing: teams can re-run reconstructed scenarios with modified agent trajectories, alternative object configurations, or updated sensor parameters to verify that software updates produce the intended improvements without introducing new failure modes.

Safe Testing of Rare and Dangerous Edge Cases

Scenarios that are rare, dangerous, or impractical to reproduce in the physical world (eg. sudden obstacles, adverse weather, sensor failures, adversarial pedestrian behavior) can be constructed and executed safely in SuperSim. Because the baseline environments are grounded in real captured data, the resulting edge-case scenarios are statistically

representative of the conditions the deployed system will actually encounter, closing the generalization gap that afflicts fully synthetic edge-case libraries.

Scalable Scenario Libraries

A single real-world capture session can be expanded into thousands of distinct simulation scenarios through parametric variation of objects, behaviors, environmental conditions, and viewpoints, entirely without artist involvement. This transforms the economics of scenario library construction: rather than requiring weeks of manual effort per scenario, a single data collection seeds an entire library programmatically and continuously.

Repeatable Virtual Test Environments

SuperSim supports the construction of standardized virtual test facilities, such as repeatable tracks, intersection layouts, and controlled scenarios – reconstructed directly from real locations. These facilities provide the consistency and reproducibility required for rigorous cross-build benchmarking and regulatory validation workflows.

Continuous Data Flywheel

Integrating SuperSim also enables customers to feed data collected from real-world operations continuously back into their autonomy stack. Each new capture expands the simulation library, and the refined scenarios in turn improve the models that perform real-world inference, creating a compounding data flywheel that accelerates system improvement in real time.

Time & Resources Conserved

Across all of these use cases, the most immediately impactful differentiator is delivery speed and cost. Where legacy simulation pipelines require days or weeks and large teams to produce a usable environment from collected sensor data, SuperSim's reconstruction pipeline delivers simulation-ready scenes in hours. This changes the economics of simulation fundamentally, enabling teams to iterate at the cadence of software development, not content production.

Conclusions & Implications

Third Dimension AI is already working with several automotive customers who have integrated SuperSim to uplevel their simulation stack.



These deployments demonstrate that neural simulation complemented with world modeling is not simply a faster or cheaper way to do what existing tools already do, but that it enables a completely new simulation approach that legacy approaches simply cannot support. With SuperSim, automotive manufacturers benefit from:

- Reality-based simulation: Start from real captured environments rather than synthetic guesses, closing the domain gap at its source
- Faster iteration: Turn days or weeks of environment creation into hours, enabling simulation to keep pace with software development
- Better real-world performance: Reduce sim-to-real failures for autonomous vehicle teams by grounding training and validation in photorealistic, sensor-accurate environments

SuperSim fundamentally reimagines how simulation works, so it is not merely a visual, cost, or speed-of-delivery improvement. By layering world models on top of neural reconstructions, SuperSim is a full-stack rethink of how autonomy systems learn, test, and improve. When simulation is fast enough, realistic enough, and generative enough to operate at the pace of software development, the entire paradigm for how autonomous systems are developed, validated, and certified becomes open to revision.

Third Dimension AI is building the infrastructure that makes that revision possible.

An Empirical UDP Network Assessment of Real-Time Latency and Data Integrity for a Distributed Heterogeneous Simulation Setup

Machiel van der Stelt¹, Thomas Bräunl¹, Paul Roberts¹

(1) The University of Western Australia, 35 Stirling Hwy, Crawley WA 6009, Australia, e-mail: 22888094@student.uwa.edu.au, {thomas.braunl, paul.l.roberts}@uwa.edu.au

Abstract – Most road safety studies involving Virtual Reality (VR) Driving Simulators (DS) focus on single road users interacting with their environment or automated vehicles. However, in reality there are interactions between multiple road users (e.g., cars and cyclists) and understanding those interactions requires connected, independent driving simulators. This feasibility study assesses whether a real-time, peer-to-peer User Datagram Protocol (UDP) connection over Ethernet can reliably and accurately link a heterogeneous setup consisting of a pre-existing Unity3D bicycle simulator and a pre-existing SCANeR Studio car simulator for road safety interactions.

Existing literature offers limited guidance on connecting independent driving simulators and connection stability due to the complexity and niche application of such setups.

