vehicles

MDPI

Article

Development of a 6-DoF Driving Simulator with an Open-Source Architecture for Automated Driving Research and Standardized Testing

Martin Meiners *,†, Benedikt Isken † and Edwin N. Kamau

Institute of Automotive Engineering, University of Applied Sciences Cologne, Betzdorfer Straße 2, 50679 Cologne, Germany; benedikt.isken@th-koeln.de (B.I.); edwin.kamau@th-koeln.de (E.N.K.)
* Correspondence: martin.meiners@th-koeln.de
† These authors contributed equally to this work.

Abstract

This study presents the development of an open-source Driver-in-the-Loop simulation platform, specifically designed to test and analyze advanced automated driving functions. We emphasize the creation of a versatile system architecture that ensures seamless integration and interchangeability of components, supporting diverse research needs. Central to the simulator's configuration is a hexapod motion platform with six degrees of freedom, chosen through a detailed benchmarking process to ensure dynamic accuracy and fidelity. The simulator employs a half-vehicle cabin, providing an immersive environment where drivers can interact with authentic human–machine interfaces such as pedals, steering, and gear shifters. By projecting complex driving scenarios onto a curved screen, drivers engage with critical maneuvers in a controlled virtual environment. Key innovations include the integration of a motion cueing algorithm and an adaptable, cost-effective open-source framework, facilitating collaboration among researchers and industry experts. The platform enables standardized testing and offers a robust solution for the iterative development and validation of automated driving technologies. Functionality and effectiveness were validated through testing with the ISO lane change maneuver, affirming the simulator's capabilities.

Keywords: driving simulator; XiL-simulation; automated driving; motion platform; validation

check for updates

Academic Editors: Fernando Viadero-Monasterio, Daniel Garcia-Pozuelo, Basilio Lenzo, Pablo de Villeros and José Ramón Trillo

Received: 16 July 2025
Revised: 14 August 2025
Accepted: 18 August 2025
Published: 21 August 2025

Citation: Meiners, M.; Isken, B.; Kamau, E.N. Development of a 6-DoF Driving Simulator with an Open-Source Architecture for Automated Driving Research and Standardized Testing. *Vehicles* **2025**, *7*, 86. https://doi.org/10.3390/vehicles7030086

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Driver-in-the-Loop (DiL) simulators incorporate human drivers into the simulation process, enabling real-time interaction and feedback mechanisms. These simulators are particularly beneficial for evaluating advanced driver assistance systems (ADAS) [1,2], as well as autonomous driving systems, since they offer standardized testing environments that comply with safety regulation requirements [3,4]. Moreover, DiL simulators play a crucial role in assessing vehicle and hardware performance [5,6] and serve as valuable tools for training and educational purposes [7,8].

Existing driving simulators come in a variety of forms, ranging from commercial ready-to-use systems [9–11] to large-scale research platforms [12–14] and more modest academic setups [15–17]. Industry leaders offer sophisticated, full-fledged simulator solutions often including proprietary software tightly integrated with their hardware platforms. These commercial systems generally feature closed architectures, with software and hardware optimized to work exclusively together, limiting user flexibility in modifying or expanding

the system. In contrast, large university-based simulators represent complex facilities with advanced capabilities including motion platforms, immersive visual systems, and real vehicle cockpits. While these academic simulators often provide relatively open interfaces for research customization, their construction requires substantial institutional resources, technical expertise, and often collaboration with specialized companies, making them challenging to replicate for smaller institutions or independent researchers. Furthermore, much of the published literature on driving simulators describes fixed-base simulators equipped with standard input devices like steering wheels, pedals, and multiple screens or head-mounted displays, but lacking sophisticated motion platforms.

The primary objective of our work is to present the development of a versatile DiL simulation platform, specifically designed to test and analyze automated driving functions. Our approach emphasizes an open-source architecture, ensuring broad accessibility and flexibility. This aligns with the growing demand for cooperative and multidisciplinary research endeavors that can adapt to evolving technological landscapes.

Leveraging a hexapod motion platform, our simulator supports comprehensive interaction with complex driving scenarios, including critical events such as the ISO lane change test, within a meticulously calibrated virtual environment. This configuration ensures the accurate simulation of diverse driving conditions, thus enhancing the realism and fidelity essential for precise evaluations. Additionally, by maintaining a balance between performance and ease of use, our platform promises to be a powerful tool for both research and industrial applications, encouraging the continuous advancement of automated and autonomous driving technologies.

2. Related Work

DiL simulators are designed with varying attributes and structural characteristics to meet key requirements, determined by the type of use case. Those key requirements address the following specific aspects of the simulation environment:

Realism and Fidelity: Ensuring that the simulator provides realistic cues and feedback to the driver is imperative. This involves precise vehicle dynamics, authentic motion feedback and high-quality visual and auditory environments [18,19]. The fidelity and realism of a simulator are intrinsically linked to both absolute and relative validity, and the exploration and comprehension of this relationship remain active areas of research [19]. It is crucial to define the necessary level of required realism and validity for each testing scenario, acknowledging that achieving 100% realism is neither required nor practically feasible. As an example, a DiL simulator dedicated specifically to vehicle dynamics necessitates a higher degree of dynamic accuracy and fidelity compared to one designed primarily for testing certain types of ADAS. Regarding the DiL simulator that is developed in this work, it requires the representation of dynamics from often occurring driving scenarios in urban areas, highway roads and rural areas including edge cases like a lane change maneuver. Additional to the movement, a realistic look and feel is required, as well as realistic visual feedback.

System Integration and Open Architectures: This involves the seamless integration of various hardware and software components, typically through modular and scalable architectures [5,20]. As illustrated in common architectural frameworks, it is essential to design these systems with the driver as the central element in a feedback loop, ensuring that inputs and outputs are linked to the driver's interactions and responses [21–23].

Latency and Real-Time Performance: Minimizing system delays is crucial to ensure that driver input is accurately reflected in the simulation in real time [24]. Potential disturbances can arise from various factors. These include physical factors (spring and damper effects, inertia, friction), electrical issues (signal rate limitations, errors, disturbances) and

computational challenges (signal and data processing speeds, data exchange protocols). A study by [25] investigated different latencies in driving platform movement and human perception; their results show that distinctions below 40 ms are only identified by 20% of the population, with their state-of-the-art simulator showing a default latency of 30 ms. Based on this, the target for transport delay is a maximum latency of 40 ms.

3. System Architecture Design

Considering the requirement for an open system mentioned in Section 2, the developed system architecture for the driving simulator is depicted in Figure 1, encompassing various subsystems that work together to create a realistic driving simulation. Key elements of this architecture include the human–machine interface (HMI), graphics interface, vehicle-/environment simulation, simulation data interface, motion platform control software, the physical motion platform, and the human driver. For clarification: The term "vehicle-/environment simulation" refers to "driving simulator software", i.e., software that calculates the physics of vehicles, enables traffic simulation, simulates the virtual environment and more. The term "driving simulator" is used for the complete system developed in this paper and described in the system architecture.

In the application case of a DiL simulation, the HMI is used by the driver through the operation of pedals, steering wheel, gear shifter and other input devices to send control commands to the vehicle represented in the simulation. Within the simulation, the dynamic behavior of the vehicle, based on predefined vehicle and environment models, is graphically displayed through the graphics interface. Furthermore, feedback cues such as steering wheel torque (also known as force feedback) or pedal force are sent back to the driver, depending on the underlying calculation model and driving conditions. Additionally, the generated vehicle telemetry data—especially in the form of translational accelerations and rotational velocities—is transmitted as a signal to the simulation data interface. A central component of the simulation data interface is the motion cueing algorithm (MCA), which converts the simulated vehicle telemetry into target positions for the motion platform. These motion cues serve as feedback for the driver. A significant distinction from conventional architectures is the integrated interface for sensors, which allows for the extraction of actual motion platform movements or driver behaviour using electroencephalography (EEG), electrocardiogram (ECG), or eye-tracking devices. By adding diagnostic and monitoring software interfaces, the data can be processed and evaluated in the context of the simulated driving scenario.

Overall, the open system architecture enables a modular approach, where each interface can be replaced by another. This supports comparative tests and studies of different MCA, HMI, as well as motion platforms and vehicle-/environment simulations.

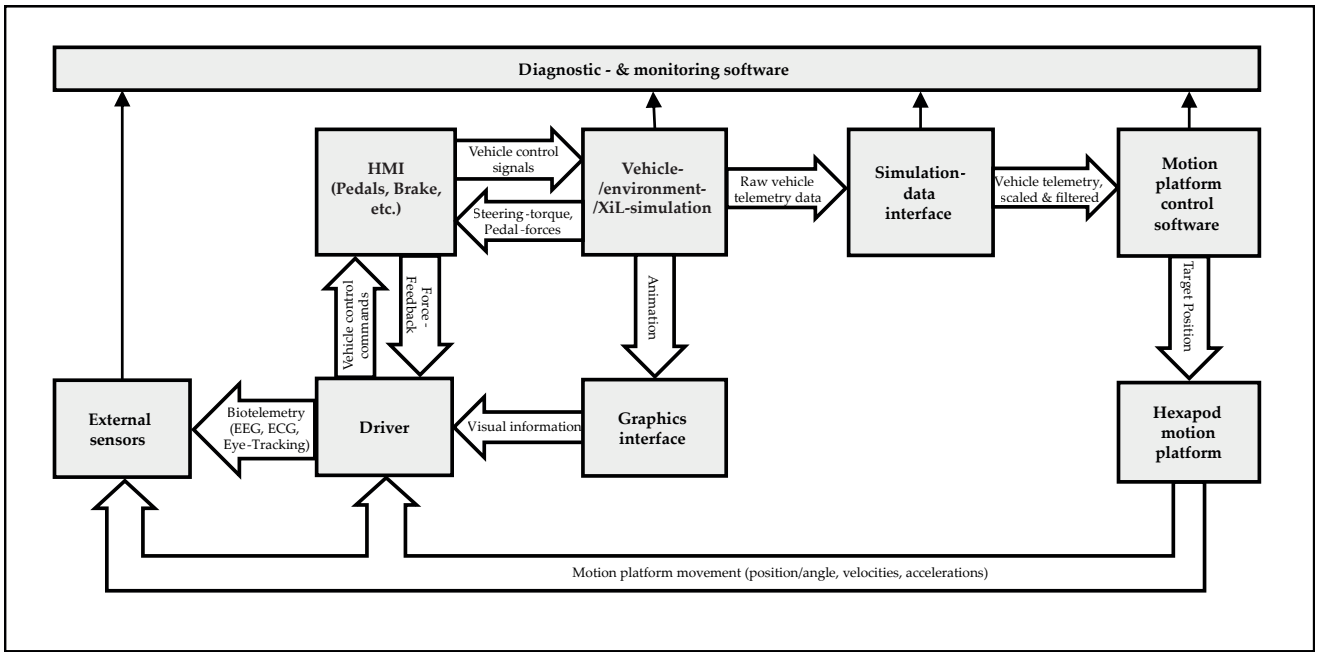


Figure 1. Architecture of the main simulator environment. The common Driver-in-the-Loop architecture is enhanced with external sensors and diagnostic interfaces. The open interface architecture ensures a high degree of versatility and flexibility in switching between components, facilitating comparative tests.

3.1. Vehicle-/Environment Simulation Software

As stated in the previous section, the approach of the architecture is to enable different vehicle-/environment simulation software to work with the driving simulator. In recent years, a considerable number of novel vehicle-/environment simulations have been developed. Ref. [26] conducted a comprehensive review of well over 100 simulation software programs, providing a detailed catalog of these recent developments. While not all of them are suited for application in the driving simulator architecture, a comprehensive comparison of all vehicle-/environment simulation software would exceed the scope of this paper. Therefore, five vehicle-/environment simulations where prechosen and reviewed based on possible integration into the driving simulator and on ADAS and automated driving (AD) research capabilities. The following four subcriteria were chosen to evaluate the vehicle-/environment simulations.

Graphics: In order to enhance the level of immersion in the simulator, it is essential to achieve a high-quality visual simulation. The manner in which the visual environment is implemented in the driving simulator (i.e., screen-based or head-mounted display) determines the necessity for the software to provide a method for applying the graphics to the visual environment. This can be achieved through the use of an API or a virtualization UI provided by the system. The "Animation" connection between the modules "Vehicle-/environment/XiL-simulation" and "Graphics interface" in Figure 1 emphasizes this.

Interfaces: The vehicle-/environment simulations, as shown in Figure 1, need to communicate with HMI devices and the simulation data interface. This is necessary to give feedback and motion cues to the driver as well as control signals back to the simulation. A well-fitted software for the driving simulator provides different interfaces to achieve these connections. These can be through APIs (i.e., Python, C++, etc.) or frameworks and protocols (i.e., ROS, CAN, etc.).

Sensors: Research in the field of ADAS and AD in simulation require well-simulated sensors. A software offering a wide range of sensors for ADAS/AD with advanced physics models is a requirement for the driving simulator.

Scenario-based Testing: Testing and validation of ADAS/AD requires simulation of a variety of scenarios. A comprehensive vehicle-/environment simulations software for ADAS/AD should provide a way to create unique scenarios through tools or APIs.

As illustrated in Table 1, a total of five vehicle-/environment simulations were evaluated for the driving simulator. The filled circle indicates that the software fully complies with the specified criteria, whereas the empty circle signifies an absence of compliance or achievement of the criteria. The two vehicle-/environment simulations sticking out in this comparison are CARLA and CarMaker. Both software applications provide very high-quality graphics and different interfaces to communicate with the software. They also provide a wide range of sensors to add to a vehicle in simulation. Scenario creation is possible in both CARLA and CarMaker, with both supporting the OpenScenario standard [27]. Carmaker has a slight edge providing a GUI tool to create scenarios. While CARLA due to it being open source and developers being able to build it from source means scenario creation possibilities are extensive.

Table 1. Comparison of five popular vehicle-/environment simulation software based on criterias of integration into the driving simulator and advanced driver assistance systems (ADAS)/automated driving (AD) capabilities. * These softwares are no longer maintained [28–32].

Simulation	Open Source	Integration		ADAS/AD Capabilities	
		Graphics	Interfaces	Sensors	Scenario-Based Testing
CARLA	Yes	●	●	●	◕
AirSim *	Yes	◑	◕	◑	◑
Gazebo	Yes	◔	●	◕	◕
CarMaker	No	●	◕	●	●
OpenDS *	Yes	◑	◑	◔	◑

Prior to the selection of vehicle-/environment simulation software for the driving simulator, Simulink was chosen to implement the system architecture. The decision was taken on the basis of specific features of Simulink, in particular its ability for visual block modelling, its versatility in integrating different systems, and its real-time capability. The aforementioned decision exerts a significant influence on the selection of vehicle-/environment simulation software. The decision to utilize CarMaker as the primary software for this study was made on the basis of its provision of a native Simulink interface and a flexible visualisation tool for graphic animation with MovieNX. A comprehensive study is planned, encompassing various vehicle-/environment simulations. These simulations will be integrated into the driving simulators system and assessed for their efficacy in ADAS/AD research.

3.2. XiL Testing

Additionally, real vehicle hardware and software can be integrated to enable X-in-the-Loop (XiL) testing, fostering collaboration across various domains and technologies. As part of this effort, the conceptual integration of a Speedgoat HiL simulator has been investigated and is planned for implementation in the near future. This integration will enable real-time simulation and emulation of autonomous driving functions, hardware-based bus communication, and restbus simulation. The system is based on a Speedgoat 38U rack configuration, featuring two performance and one baseline real-time target machine enabling rapid control prototyping. It supports high-speed Ethernet communication (UDP, TCP/IP, EtherCAT, XCP), provides eight CAN interfaces (CAN, CAN-FD, CANopen), 14 digital I/O channels, 16 analog I/O channels, as well as PWM input and output channels.

In addition, the platform includes an AI-ready, freely programmable FPGA to support advanced processing and custom interface development.

4. Development

4.1. Implementation of the Driving Simulator Operating Environment in Simulink

Implementing the concept of the system architecture, the Driving Simulator Operating Environment (DSOE) is developed using Matlab Simulink R2022b (see Figure 2). Similar to the main architecture principle shown in Figure 1, HMI signals from shifter (1), pedals (2), and steering wheel (3) are transmitted to the vehicle-/environment simulation (4). This simulation operates using IPG CarMaker 12.0 and its Simulink interface, CarMaker4Simulink. The simulated vehicle telemetry signals are sent to the simulation data interface (5), where a Classical Washout Algorithm (CWA) scales, filters, and integrates the data before transmitting it to the motion platform control software. This is achieved by integrating the ForceSeatMI API (a machine interface from the company that developed the platform) into a S-Function block. In this application, the motion platform control software is a standalone solution (not within Simulink) and receives the required positions, controlling the platform's end effector to generate movements.

Diagnostics and monitoring software is added in two ways. The first involves adding internal sensors (6) that monitor objects and variables defined through the vehicle/environment simulation. The second involves integrating external sensors (7) using Vector's CANopen APIs, allowing sensors to be accessed via CANopen. In both cases, the data is stored in workspace variables and can be accessed immediately after a simulation. The presented DSOE implementation enables testing of the overall system architecture concept. To ensure real-time performance during simulations, each interface task is managed by separate computers using UDP and ROS 2.

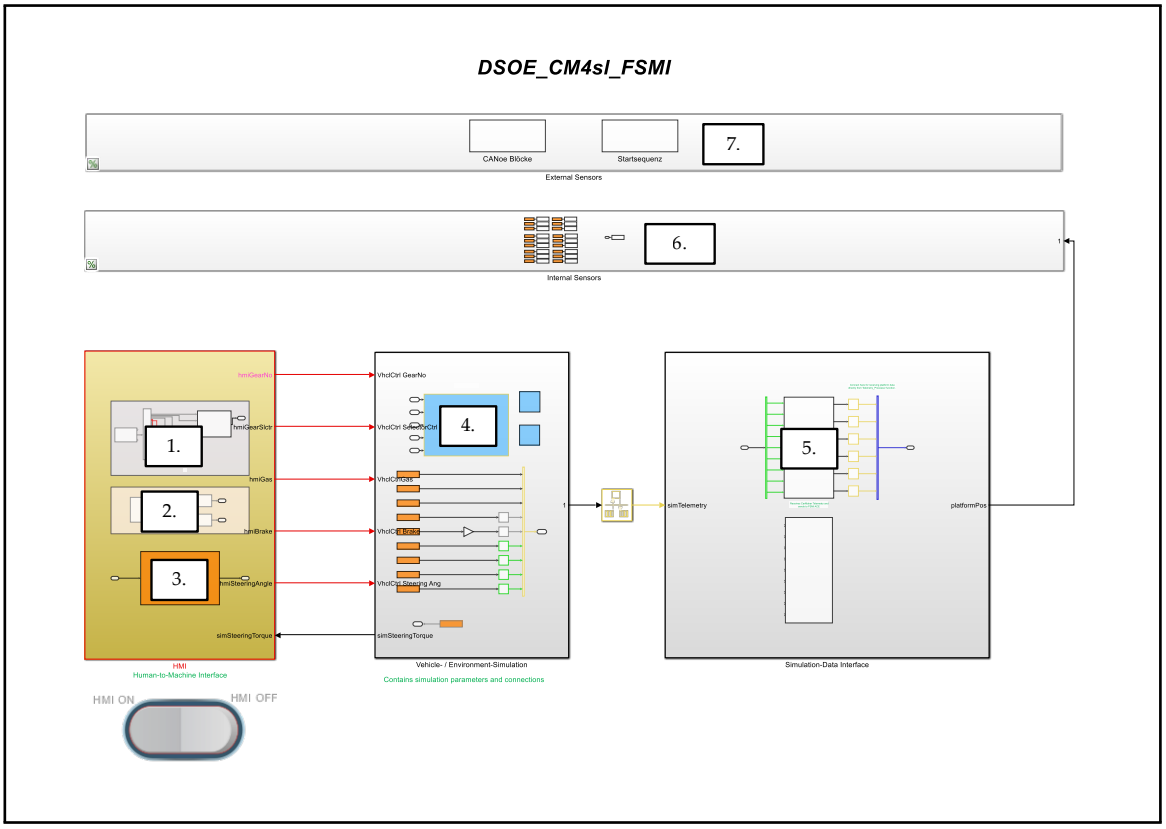


Figure 2. Simulink implementation of the driving simulator operation environment. The human–machine interface signal-processing subsystem consists of (1): gear shifter signal subsystem, (2): pedal

signal subsystem, (3): steering wheel signal subsystem. Further subsystems: (4): vehicle-/environment simulation subsystem, (5): simulation data interface subsystem, (6): internal sensor subsystem, (7): external sensor subsystem.

4.2. Motion Platform Choice

Based on the previously established criteria, the hexapod motion platform PS-6TL-800 from Motion Systems was chosen. In addition to fundamental requirements such as payload capacity, cost considerations, usability and functional safety, specific demands concerning immersion and realism were also addressed. These include the need to provide realistic motion feedback across various typical driving scenarios, effectively distinguish between normal driving conditions and edge cases, and enhance the immersion level and sensitivity of test drivers to hazardous scenarios. Consequently, it was crucial to compare the dynamic characteristics of each potential motion platform to ensure these requirements were met.

In translational platform dynamics, there is a notable interdependency among displacement, velocity and acceleration. Predominant algorithms, including the CWA, provide certain motion cues through acceleration. However, due to the limitations in platform dynamic range, both the duration and efficacy of these cues are constrained. Consequently, in vehicle simulations, medium- to high-frequency and initial acceleration scenarios (such as those from transient vehicle responses) are cued through translational platform dynamics. This is also known as acceleration-onset cueing [33].

To accurately assess the translational behavior, merely examining the static characteristics would be insufficient. An analysis encompassing the interdependencies of displacement, velocity and acceleration is necessary. Therefore, a comprehensive approach similar to [34] accompanies the benchmarking of motion platforms. For the benchmark, 20 hexapod motion platforms were initially examined, out of which 9 were selected for further consideration due to data availability and alignment with the previously stated requirements. The detailed benchmark data can be found in Table A1.

Figure 3 illustrates the varying dynamic longitudinal and lateral capabilities a_{trans} concerning a single, undisturbed degree of freedom from a standstill, using the relationship between frequency f and limiting nominal platform position p_{nom}, velocity v_{nom} and acceleration a_{nom} (c.f. [34]):

$$a_{trans} \leq \min\left\{ p_{nom} \cdot (2\pi \cdot f)^2, v_{nom} \cdot 2\pi \cdot f, a_{nom} \right\}. \tag{1}$$

The acceleration amplitudes are plotted in a double logarithmic scale over frequency. The evaluated hexapod can reproduce all combinations of acceleration amplitudes and frequencies that fall below each line. For clarity, only one of the highest and one of the lowest achieving platforms are depicted to establish the benchmark boundaries. Within these boundaries, the PS-6TL-800 is displayed, demonstrating that it ranks among the better-performing platforms in this analysis.

The available longitudinal and lateral accelerations are initially constrained by a frequency-independent maximum acceleration of $a_{nom} = a_{max} = 7\ \mathrm{m/s^2}$, as represented by the horizontal line. For these one-dimensional motions, the velocity can be determined according to Equation (2) as the time integral of the product of acceleration and angular frequency:

$$v(\omega) = \int a \cdot \sin(\omega t)\, dt = -\frac{1}{\omega} \cdot a \cdot \cos(\omega t) + v_0\big|_{=0} = -\frac{1}{\omega} \cdot a \cdot \cos(\omega t). \tag{2}$$

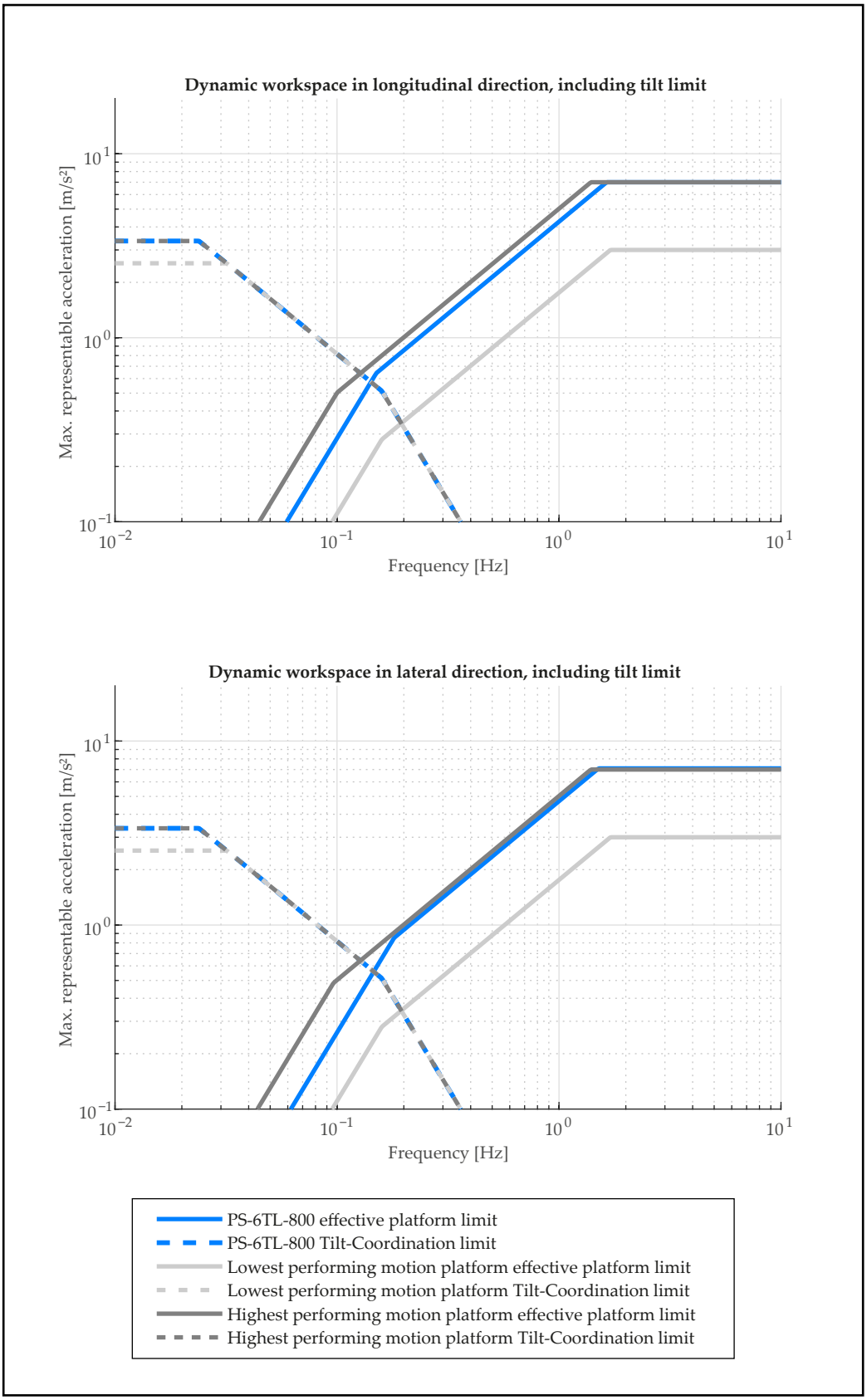


Figure 3. Longitudinal and lateral dynamic workspace benchmark. The motion platforms can replicate acceleration combinations below each line, with platforms like the PS-6TL-800 (blue) highlighting above-average performance.