A 27-second experiment with two Raspberry Pis both with Raspberry Pi Operating System (OS) installed, demonstrated that all isolated 8,012 67-byte packets were successfully returned by the echo server to the client, with 100% bit-level data integrity. It was observed that the mean of the Round-Trip Time (RTT) had an acceptable 2.61 ms; however the Standard Deviation (SD) was 1.99 ms, which was the result of highly variable RTT values.

Keywords: VR, Road-Safety, Raspberry Pi, UDP, peer-to-peer.

Copyright: © 20xx by the authors.
Licensee Driving Simulation Association.

Introduction

This feasibility study tests if two pre-existing heterogeneous DS can be connected peer-to-peer for real-time interaction. The first DS is a custom-built bicycle DS with Unity3D, and the second is a car DS with SCANeR Studio. Both systems run on their own host computers. This configuration has limited research available applied to road safety studies.

A study of a complex simulation system that explains the UDP Interface is easy to set-up, but if computing power needs to be preserved the more complex FPGA Interface can be used Scheibe, et al. (2022). The expectation is that UDP is good enough for the proposed set-up.

A study of an advanced UDP configuration is compared with a Transmission Control Protocol (TCP) setup. The UDP had better latency, while the TCP showed higher reliability Zhao, et al. (2025). The proposed setup for this study is used for streaming location data, where missing a few percent of datagrams is not an issue.

Methods

Both the Raspberry Pi 3Bs units ran Raspberry Pi OS and were connected via a peer-to-peer standardized Ethernet (UDP) connection for a latency and data integrity experiment to test real-time connection capability. Matching the real-life setup, the client Raspberry Pi had its script written in Python, as used in a pre-existing SCANeR Studio DS, and the other Raspberry Pi used as the echo server had its script written in C#, as used in a pre-existing Unity3D DS. Total of 8,012 isolated echo packets were sent from the client to the echo server and returned with the latency and data integrity recorded. Each UDP packet had a total size of 67 bytes. This included a 21-byte payload ASCII-encoded. Of this payload, 13 bytes represented a 13-digit timestamp, with the remaining 8 bytes being the static string 'Test at '. The UDP headers and trailers contributed 46 bytes. At any moment there was only one packet occupying the network to guarantee controlled packet transmissions and isolated RTT value collections.

Results

Fig. 1 shows that in the 27-second experiment, the mean RTT is 2.61 ms and that the SD is 1.99 ms, which is low enough to cover animation screen rates. Also, the RTT time extends to 7 SD to the right of the mean. This demonstrates a large variance of the RTT, which largely comprises T1, T2 and T3 as shown in Eq. 1.

$$RTT = T1 + T2 + T3 \quad (1)$$

T1 is largely determined by the physical transmission latency, starting from the moment a packet is being sent from the client and being received by the echo server. T3 is determined in the same manner for the return message. T2 is the time the echo server needs for required software processes between receiving a packet and returning it. T2 is determined by the OS (ΔT_{OS}) and different application software processes (ΔT_{app}), as shown in Eq. 2.

$$T2 = \Delta T_{app} + \Delta T_{OS} \quad (2)$$

Fig. 2 shows that the jitter exhibits notable variability. The RTT variance likely shares a common origin with the stochastic jitter generated locally by the client Raspberry Pi itself. Fig 3. clearly shows that the RTT, as shown with the grey line, has excessive variance in both frequency and amplitude. The rolling average of 30 successive values was applied to both the RTT mean and the RTT SD, as shown in the blue and red lines respectively in Fig. 3. This revealed no long-term trend in the RTT but rather stochastic variability. Also, the rolling SD represents local variance, exhibiting purely stochastic behaviour.