By rearranging and substituting the maximum velocity, as well as using $\omega = 2\pi f$ and $|\cos(\omega t)| \leq 1$, the maximum lateral acceleration $a_{trans,v}$ as a function of acceleration frequency f can be represented as shown in Equation (3) (c.f. [34]):

$$a_{trans,v}(f) \leq v_{\text{nom}} \cdot 2\pi \cdot f. \tag{3}$$

At medium frequencies (approximately 0.15 Hz to 1.6 Hz for longitudinal and about 0.8 Hz to 1.5 Hz for lateral), the displayable acceleration amplitudes decrease and are constrained by the maximum platform velocities $v_{\text{nom,long}} = 0.68$ m/s^2 and $v_{\text{nom,lat}} = 0.75$ m/s^2. At lower frequencies, specifically up to 0.15 Hz for longitudinal motion and 0.8 Hz for lateral motion, the nominal acceleration of 0.85 m/s^2 is limited by the maximum displacement boundaries p_{nom} of the platform. This constraint arises because the frequency dependency is squared as a result of double integration of acceleration over time

$$a_{trans,\max}(f) \leq p_{\text{nom}}(2\pi \cdot f)^2. \tag{4}$$

Subsequently, lower frequency accelerations, such as quasi-static vehicle behavior during prolonged turns or constant cornering, are depicted by algorithm components like the tilt coordination of the CWA. This approach is subject to known limitations arising from human perception and platform tilt angles. To assess the effectiveness of acceleration representation through tilt coordination, the graphs illustrate the capabilities outlined in Equation (5) (c.f. [34]):

$$a_{tilt} \leq \min\left\{\sin\left(\frac{\ddot{\phi}_{sens}}{(2\pi \cdot f)^2}\right), \sin\left(\frac{\dot{\phi}_{sens}}{2\pi \cdot f}\right), \sin(\phi_{max})\right\} \cdot g. \tag{5}$$

Regarding [35], even though platforms can achieve greater rotational angles, the maximum tilt angle is restricted to $\phi_{\max} = 20^\circ$ to prevent false cueing. The graphs illustrate displayable maximum accelerations of approximately 3.36 m/s² for frequencies up to about 0.02 Hz; beyond this threshold, they are constrained by the perception threshold for rotation rates of $\dot{\phi}_{\text{sens}} = 3^\circ/\text{s}$. As frequency increases, the maximum acceleration decreases, primarily due to the rotational acceleration perception threshold of $\ddot{\phi}_{\text{sens}} = 3^\circ/\text{s}^2$ [34,36].

In the context of rotational dynamics, typical values for roll and pitch accelerations observed in vehicles as in [37] are compared against the maximum rotational performance of the platform. Generally, in vehicle dynamics applications, the roll and pitch movements are expected to be displayed without any need for scaling or filtering [34]. Consequently, the evaluation of rotational dynamics is conducted through straightforward comparisons of the peak angular velocities and accelerations, with results depicted in Figure 4. The diagram illustrates the maximum rotational dynamics of both the highest and lowest performing platforms, along with the average performance across all benchmarked models. The characteristic angular values of the PS-6TL-800 are highlighted in blue, indicating that it consistently exceeds the average in each movement category, thereby classifying it among the better-performing platforms.

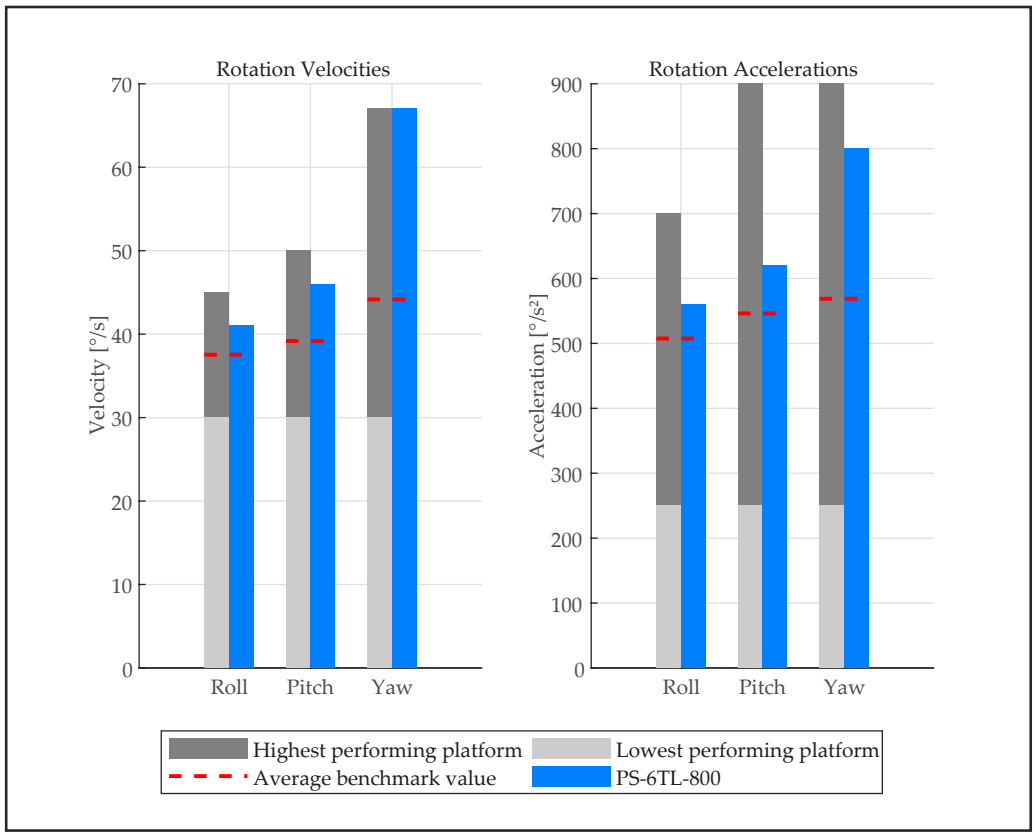


Figure 4. Benchmark results for rotational dynamic work-spaces. Out of 20 motion platforms examined, nine were selected for benchmarking. With rotational velocities up to 65°/s and accelerations up to 800°/s^2, the PS-6TL-800 motion platform surpasses average values of 44°/s and 568°/s^2.

4.3. Driver's Cabin and HMI Design

The main task of the driver's cabin is to create a realistic and immersive driving environment that gives participants the sensation of being in a real vehicle. Utilizing the dimensions of an actual vehicle is unparalleled in replicating the interior space of a real vehicle and allows for the integration of genuine HMI and thus simulation interfaces.

For this purpose, the front half of a Kia XCEED 2022 was prepared to serve as the cabin of the simulator. To reduce weight, unnecessary components were removed from the vehicle. The preparation process involves several meticulous steps: first, the rear interior section was disassembled using a four-post lift. Following this, the headliner, pillar panels, and panoramic glass sunroof were removed. Subsequently, the underbody panels, rear doors, and side sills were disassembled. Afterward, the drivetrain, including the engine and transmission, was dismantled, and the fuel tank and lines were removed. The vehicle was moved to a scissor lift, where the subframe, steering, and all parts of the front axle were disassembled. As shown in Figure 5, the rear was prepared to be separated from the front section, ensuring that the driver's cabin remains intact. The rear section was closed off to maintain structural stiffness. The construction of the cabin mounting took place, designed to comply with the motion platform's requirements for maximum payload and moment of inertia. The net weight of the constructed cabin is 497 kg, providing space for up to two individuals, each weighing up to 100 kg. Additionally, the cabin was equipped to integrate various HMI components and measurement devices as defined in the system architecture and Figure 6, further enhancing its realistic and immersive qualities.



Figure 5. Vehicle disassembly and preparation. The vehicle is cut behind the B-pillar and sealed off to maintain structural stiffness.

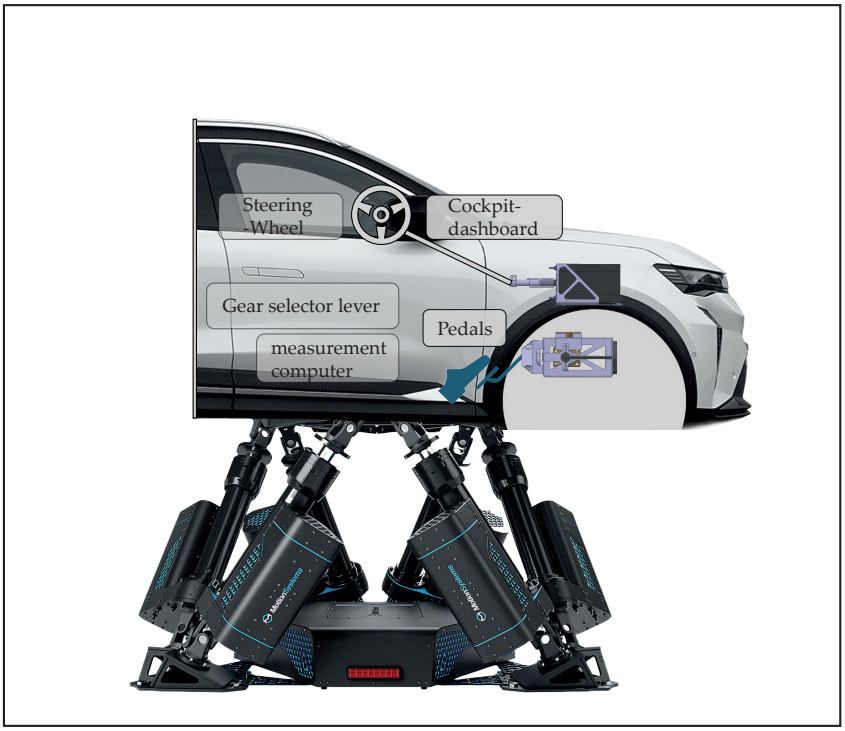


Figure 6. Schematic overview of vehicle human–machine interface components that are integrated into the Driver-in-the-Loop simulator.

4.3.1. HMI Architecture Design

The HMI architecture, depicted in Figure 7, is comprised of several key components: a sensor, an actuator (such as in the case of a force feedback wheel or haptic pedals), a microcomputer, and the DSOE that runs on a computer.

Initially, raw digital or analog sensor values are extracted and processed. These signals are communicated to the HMI computer using Microsoft's HID library. The Simulink DSOE contains embedded S-Functions that facilitate the operation of the HID interface. Subsequently, the data is translated into vehicle control signals suitable for the simulated vehicle. The method of data transfer from the microcomputer to the DSOE is contingent upon the specific type and transfer rate required. Depending on these factors, different communication protocols such as CAN, USB, or Ethernet can be employed to ensure efficient data transmission. In future iterations of this setup, a centralized HMI/sensor computer is envisioned. This centralized system would process all incoming data and distribute it to the desired nodes, thereby streamlining operations and potentially enhancing system efficiency and performance.

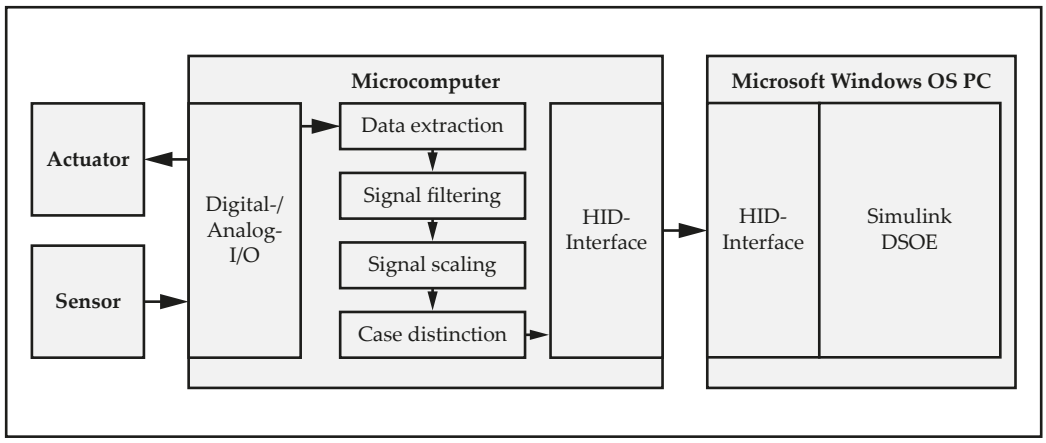


Figure 7. Human–machine interface signal processing architecture. The actuator/sensor signals are processed by a microcomputer and sent to the operating PC using a HID interface.

4.3.2. Pedals

While Figure 8 illustrates the schematics of the integrated brake pedal system, the real system is shown in Figure 9. The foundational hardware is the Ricmotech GTpro3 sim-racing braking system by Tilton. The brake cylinders (1) and the secondary cylinder (5) are retained in this system, including the pressure sensor (7). The original pedal assembly is not used in the simulator. Instead, both brake cylinders are interconnected through a balance bar (2), which enables the adjustment of brake balance between them. The primary distinction between the brake cylinders is the internal cross-section of their pressure chambers. The balance bar can be operated using a brake balance adjuster (3), offering the significant advantage of allowing adjustments from the driver's seat without modifying the installed brake pedals. This adjustment capability extends to being adjusted while driving, which is advantageous for direct comparisons.

The pedal force is conveyed to the brake cylinders via a push rod (4) that is guided by 3D-printed bearings. Both brake cylinders are filled with oil and one is hydraulically connected to the secondary cylinder. This secondary cylinder features an extended piston rod that retracts during use and is surrounded by interchangeable spring elements (6) that counteract its movement, creating pedal resistance. Adjustments to the balance bar and variations in the spring elements at the secondary cylinder can be made to influence the brake pedal feel. Additionally, the pressure sensor can output a signal to measure the brake pressure, allowing for the integration of these signals into the simulation to control the simulated vehicle by using the Ricmotec GTpro3 pedal control unit. During calibration, various spring elements are tested, allowing for the deduction of an initial pedal force-travel characteristic curve.

In addition, the original accelerator pedal is used. To achieve this, the pedal's integrated sensor circuit is electrically tapped. The signals are analog voltage signals and are directly compatible with the previously described pedal control unit.

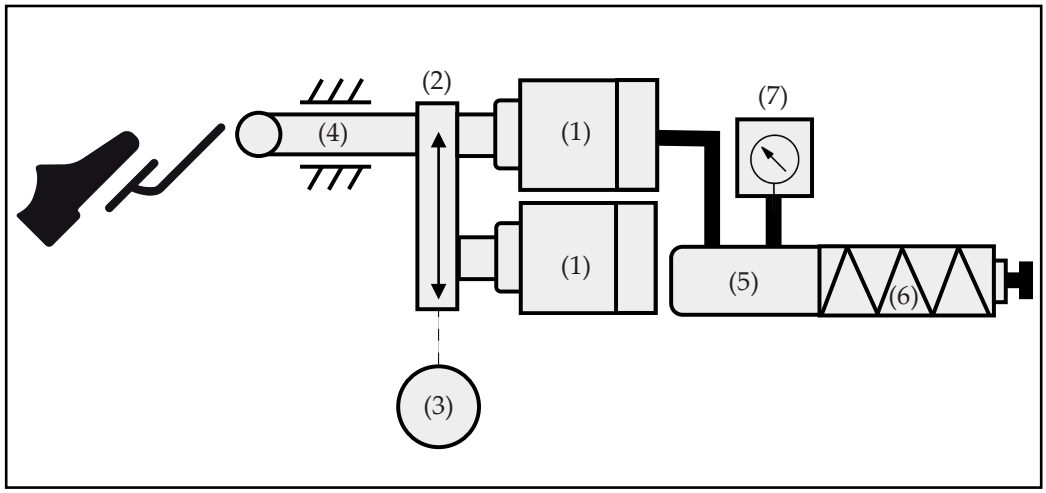


Figure 8. Schematic of brake pedal hardware. (1): brake cylinders, (2): balance bar, (3): brake balance adjuster, (4): push rod, (5): secondary cylinder, (6): spring elements, (7): pressure sensor.

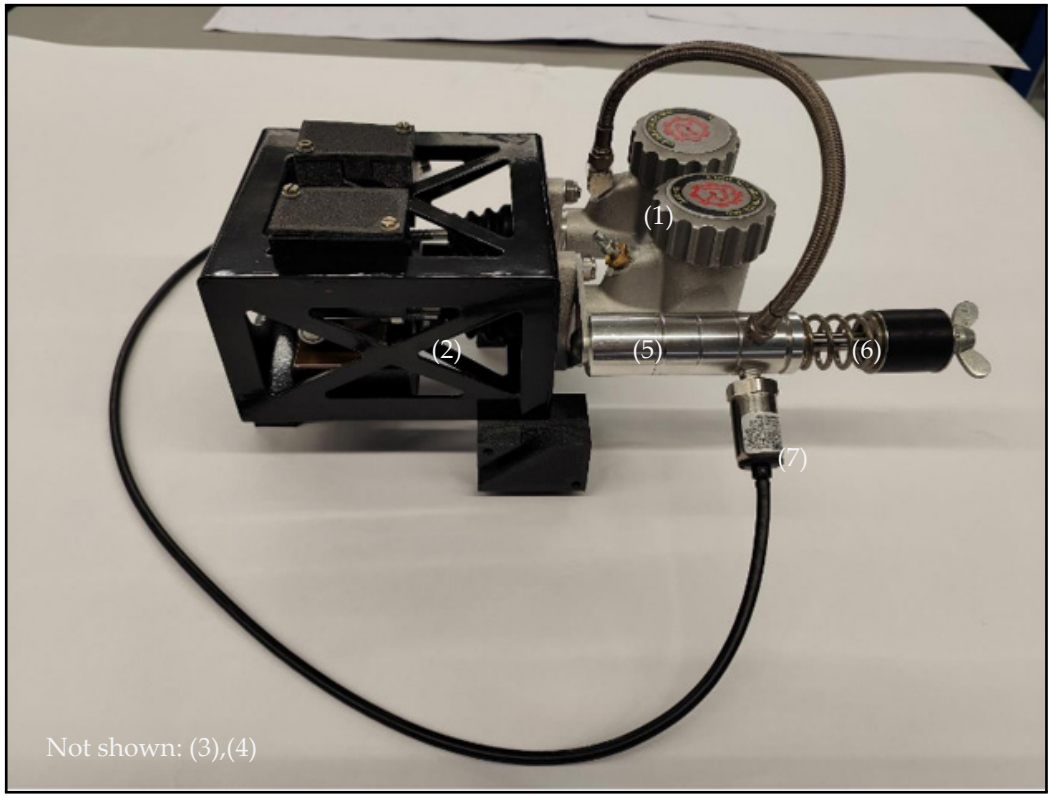


Figure 9. Implementation of the brake system hardware. (1): brake cylinders, (2): balance bar, (3): brake balance adjuster, (4): push rod, (5): secondary cylinder, (6): spring elements, (7): pressure sensor.

4.3.3. Steering

Regarding Figure 10, the steering system is built around the VRS DirectForce Pro Wheel Base, a force-feedback motor with integrated control electronics designed for digital motor sport applications. This system consists of an AC servomotor with an integrated encoder (1) and a controller (2), which together form the foundation of the operation. The motor and encoder are connected to the controller via separate signal cables (3), and the

controller itself is powered by its own dedicated power supply (5). To integrate the system into the simulator, the controller is linked using a USB interface (4), which facilitates software communication for controlling up to 25 Nm of torque and capturing wheel angle signals. To ensure compatibility with the existing setup, the original steering column (6) is preserved and the steering wheel (7) is mounted at the original position on the input shaft of the motor. The setup retains the original mounting and adjustment options of the steering wheel, allowing for flexible customization for testing purposes. A custom-manufactured connector securely fastens the steering column to the input shaft of the servomotor, ensuring reliable operation.

Additionally, custom-developed steering system software enables the parameterization of maximum angles through stop-limitations, the conversion of angles and torques in the output, and further processing within the HMI, a topic that will be detailed in future publications.

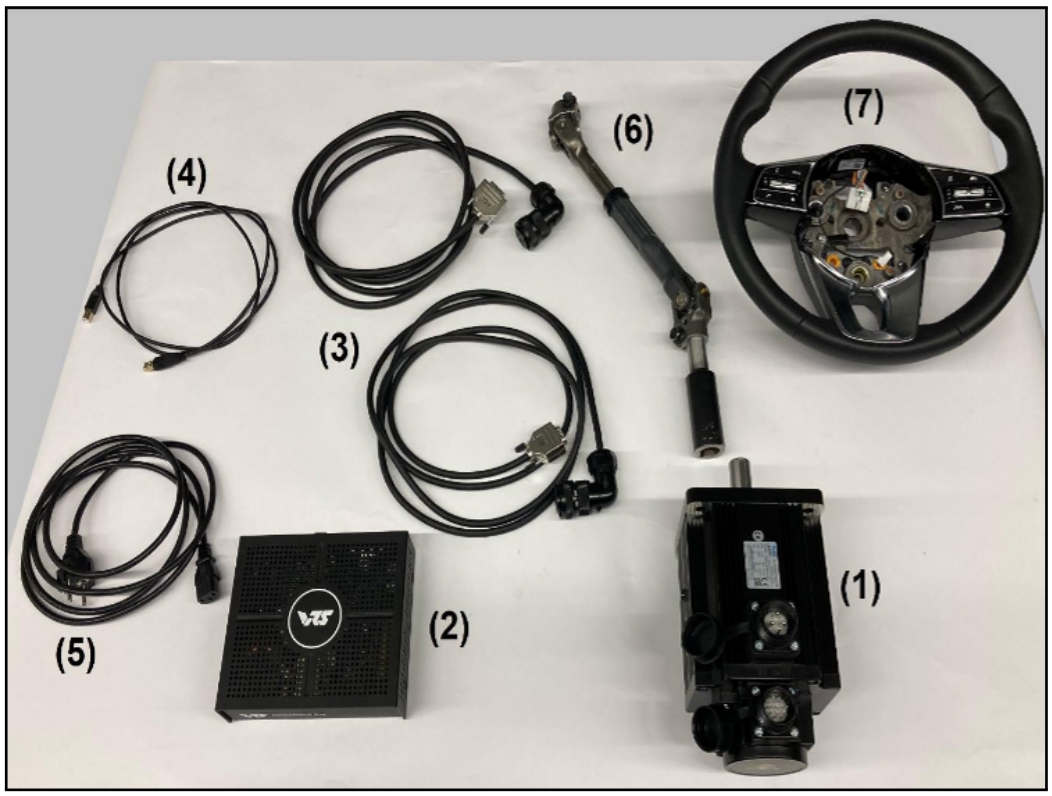


Figure 10. Overview of steering system components. (1): encoder, (2): controller, (3): encoder cables, (4): USB connection cable, (5): power supply cable, (6): vehicle steering column, (7): vehicle steering wheel.

4.3.4. Gear-Shifter

Figure 11 illustrates the operation method of the gear shifter. To ensure the usability of the gear-shifter lever (1), a push button switch (2) and a linear potentiometer (3) are integrated into its mechanical assembly. These components generate the analog voltage signals U_{S1} and U_{P1} that are connected to the analog inputs of an Arduino microcontroller. The microcontroller code processes these signals and their intervals, interpreting them to identify the drive step selected by the lever. The derived positions are outputted as HID joystick signals, providing an effective method for seamless integration with various simulation software.

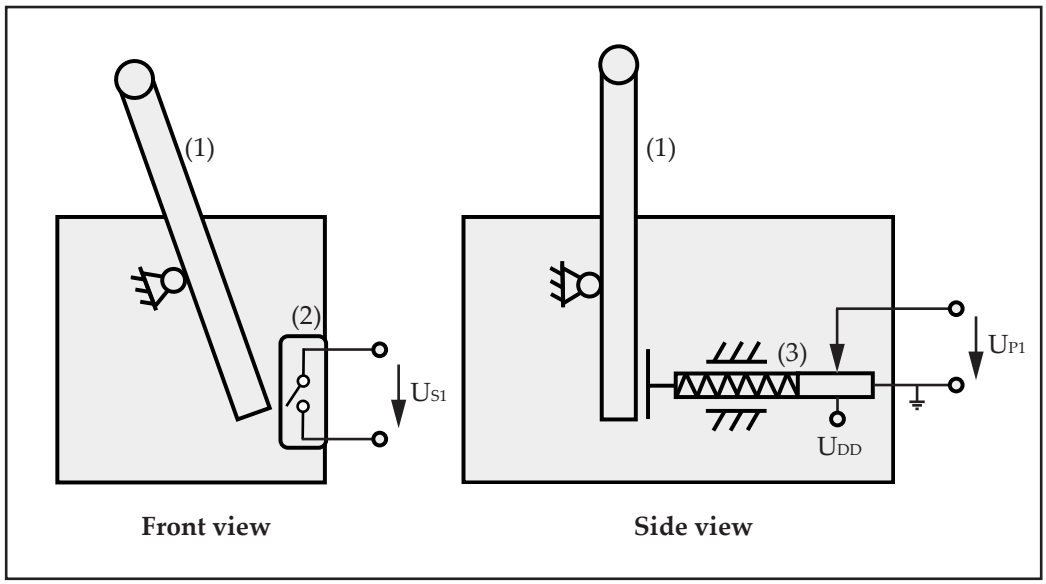


Figure 11. Schematic: modification of the original gear-shifter. (1): gear-shifter lever, (2): push button switch, (3): linear potentiometer.

4.3.5. Cockpit-Dashboard

The factory-installed digital dashboard was replaced with a more advanced display featuring enhanced video output capabilities. The implemented display shown in Figure 12 receives its video signal from a Raspberry Pi, which can output both images and animations of the speedometer, along with additional user interface features. Power is provided via a USB connection linked to a DC/DC converter that supplies an output voltage of 5 V. The video signal is transmitted through a HDMI cable.



Figure 12. Integration of the cockpit display into the driver's cabin.

4.3.6. Measurement Onboard Computer

During the integration of a measurement unit within the driving simulator, multiple hardware solutions were evaluated to identify the most suitable option. Initially, three measurement systems were considered: measurement cards, micro-computers (e.g., Raspberry Pi, Arduino), and rugged computers.

Measurement cards offer precise signal acquisition and cost-effective integration into existing computer systems. However, they are challenged by long and interference-prone cable connections, limited expandability, and inadequate capabilities for handling control tasks within the simulator.

Micro-computers, such as Raspberry Pi and Arduino, provide lightweight, compact, and economical solutions suitable for specific applications. Nevertheless, their limited

computing power, memory capacity, and connectivity restrict their ability to simultaneously manage complex control and measurement tasks.

In contrast, rugged computers meet all the necessary requirements, including automotive grade interfaces, data acquisition, signal processing, control, and live monitoring. Their integration within the driver's cabin minimizes cable runs and reduces mechanical stress. While they are notably heavier and require specific measures for safe installation, their advantages make them indispensable for advanced simulation environments.

Specifically, the OnLogic Karbon 804 series rugged computer was selected for its robust performance in managing data generated by the HMI and various sensors. It serves as a crucial interface with the simulator's main computing systems and is capable of logging and transferring data from simulations, the HMI, and sensors such as cameras, LiDARs, and bio-signals like EEG and ECG. Additionally, it provides CAN interfaces, facilitating the integration of control units into the simulator's bus system for Hardware-in-the-Loop (HiL) testing. The chosen setup closely emulates real vehicle testing conditions, where rugged computers are standard equipment.

4.4. Visual Environment Design

In the field of visual environments for simulation purposes, two primary types emerge: screens and virtual reality (VR) goggles. VR goggles require minimal hardware and offer seamless integration. While most VR goggles only provide a field of view (FOV) around 95° with exceptions of up to 180° [38], though head tracking technology in VR goggles allows for a field of observation of 360° depending on the simulation. Additionally, they provide a cost-effective hardware solution [39]. Screens, including projector screens, present advantages such as less long-term eye strain for the subject compared to VR goggles [40]. They are non-invasive, allowing drivers to maintain a free head and face, thereby enhancing monitoring capabilities through driver monitoring systems utilizing cameras and LiDAR sensors.

In selecting the ideal visual environment, a screen and projector solution was chosen. This approach closely resembles the naturalistic driving experience, providing a more immersive and realistic setting for the simulation. This setup also allows for more straightforward comparisons to real driving conditions, ensuring that insights drawn from simulations are applicable and relevant. Furthermore, the projector and screen setup allows for the easy integration of biometric sensors, such as eye tracking, facilitating enhanced data collection and analysis. While this initial setup provides a robust framework for simulation, VR goggles can still be incorporated later if required for a specific study.

To determine the optimal projector type and positioning, a CAD model was utilized. Considerations beyond the obvious specifications, such as maximum frame rate and resolution, include the throw ratio, field of view within the driver's cabin, platform motion (which influences visual immersion), the covering of projected images by driving simulator components, and spatial efficiency. These dependencies are illustrated in Figure 13. Developing the visual screen involves selecting projectors and determining screen dimensions. To represent the best compromise along these factors, the chosen visual environment features a projector screen with a 220° FOV, a diameter of 5 m, and a height of 3 m, accompanied by three projectors.