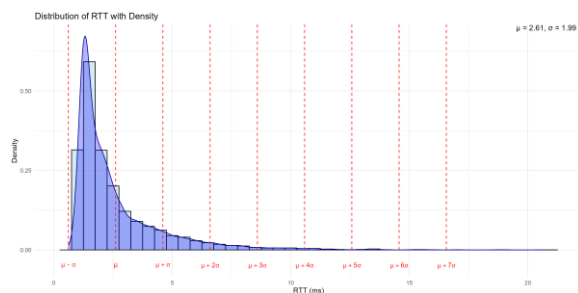


Figure 1: RTT distribution with density

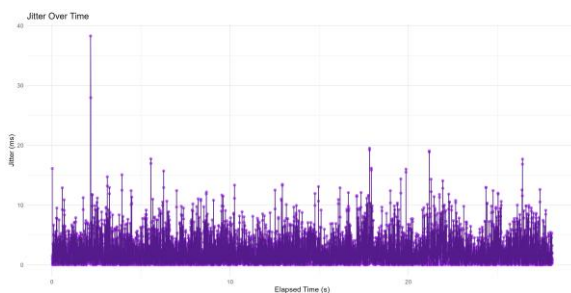


Figure 2: Client-side jitter over time

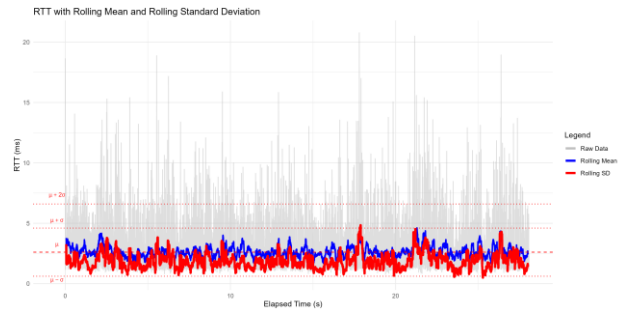


Figure 3: RTT Rolling Mean, Rolling SD & raw data over time

Conclusions

The results show that all 8,012 67-byte packets, transmitted in isolation to ensure controlled network conditions, were successfully echoed over approximately 27 seconds between the Python client script and the C# echo server script running on their respective Raspberry Pis, with 100% bit-level data integrity. The RTT showed satisfactory time to cover animation software screen rates, while the stochastic jitter, RTT variance and subsequent SD were considerable. The RTT variance is likely driven by T2, arising from both the OS and application latency, which potentially mirrors the stochastic jitter observed on the client side.

Acknowledgements

This research was supported by an Australian Government Research Training Program (RTP) Scholarship. Thanks to advisor Thomas Bräunl and co-supervisor Paul Roberts for their support. Thanks to David Harries for proofreading and general guidance.

Data availability: <https://doi.org/10.17632/d9jxc4jyzd>

References

- Scheibe, C., Kuri, A., Feng, Y., Zhao, L., Xiong, X., La Seta, P., Peng Liang, X., Knödtel, J., Holzinger, P., Reichenbach, M., Mehlmann, G., 2022. Interfacing real-time and offline power system simulation tools using UDP or FPGA systems, *Electric Power Systems Research, Elsevier*.
- Zhao, S., Zhang, Q., Zhao, Q., Zhang, X., Guo, Y., Lu, S., Song, L., Zhao, Z., 2025. Enhancing Bidirectional Modbus TCP ↔ RTU Gateway Performance: A UDP Mechanism and Markov Chain Approach, *Sensors, MDPI*.

COMPANIES' INDEX

A

AB Dynamics	17
Applied Intuition	39
ASAM e.V	29
Aumovio SE	17
AVSimulation	79

B

BMW AG	17
--------	----

D

Dassault Systèmes	45
Deontic BV	111
Drexel University	117
Dynisma Ltd	9

F

Ford Otosan	39
-------------	----

G

German Aerospace Center	17
Graz University of Technology	9

H

HONDA R&D Co., Ltd.	17
---------------------	----

I

IPG Automotive	57
IRT SystemX	23, 93

J

Jitsik	117
--------	-----

M

Magna Steyr Fahrzeugtechnik	9
MathWorks	33
MdynamiX AG	71
MetoDrive	117
Munich University of Applied Sciences	71

N

Nissan Motor Co., Ltd.	17
------------------------	----

O

Offenburg University of Applied Sciences	105
OKTAL SE	79, 93

R

RWTH Aachen University	17
------------------------	----

S

SUBARU Corporation	17
S&VL Inc.	17

T

Technical University of Munich	17
Third Dimension AI	119
TOYOTA MOTOR CORPORATION	17

U

University of Iowa	65
University of Leeds	101
University of Warwick	51
University of Western Australia	123

V

Valeo Internal Automotive Services Center	85
Valeo Schalter und Sensoren GmbH	85
VI-grade	15
VRCentrs	117

W

Würzburger Institut für Verkehrswissenschaften	17
--	----