The virtual design process commenced with the construction of the simulator model, incorporating the dimensions and dynamics of both the motion platform and the driver's cabin. Regarding the projector type and configuration, various projector types and their characteristics, such as length, aspect ratio, vertical offset and the throw distance, were considered. Possible screen dimensions were compared against each other and evaluated in the context of the projector's calculated capabilities for image size. Starting with the

image width W, it is calculated using the ratio between the projection distance d and the throw ratio R:

$$W = \frac{d}{R}. \tag{6}$$

Given the chosen projectors throw ratio of $R = 0.5$ and varying distances d, W is determined by

$$W = \frac{d}{0.5}. \tag{7}$$

The depending image height H is then calculated by using the standard aspect ratio A = 16:9

$$H = \frac{W}{A} = \frac{W \cdot 9}{16}. \tag{8}$$

The chosen projector type for this simulator is the Optoma 4K400STx (Optoma Deutschland GmbH, Mönchengladbach, Germany). Its most important aspects as well as the calculated possible screen dimensions of one single projector integrated into the system can be found in Table 2. The dimensions of the screen were also chosen in accordance with the available laboratory space. The projectors specifications were considered in the CAD model, enabling the creation of a rectangular image within the virtual environment. This image was subsequently projected onto a curved projection screen surface model, and the projectors were positioned to ensure adequate coverage of the screen. Resulting in a three-projector-cross-configuration, which allows for a space-saving enhancement of the throw distance by an additional 0.4 m, as shown in Figure 14. In this setup, the projectors are attached to the ceiling and do not move with the motion of the platform.

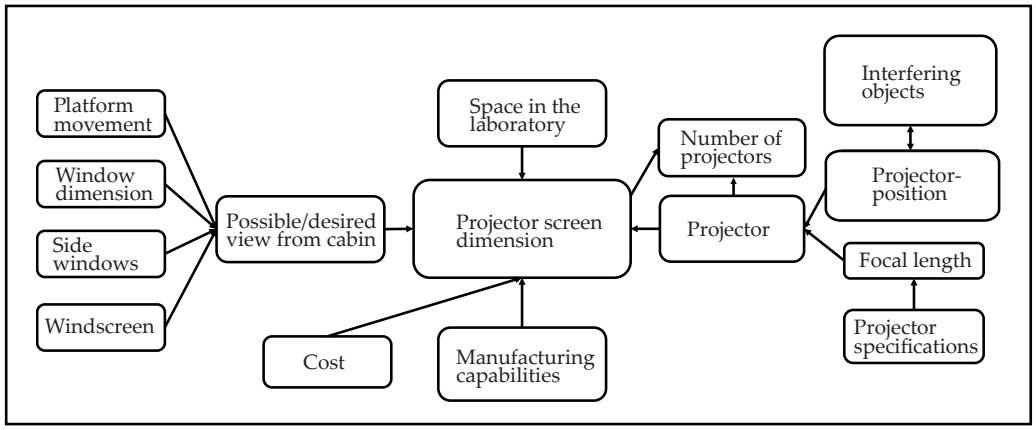


Figure 13. Considerations and dependencies of the visual environment development.

Table 2. Optoma 4K440STx projector specifications. The attributes of "effective casted image (calculated)" show the dimensions of the image a single projector can reach in the given setup.

Attribute	**Details**
Type	Optoma 4K440STx
Display/Image	
Resolution	UHD (3840 × 2160)
Contrast Ratio	1,000,000:1
Native Aspect Ratio	16:9
Horizontal Scan Rate	31–135 Khz
Vertical Scan Rate	24–120 Hz

Table 2. *Cont.*

Attribute	Details
Light Source	
Light Source Type	Lamp
Lamp Power	240 W
Brightness	4000 lm
Optical	
Throw Ratio	0.5:1
Projection Distance	0.4–3.3 m
Focal Length	7.51 mm
Lens Shift	Vertical +0%
Native Offset	116%
Effective casted projection image (calculated)	
Width	5847 mm
Height	3289 mm
Diagonal	6708 mm
Projection Distance	2900 mm
Vertical Offset	526 mm

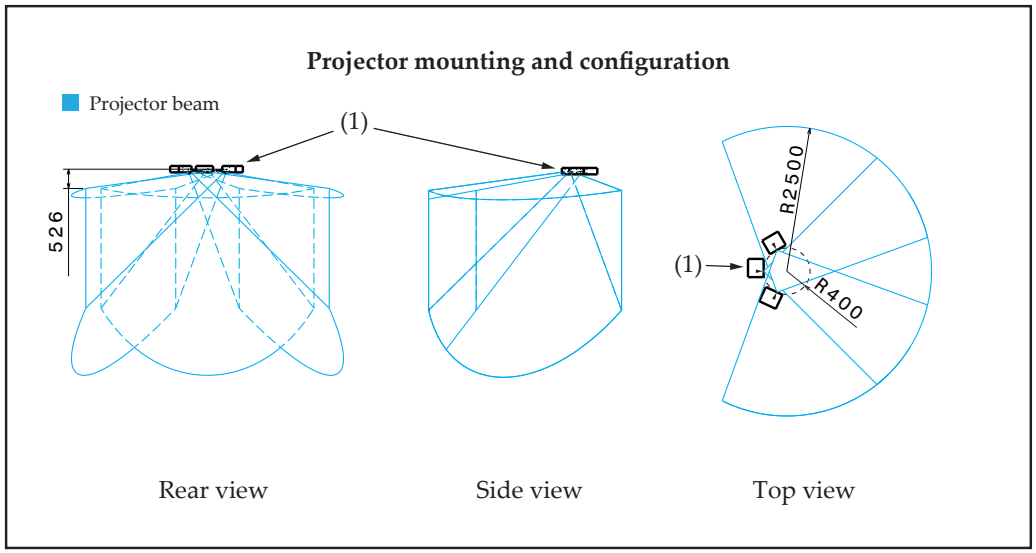


Figure 14. The graphic display illustrates the selected projector configuration. The three projectors (1) emit beams (indicated in blue) onto the projection wall. The right and left projectors are mounted in a crosswise orientation, which adds additional 0.4 m to the possible throw distance, resulting in a total throw distance of 2.9 m in total, thus enabling a larger image. The projector model incorporates a calculated vertical offset (i.e., distance between upper image border and projector lens) of 526 mm, which has been considered for the design as well.

Figure 15 illustrates the results of a CAD-based graphical analysis of the available field of view during platform motion. This analysis involves projecting the human field of view, according to [41], from the driver's seat through the car's windshield onto the screen, where the coverage of this field of view on the screen is evaluated. Evaluations are performed for various angular positions on the motion platform. The translational components are relatively minor and can therefore be omitted. Table 3 presents the measurement results, detailing the starting angle of the FOV cut-off and the percentage of maximum cut-off at

the maximum angle. The percentage of cut-off is calculated by the ratio of the area of the human FOV outside the screen boundaries with the area of the original full human FOV.

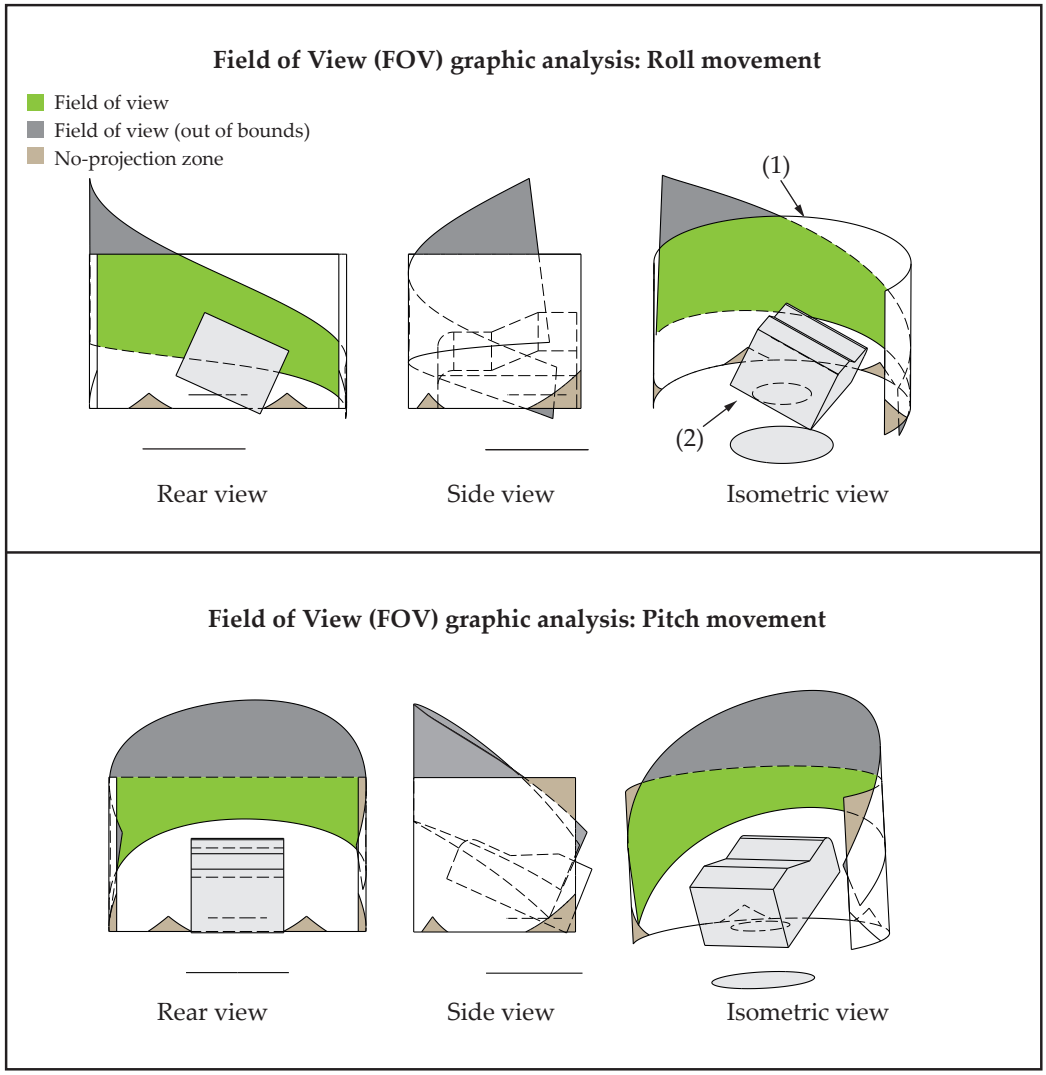


Figure 15. Graphic analysis of available field of view depending on given the angular motions roll (**upper panel**) and pitch (**lower panel**). Starting at certain angles, the human field of view (indicated in blue) extends beyond the screen boundaries (indicated in gray), potentially causing disruptions in immersion for the driver. (1): projector screen, (2): vehicle cabin and motion platform.

Table 3. Field of view (FOV) loss analysis.

Motion	Start Angle FOV-Loss [°]	Max. Angle Motion Platform [°]	Max. FOV-Loss [m^2]	Eff. FOV Surface [m^2]	Rel. FOV-Loss [%]
Roll	11.2	24.5	2.3	21.15	11
Pitch	8.1	23.6	7.47		35
Yaw	12.2	37	1.81		9

A FOV cut-off by 35% at maximum pitch motion through the platform is quite extensive, but had to be accepted regarding other limitations. Mitigation options such as extending the projector screen setup or utilizing an enhanced screen configuration (for example dome-like screens) have been considered. However, these alternatives have not been pursued due to the constraints in available space within the laboratory and extensive costs. Further system improvements have to be considered carefully, since it needs further

investigation of the level of immersion break caused by the FOV cut-off. A systematic review on the circumstances—both quantity and duration—under which these platform limits are reached in ADAS and AD scenarios is to be conducted.

Additionally, the graphical analysis depicted in Figure 16 employs the CAD model to examine the projections and potential shadow casting. The integration of the projector beams into the CAD model facilitates the analysis of potential obstructions, such as the driver's cabin, which could obscure the projector beams and cast shadows on the screen. The analysis reveals that while shadows are indeed cast due to the intersection of the light beam with the driver's cabin in the chosen projector arrangement, these shadows are minimal and lie outside the driver's FOV.

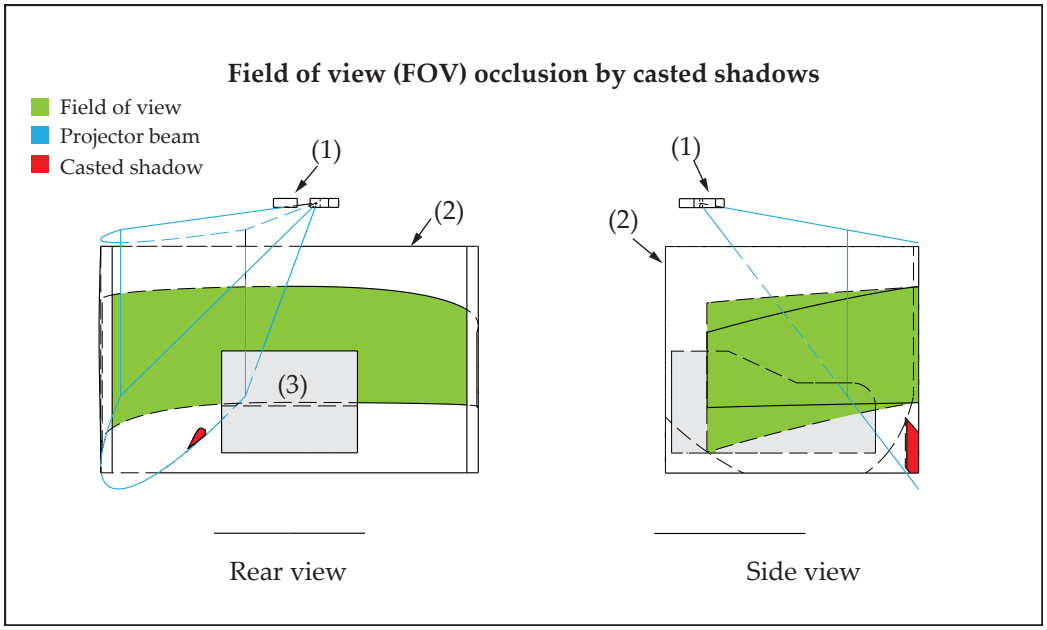


Figure 16. The graphical analysis of available field of view (FOV) depending on shadow casting objects shows small casted shadows (indicated in red) at the bottom of the screen by the projector beam's (indicated in blue) crossing with the driver's cabin. They are considered non-critical since they are not interfering with the FOV and therefore out of sight. (1): projectors, (2): projector screen, (3): vehicle cabin and motion platform.

After constructing and setting up the visual environment, the modeled visual preconsiderations were validated. Figure 17 demonstrates that the assumptions made during the development phase are accurate and that the requirements are being met. In conclusion, the FOVs are partially out of the area of the projected image, with the resulting impact for upcoming ADAS and AD research requiring further investigation. Subsequent development steps involve the transformation of the graphic image output, including curving and edge blending, which are not covered in this work.



Figure 17. Panoramic photo of cockpit view on the visual environment.

5. Testing and Validation

An ISO 3888-2 [42] double lane change driving scenario was utilized during the pre-parametrization phase, simulation stage, and ultimately for validation purposes. The double lane change was selected due to its status as a standard for vehicle dynamics testing, providing typical dynamics that are also necessary for simulating ADAS edge testing cases. This includes scenarios such as evading objects or executing emergency braking and lane change maneuvers, thus offering a solid foundation for further research.

The scenario was integrated by the simulation data interface of the DSOE using IPG CarMaker 12.0, with simulation and motion cueing data referenced in Figure A1 and Table A2. In this study, the vehicle was operated by the user and the telemetry data output during the simulation was transferred to the Acceleration Control Engine (ACE), a CWA developed by the company that created the motion platform. Regarding the system architecture shown in Figure 1 the ACE acts as the simulation data interface. To compare previous assumptions with the real test results, a simulated model of the hexapod was employed to assess the system's behavior. This model was integrated into the DSOE at the motion platform interface, enabling comprehensive testing of each component's functionality and usability, as well as evaluating how effectively the components work together within the DSOE. Incorporating functionalities from the platform control software, it calculates the needed length of each of the six linear actuators to provide the required position using inverse kinematics. The multibody dynamics and kinematic dependencies were developed using Simscape. The model is fully parametrizable and offers adjustments in platform dimension, location of masses, inertias and friction effects. For the investigations in this work, only point masses and dimensions were designed, excluding friction and inertia effects. The main purpose of the hexapod model was to provide visual information about the platform's behavior, ensuring qualitatively correct movement and monitoring the percentage of deviations from the platform's calculated minimum and maximum leg lengths during a specific scenario. The real motion platform PS-6TL-800 was also integrated into the DSOE's motion platform interface. By utilizing its internal motion sensors through the diagnostic interface, the movements of the motion platform were measured and extracted.

Figures 18–20 illustrate the dynamics resulting from the simulated maneuver. They show noticeable changes in roll and yaw dynamics and the lateral movement (sway) of the platform. Since the lane change maneuver primarily involves lateral dynamics and exhibits minimal longitudinal acceleration and vertical displacement, significant pitch, surge, and heave movements were excluded from the scope of evaluation. The simulated motion platform operates within the DSOE and successfully replicates the platform motion signals during simulation. As expected, the simulated values match the output from the ACE acting as the simulation data interface. According to the motion platform's static limits (indicated by the red dashed line), the system boundaries are respected, with no exceedances observed throughout the driving simulation. The measured platform values are qualitatively accurate, as they show a similar trend to the simulated and input signals, though they exhibit a maximum delay of ca. 27 ms. Such a delay was to be expected, as the simulated model does not account for inertias, friction, or other disturbances in the transfer function.

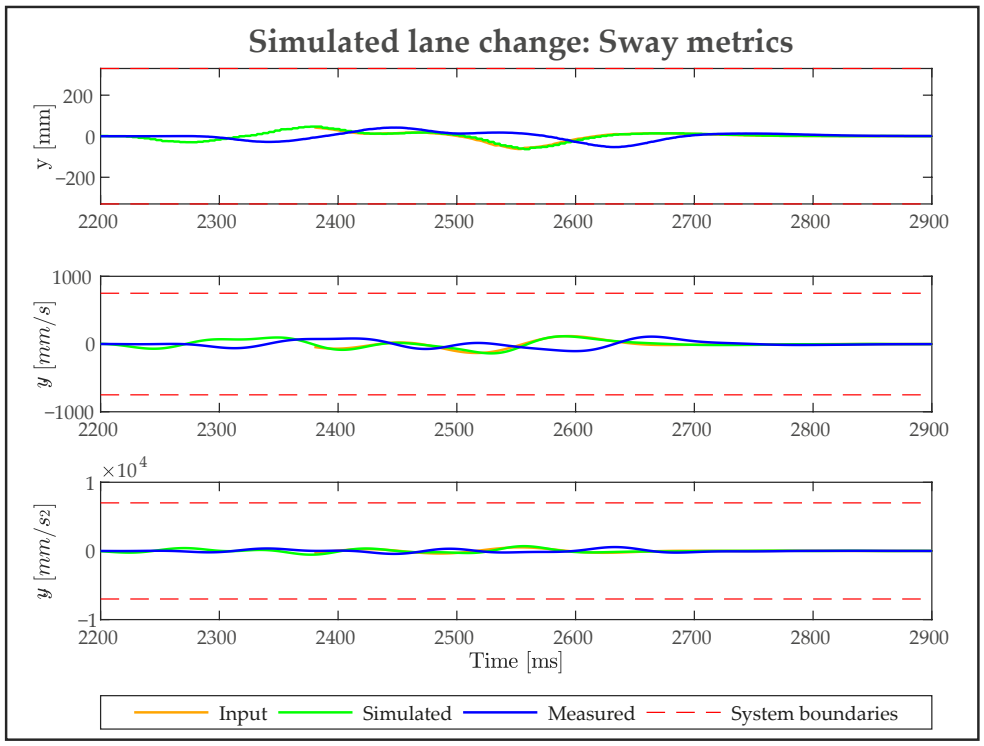


Figure 18. Sway movement results of simulated lane change maneuver. The illustrated signals are as follows. Input: Desired top-table position calculated by the motion cueing algorithm. Simulated: Platform movement by the Simscape model. Measured: Movement measured from the moition platform of the simulator.

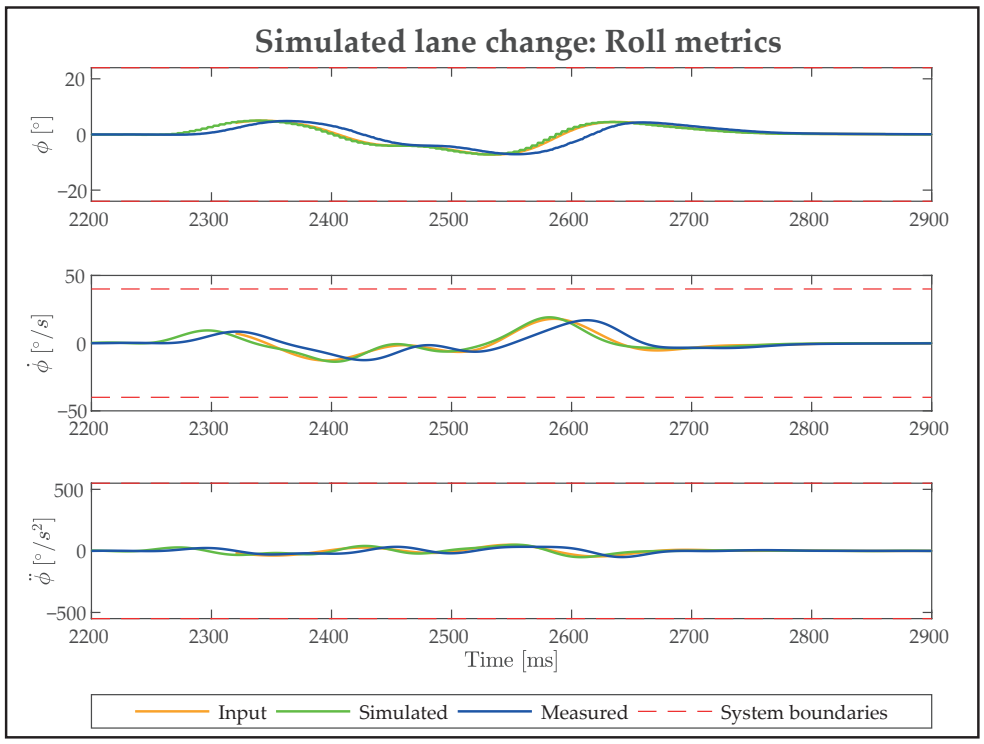


Figure 19. Roll movement results of simulated lane change maneuver. The illustrated signals are as follows. Input: Desired top-table position calculated by the motion cueing algorithm. Simulated: Platform movement by the Simscape model. Measured: Movement measured from the moition platform of the simulator.

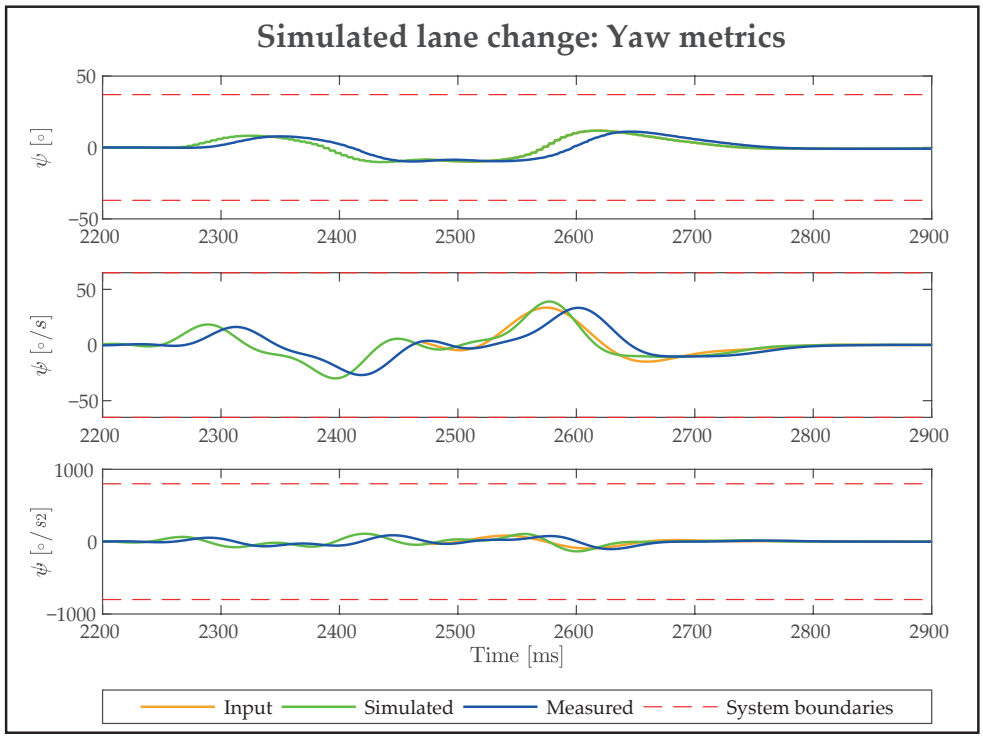


Figure 20. Yaw movement results of simulated lane change maneuver. The illustrated signals are as follows. Input: Desired top-table position calculated by the motion cueing algorithm. Simulated: Platform movement by the Simscape model. Measured: Movement measured from the moition platform of the simulator.

6. Conclusions/Discussion

In this work, a dynamic DiL-simulator was designed, constructed and tested. The study underscored the importance of meticulously defining requirements and designing a robust architecture from the outset. It demonstrated the value of hardware pre-evaluation through benchmarking, where data sheet values of motion platforms were contextualized against each other. The construction and design process was notably hardware-centric, showcasing the potential to modify and integrate real vehicle components into the DiL system. For the visual environment, a CAD-based method was presented to design it without conflicts, making it possible to measure the system quality based on the coverage degree of the projection on the screen and its alignment with the FOV. The implemented DSOE software framework, characterized by its open architectural logic, enables the interchangeability of subsystems. A simulated ISO lane change test validated the simulator's full functionality, proving the system's capability and effectiveness.

The open-source DiL simulation platform highlights its significant potential in testing vehicle systems, particularly human–machine interaction and ADAS. The integration of a 6-DoF hexapod platform combined with the CWA enables realistic driving experiences, critical for evaluating and developing advanced automated driving functions.

To enhance the effectiveness of comparing different motion platforms, more comprehensive data is essential. Current reliance on static values is insufficient. Instead, it requires insight into interdependencies among DOF and frequency. Latency considerations are also frequently overlooked, yet are critical for accurate assessment. Many crucial values can only be obtained once the platform is operational and measurements are conducted, underscoring the necessity for motion platform data sheets to be more precise, detailed, and transparent.

During development, initial motion cueing parameters have been established, yet further refinement is required to optimize platform performance. It is essential to simulate

driving scenarios tailored to specific use cases, which necessitates considering dynamic rather than static constraints. Presently, the lane change maneuver is predominantly cued through angular movements like roll and yaw, due to sensitive tilt coordination parametrization. However, lane changes provide opportunities to amplify translational acceleration cues and reduce tilt coordination.

Although previously conducted test showed a lower transport delay, the delay of 27 ms in this benchmark lies within the expected values. These earlier tests were implemented with less complex operating systems that demanded lower computational capacity, as they did not involve vehicle environment simulation or HMI signal processing. Therefore, it is reasonable to assume that the observed delays or latency issues are attributed to an overloaded computational system. The transport delay did not exceed the target of less than 40 ms. Still future refinements of the DiL system plan to distribute the computational tasks of the DSOE across multiple dedicated computers. This approach is expected to ideally reduce the delay to the levels observed in previous studies.

Overall, this study showcases the technical advancements of the simulator's architecture and its open-source approach, offering a versatile and cost-effective solution for standardized testing. This platform serves as a valuable tool for researchers and industry experts aiming to develop and test automated driving technologies through flexible and comprehensive testing methods.

7. Future Work

Future efforts will focus on expanding the HiL and Software-in-the-Loop (SiL) testing capabilities within the DiL framework. This expansion aims to fully utilize the onboard logging network, enabling more detailed and comprehensive data collection for driver monitoring system and behavior studies. In addition, we intend to document and publish the steering wheel software development process and provide access to the source code. This will facilitate further innovation and collaboration in the field of HMI research. Furthermore, we aim to enhance the adjustability of the HMI devices. For the steering wheel, this entails having full adjustability of the steering shaft and universal joints, allowing it to function as an in-the-loop test bench for power steering applications. For the gas and brake pedal unit, the introduction of a parametrizable servo motor will serve as a force feedback generator. This will facilitate the seamless integration of simulated force-travel curves and enable the simulation of vibrations akin to those produced by an ABS system or road surface irregularities.

Another key aspect of future work involves gathering more comprehensive data to enhance the precision of motion platform metrics, thereby enabling more accurate comparisons between different motion platforms. Additionally, planned studies will focus on categorizing the realism and fidelity of the DiL simulator. This effort addresses the issue of insufficient transparency and the lack of standardized parameters and categorizations in many simulator studies, which are critical for making these studies more comparable and ultimately, for drawing valid conclusions applicable to real-life scenarios.

In this context, further comparisons of different simulation software within the DSOE will be conducted to determine the most effective tools for improving our simulation capabilities. To foster transparency and collaborative advancement, all data and scripts developed during this project will be published and made freely accessible. The open-source approach will support a wider community of researchers and developers, enabling them to build upon our work and contribute to the evolving development landscape of automated driving technologies. Ultimately, we intend to enhance the existing XiL environment with additional test benches and components. These will work in a customizable network,

allowing researchers to seamlessly connect and incorporate various functions and test benches according to their specific experimental needs and preferences.

Author Contributions: Conceptualization, M.M., E.N.K. and B.I.; methodology, M.M., E.N.K. and B.I.; software, M.M. and B.I.; validation, M.M., investigation, M.M., formal analysis, M.M., E.N.K. and B.I., resources, M.M., E.N.K. and B.I.; data curation M.M.; writing—original draft preparation, M.M., E.N.K. and B.I.; writing—review and editing, M.M., E.N.K. and B.I.; visualization, M.M.; supervision, E.N.K.; project administration, E.N.K.; funding acquisition, E.N.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partly funded by Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen grant number 005-2211-0047.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article are available at https://git-ce.rwth-aachen.de/asil-lab-cologne-open/publications/devdil/supplementary_material (accessed on 20 August 2025).

Acknowledgments: This work was supported by Hyundai Motor Europe through the donation of a Kia XCEED 2022 vehicle, integrated into the Driver-in-the-Loop simulator, and through their collaboration as an industry partner.

Conflicts of Interest: The authors declare no conflicts of interest. The sponsors had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

ACE	Acceleration control engine
AD	Automated driving
ADAS	Advanced driver assistance system
CWA	Classical washout algorithm
DiL	Driver-in-the-Loop
DoF	Degree of freedom
DSOE	Driving simulator operating environment
ECG	Electrocardiogram
EEG	Electroencephalography
FOV	Field of view
HiL	Hardware-in-the-Loop
HMI	Human–machine interface
MCA	Motion cueing algorithm
SiL	Software-in-the-Loop
VR	Virtual reality
XiL	X-in-the-Loop

Appendix A

Appendix A.1

Table A1. Motion-platform benchmark.

Motion-Platform Benchmark							
Company	**Motion Systems (Poland)**			**Motion for Simulators (Belgium)**			**GFORCE Factory (Netherlands)**
Type	PS-6TL-350	PS-6TL-800	PS-6TL-1500	MFS-6DOF-1	MFS-6DOF-2	MFS-6DOF-3	EDGE 6D MOTION SIMULATOR
Source	[43]	[44]	[45]	[46]	[46]	[46]	[47]
				Excursion			
Surge [m]	0.3	0.37	0.36	0.15	0.15	0.217	0.13
Sway [m]	0.26	0.33	0.32	0.15	0.15	0.185	0.1
Heave [m]	0.187	0.22	0.27	0.175	0.175	0.18	0.07
Roll [°]	20.6	24.5	21.3	21	21	22	15
Pitch [°]	19.9	23.6	20.4	25	25	25	15
Yaw [°]	27.8	37	24.9	20	20	20	15
				Velocity			
Surge [m/s]	0.68	0.68	0.7	0.32	0.32	0.6	0.5
Sway [m/s]	0.7	0.75	0.7	0.32	0.32	0.8	0.5
Heave [m/s]	0.37	0.40	0.4	0.32	0.32	0.6	0.3
Roll [°/s]	45	41	35	40	40	40	35
Pitch [°/s]	50	46	35	40	40	40	35
Yaw [°/s]	60	67	35	40	40	40	40
				Acceleration			
Surge [m/s^2]	6.4	7	7	6.87	6.87	4.9	6
Sway [m/s^2]	5	7.1	7	6.87	6.87	4.9	6
Heave [m/s^2]	5	5	5	9.81	9.81	7.85	5
Roll [°/s^2]	600	560	250	500	500	700	600
Pitch [°/s^2]	650	620	250	500	500	900	600
Yaw [°/s^2]	700	800	250	500	500	550	600

Table A1. *Cont.*

Simulator-Benchmark				
Company	**Shenzen Uni Technology (China)**	**Suzhou Fondari Automation Equipment (China)**	**Brunner Innovation (Switzerland)**	**DOF Interact (Thailand)**
Type	Custom 6DOF Platform	Hexapod 6-Dof Electric Motion Platform	6DOF MOTION 1000	6DOF Motion Platform
Source	[48]	[49]	[50]	[51]
		Excursion		
Surge [m]	0.3	0.5	0.14	0.15
Sway [m]	0.3	0.5	0.14	0.15
Heave [m]	0.3	0.5	0.125	0.15
Roll [°]	35	35	15	15
Pitch [°]	35	35	15	15
Yaw [°]	35	35	16	19
		Velocity		
Surge [m/s]	0.5	1	0.25	0.25
Sway [m/s]	0.5	1	0.28	0.28
Heave [m/s]	0.5	1	0.28	0.28
Roll [°/s]	35	60	30	30
Pitch [°/s]	35	60	30	30
Yaw [°/s]	35	60	30	30
		Acceleration		
Surge [m/s^2]	19.61	9.81	3	16.67
Sway [m/s^2]	19.62	9.81	3	15.69
Heave [m/s^2]	19.62	9.81	4	19.62
Roll [°/s^2]	300	200	250	50
Pitch [°/s^2]	300	200	250	50
Yaw [°/s^2]	300	200	250	50

Table A1. *Cont.*

Simulator-Benchmark					
Company		**Servos & Simulation Inc. (USA)**		**VANHALTEREN (Netherlands)**	**DOF Reality (USA)**
Type	710-6-1000-220	710-6-2000-220	710-6-4500-220	eMotion-1500	Consumer P6
Source	[52]	[52]	[52]	[53]	[54]
			Excursion		
Surge [m]	n.a.	n.a.	n.a.	0.63	n.a.
Sway [m]	n.a.	n.a.	n.a.	0.5	n.a.
Heave [m]	0.1	0.1	0.2	0.38	n.a.
Roll [°]	32	32	32	24	12.5
Pitch [°]	32	32	32	28	12.5
Yaw [°]	32	32	32	27	12.5
			Velocity		
Surge [m/s]	0.25	0.25	0.25	0.79	n.a.
Sway [m/s]	0.28	0.28	0.28	0.81	n.a.
Heave [m/s]	0.28	0.28	0.28	0.55	n.a.
Roll [°/s]	100	100	100	n.a.	105
Pitch [°/s]	100	100	100	n.a.	105
Yaw [°/s]	100	100	100	n.a.	105
			Acceleration		
Surge [m/s^2]	7.3575	7.3575	7.3575	6.867	4.9
Sway [m/s^2]	7.3575	7.3575	7.3575	6.867	4.9
Heave [m/s^2]	7.3575	7.3575	7.3575	9.81	4.9
Roll [°/s^2]	100	100	100	600	n.a.
Pitch [°/s^2]	100	100	100	600	n.a.
Yaw [°/s^2]	100	100	100	900	n.a.

Appendix A.2

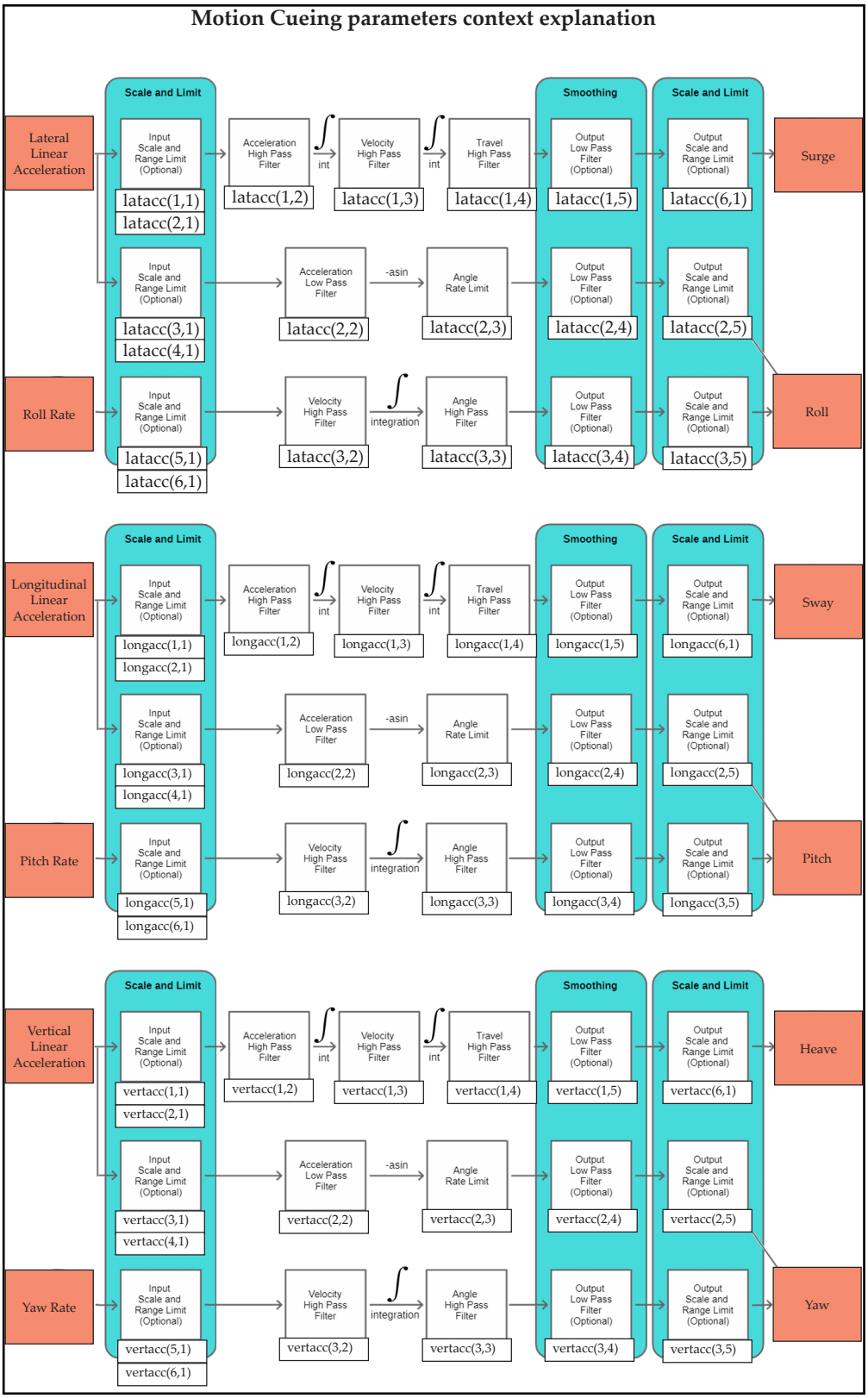


Figure A1. An overview of motion cueing parameters including their names.

Table A2. Documentation of the ISO lane change simulation.

Test Documentation DiL Testbench		
Date (DD/MM/YYYY):	10/03/2025	
Test number:	14	
Test iteration:	2	
Test name:	IPG CarMaker: Lane-change demonstration	
Test aim:	Demonstrate the function of the driving simulator. Extract simulator movement data for post-processing.	
Test description:	S. ISO 3888-2 [42]. The vehicle is controlled using the HMI. Lane passing velocity = 80 km/h.	
Test result:	System works as expected. For data evaluation see Section 5.	
Vehicle- Environment Simulation		
Type	Description	
Simulation-Software	IPG CarMaker 12.0	
Vehicle dataset	Demo_Kia_EV6	
Scenario dataset	LaneChange_ISO3888-2.rd5	
Environment dataset	Day, clear, no wind, dry, 24 °C	
Software framework		
Type	Name	Description
Main operating framework	DSOE	Driving Simulator Operation System
Simulation data interface	FSPM Simulink	Motion Systems ForceSeat PM Simulink Plugin
Simulation data interface profile	FSMI_CM_ACE	6-DoF classical washout algorithm of manufacturer

Table A2. *Cont.*

Motion cueing parameters		
Lateral acceleration and roll rate		
Variable	Value	Description
Tilt-Coordination	1	Flag for Tilt-Coordination enable/disable
latacc(1,1)	3	Input limitation Factor of the sigmoid function that scales and saturates the input
latacc(2,1)	20	Input limitation: maximum of absolute input value
latacc(3,1)	0.3	TC Input limitation Factor of the sigmoid function that scales and saturates the input
latacc(4,1)	10	TC Input limitation: maximum of absolute input value
latacc(5,1)	2.2	Roll Input limitation maximum value
latacc(6,1)	300	Roll Input limitation maximum value
latacc(1,2)	8	Acceleration filter Highpass cutoff frequency [Hz]
latacc(2,2)	2	Acceleration filter Lowpass cutoff frequency [Hz]
latacc(3,2)	3	Angular velocity filter Highpass cutoff frequency [Hz]
latacc(1,3)	0.6	Velocity filter Highpass cutoff frequency [Hz]
latacc(2,3)	700	Angle rate limit maximum rate [deg/s]
latacc(3,3)	0.5	Angle filter Highpass [Hz]
latacc(1,4)	0.4	Travel filter Highpass cutoff frequency [Hz]
latacc(2,4)	2	Angle output Lowpass 1 cutoff frequency [Hz]
latacc(3,4)	3	Angle output Lowpass 2 cutoff frequency [Hz]
latacc(1,5)	25	Travel Output filter Lowpass cutoff frequency [Hz]
latacc(2,5)	−0.75	Angle output 1 factor: slope of linear function that scales the input
latacc(3,5)	−1	Angle output 2 factor: slope of linear function that scales the input
latacc(1,6)	−1.5	Travel Output Limit

Table A2. *Cont.*

Longitudinal acceleration and pitch rate		
Variable	Value	Description
Tilt-Coordination	1	Flag for Tilt-Coordination enable/disable
longacc(1,1)	0.5	Input limitation Factor of the sigmoid function that scales and saturates the input
longacc(2,1)	20	Input limitation: maximum of absolute input value
longacc(3,1)	0.1	TC Input limitation Factor of the sigmoid function that scales and saturates the input
longacc(4,1)	10	TC Input limitation: maximum of absolute input value
longacc(5,1)	2	Roll Input limitation Factor
longacc(6,1)	300	Roll Input limitation maximum value
longacc(1,2)	0.4	Acceleration filter Highpass cutoff frequency [Hz]
longacc(2,2)	1.5	Acceleration filter Lowpass cutoff frequency [Hz]
longacc(3,2)	3	Angular velocity filter Highpass cutoff frequency [Hz]
longacc(1,3)	0.2	Velocity filter Highpass cutoff frequency [Hz]
longacc(2,3)	700	Angle rate limit maximum rate [deg/s]
longacc(3,3)	0.5	Angle filter Highpass cutoff frequency [Hz]
longacc(1,4)	0.65	Travel filter Highpass cutoff frequency [Hz]
longacc(2,4)	2	Angle output Lowpass 1 cutoff frequency [Hz]
longacc(3,4)	3	Angle output Lowpass 2 cutoff frequency [Hz]
longacc(1,5)	3	Travel Output filter Lowpass cutoff frequency [Hz]
longacc(2,5)	−1	Angle output 1 factor
longacc(3,5)	−1	Angle output 2 factor
longacc(1,6)	1.1	Travel Output Limit

Table A2. *Cont.*

Vertical acceleration and yaw rate		
Variable	Value	Description
Tilt-Coordination	0	
vertacc(1,1)	12	Input limitation Factor of the sigmoid function that scales and saturates the input
vertacc(2,1)	2.5	Input limitation: maximum of absolute input value
vertacc(3,1)	0.5	TC Input limitation Factor of the sigmoid function that scales and saturates the input
vertacc(4,1)	9.81	TC Input limitation: maximum of absolute input value
vertacc(5,1)	6	Roll Input limitation Factor
vertacc(6,1)	3000	Roll Input limitation maximum value
vertacc(1,2)	7	Acceleration filter Highpass cutoff frequency [Hz]
vertacc(2,2)	0	Acceleration filter Lowpass cutoff frequency [Hz]
vertacc(3,2)	4	Angular velocity filter Highpass cutoff frequency [Hz]
vertacc(1,3)	3	
vertacc(2,3)	0	Angle rate limit maximum rate [deg/s]
vertacc(3,3)	0.05	Angle filter Highpass cutoff frequency [Hz]
vertacc(1,4)	1	Travel filter Highpass cutoff frequency [Hz]
vertacc(2,4)	0	Angle output Lowpass 1 cutoff frequency [Hz]
vertacc(3,4)	1	Angle output Lowpass 2 cutoff frequency [Hz]
vertacc(1,5)	7	Travel Output filter Lowpass cutoff frequency [Hz]
vertacc(2,5)	0	Angle output 1 factor: slope of linear function that scales the input
vertacc(3,5)	1.3	Angle output 2 factor: slope of linear function that scales the input
vertacc(1,6)	−0.7	Travel Output Limit

References

1. Perrelli, M.; Cosco, F.; Polito, D.; Mundo, D. Development and Validation of a Vehicle Simulation Platform for Driver-in-the-Loop Testing. In *Advances in Italian Mechanism Science*; Springer International Publishing: Cham, Switzerland, 2022; pp. 355–360. [CrossRef]
2. Hendrik, A.; Schmid, C. Driver-in-the-loop Simulation for Advanced Driver Assistance Systems. In Proceedings of the 12th International Symposium on Advanced Vehicle Control, AVEC 2014, Tokyo, Japan, 22–26 September 2014.
3. Yu, C.; Jung, J.; Lee, H.; Gwon, Y.; Lee, H. A Study on the Establishment of VILS Based on Simulation for ADAS Inspection. *Trans. Korean Soc. Automot. Eng.* **2022**, *30*, 873–879. [CrossRef]
4. Schwarz, C.; Ahmad, O.; Brown, T.; Gaspar, J.; Wagner, G.; McGehee, D.V. The Long and Winding Road: 25 Years of the National Advanced Driving Simulator. *IEEE Comput. Graph. Appl.* **2023**, *43*, 121–128. [CrossRef] [PubMed]
5. Swanson, K.S.; Brown, A.A.; Brennan, S.N.; LaJambe, C.M. Extending Driving Simulator Capabilities Toward Hardware-in-the-Loop Testbeds and Remote Vehicle Interfaces. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops), Gold Coast City, Australia, 23 June 2013; pp. 115–120. [CrossRef]
6. Sekar, R.; Jacome, O.; Chrstos, J.; Stockar, S. Assessment of Driving Simulators for Use in Longitudinal Vehicle Dynamics Evaluation. *SAE Tech. Pap.* **2022**, *2022*, 1–10. [CrossRef]
7. Campos, J.L.; Bédard, M.; Classen, S.; Delparte, J.J.; Hebert, D.A.; Hyde, N.; Law, G.; Naglie, G.; Yung, S. Guiding Framework for Driver Assessment Using Driving Simulators. *Front. Psychol.* **2017**, *8*, 1428–1431. [CrossRef] [PubMed]
8. Quan, C.Y.; Mansor, S.; Jian, C.J.; Rahman, M.M.; Karim, H.A.; Weng, B.K. Modelling and Evaluation of Driving Simulator for Driving Education in Malaysia. *J. Logist. Inform. Serv. Sci.* **2023**, *10*, 211–220. [CrossRef]
9. VI-grade GmbH. VI-Grade Simulators. Available online: https://www.vi-grade.com/en/solutions/driving_simulators/ (accessed on 13 August 2025).
10. Dynisma Ltd. Dynisma Simulator. Available online: https://www.dynisma.com/dmg-1-automotive (accessed on 13 August 2025).
11. Moog Inc. Moog Simulators. Available online: https://www.moog.de/maerkte/simulation/Driver_Simulators.html (accessed on 13 August 2025).
12. TU Dresden. Dresden Driving Simulator. Available online: https://tu-dresden.de/bu/verkehr/kft/die-professur/ausstattung/fahrsimulator-1?set_language=en (accessed on 13 August 2025).
13. IKA RWTH Aachen. Highly Dynamic Driving Simulator. Available online: https://www.ika.rwth-aachen.de/en/competences/equipment/simulator-center/highly-dynamic-driving-simulator.html (accessed on 13 August 2025).
14. University of Leeds. Car Simulator (UoLDS). Available online: https://uolds.leeds.ac.uk/facility/driving-simulator/ (accessed on 13 August 2025).
15. Michalík, D.; Jirgl, M.; Arm, J.; Fiedler, P. Developing an Unreal Engine 4-Based Vehicle Driving Simulator Applicable in Driver Behavior Analysis—A Technical Perspective. *Safety* **2021**, *7*, 25. [CrossRef]
16. Čulík, K.; Kalašová, A.; Štefancová, V. Evaluation of Driver's Reaction Time Measured in Driving Simulator. *Sensors* **2022**, *22*, 3542. [CrossRef] [PubMed]
17. Remonda, A.; Hansen, N.; Raji, A.; Musiu, N.; Bertogna, M.; Veas, E.; Wang, X. A Simulation Benchmark for Autonomous Racing with Large-Scale Human Data. *arXiv* **2024**. [CrossRef]
18. Capustiac, A.; Banabic, D.; Schramm, D.; Ossendoth, U. Motion Cueing: From Design Until Implementation. *Proc. Rom. Acad. Ser. A Math. Phys. Tech. Sci. Inf. Sci.* **2011**, *12*, 249–256.
19. Wynne, R.A.; Beanland, V.; Salmon, P.M. Systematic Review of Driving Simulator Validation Studies. *Saf. Sci.* **2019**, *117*, 138–151. [CrossRef]
20. Hartfiel, B.; Tomaszek-Staude, W.; Buchholz, C.; Fresemann, C.; Stark, R. Use Case-Oriented Design Recommendations for Driving Simulators for Human-Centered Protection of Safety and Comfort Functions; [Use Case Orientierte Gestaltungsempfehlungen für Fahrsimulatoren zur Menschzentrierten Absicherung von Sicherheits-und Komfortfunktionen]. *VDI Berichte* **2018**, *2018*, 655–669.
21. Miunske, T. *Ein Szenarienadaptiver Bewegungsalgorithmus für die Längsbewegung eines Vollbeweglichen Fahrsimulators*, 1st ed.; Number 55 in Wissenschaftliche Reihe Fahrzeugtechnik Universität Stuttgart; Springer Vieweg Wiesbaden: Wiesbaden, Germany, 2020; p. 5. [CrossRef]
22. Schöner, H.P.; Häcker, J.; Nagel, K. Dynamische Fahrsimulatoren. In *Handbuch Assistiertes und Automatisiertes Fahren*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2024; pp. 157–158. [CrossRef]
23. Rothermel, T. *Ein Assistenzsystem für die Sicherheitsoptimierte Längsführung von E-Fahrzeugen im Urbanen Umfeld*, 1st ed.; Number 47 in Wissenschaftliche Reihe Fahrzeugtechnik Universität Stuttgart; Springer Vieweg Wiesbaden: Wiesbaden, Germany, 2018; p. 24. [CrossRef]
24. van Doornik, J.; Brems, W.; Vries, E.D.; Wiedemann, J. Implementing Prediction Algorithms to Synchronize and Minimize Latency on a Driving Simulator. In Proceedings of the Driving Simulation Conference 2016 Europe, Paris, France, 7–9 September 2016; Driving Simulation Association: Paris, France, 2016; pp. 51–59.

25. Melzi, S.; Previati, G. Driving Simulator: Analysing the Impact of Mechanical Latency on the Perception of Lateral Dynamics. *J. Vib. Control* **2025**, *31*, 1510–1523. [CrossRef]
26. Koroglu, Y.; Wotawa, F. Towards a Review on Simulated ADAS/AD Testing. In Proceedings of the 2023 IEEE/ACM International Conference on Automation of Software Test (AST), Melbourne, Australia, 15–16 May 2023; pp. 112–122. [CrossRef]
27. Association for Standardization of Automation and Measuring Systems (ASAM). OpenScenario. Available online: https://www.asam.net/standards/detail/openscenario-xml/ (accessed on 13 August 2025).
28. CARLA Team. CARLA Simulator. Available online: https://carla.org/ (accessed on 13 August 2025).
29. Microsoft. AirSim. Available online: https://microsoft.github.io/AirSim/ (accessed on 13 August 2025).
30. OSRF. Gazebo. Available online: https://gazebosim.org/home (accessed on 13 August 2025).
31. IPG Automotive. CarMaker. Available online: https://www.ipg-automotive.com/de/produkte-loesungen/software/carmaker/ (accessed on 13 August 2025).
32. Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI). OpenDS. Available online: https://opends.dfki.de/packages/ (accessed on 13 August 2025).
33. Strachan, I. Visual and Motion Cues in the Real World and in Simulators, 2019. Updated version. Available online: https://www.raes-fsg.org.uk/directdocs/Sim%20&%20Real%20World%20Cueing%20-%20Paper_2019-11-21.pdf (accessed on 20 August 2025).
34. Brems, W. *Querdynamische Eigenschaftsbewertung in einem Fahrsimulator*, 1st ed.; Wissenschaftliche Reihe Fahrzeugtechnik Universität Stuttgart; Springer Vieweg Wiesbaden: Wiesbaden, Germany, 2018; Volume 1, pp. 51–58. [CrossRef]
35. Fischer, M. *Motion-Cueing-Algorithmen für eine Realitätsnahe Bewegungssimulation*; Technical Report; Deutsches Zentrum für Luft- und Raumfahrt e.V.: Köln, Deutschland, 2009.
36. Groen, E.L.; Bles, W. How to Use Body Tilt for the Simulation of Linear Self Motion. *J. Vestib. Res.* **2004**, *14*, 375–385. [CrossRef]
37. Reymond, G.; Kemeny, A. Motion Cueing in the Renault Driving Simulator. *Veh. Syst. Dyn.* **2000**, *34*, 249–259. [CrossRef]
38. Sauer, Y.; Sipatchin, A.; Wahl, S.; García García, M. Assessment of Consumer VR-headsets' Objective and Subjective Field of View (FoV) and Its Feasibility for Visual Field Testing. *Virtual Real.* **2022**, *26*, 1089–1101. [CrossRef]
39. Filio, D.; Ziraldo, E.; Dony, L.; Gonzalez, D.; Oliver, M. Comparison between Wrap Around Screens and a Head Mounted Display on Driver Muscle and Kinematic Responses to a Pedestrian Hazard. *Appl. Ergon.* **2023**, *106*, 103878. [CrossRef] [PubMed]
40. Blissing, B.; Bruzelius, F.; Eriksson, O. The Effects on Driving Behavior When Using a Head-mounted Display in a Dynamic Driving Simulator. *ACM Trans. Appl. Percept.* **2022**, *19*, 1–18. [CrossRef]
41. Spector, R.H. Visual Fields. In *Clinical Methods: The History, Physical, and Laboratory Examinations*, 3rd ed.; Walker, H.K., Hall, W.D., Hurst, J.W., Eds.; Butterworths: Boston, MA, USA, 1990; Chapter 116.
42. ISO/DIS3888-2:2025; Passenger cars - Test track for a severe lane-change manoeuvre - Part 2: Obstacle avoidance. International Organisation for Standardization: Geneva, Switzerland, 2025.
43. Motion Systems. PS-6TL-350. 2023. Available online: https://motionsystems.eu/product/motion-platforms/ps-6tl-350/ (accessed on 21 October 2023).
44. Motion Systems. PS-6TL-800. 2023. Available online: https://motionsystems.eu/product/motion-platforms/ps-6tl-800/ (accessed on 21 October 2023).
45. Motion Systems. PS-6TL-1500. 2023. Available online: https://motionsystems.eu/product/motion-platforms/ps-6tl-1500/ (accessed on 21 October 2023).
46. MOTION FOR SIMULATORS S.R.L. 6 DOF Motion Platforms Provide it All: Pitch, Roll, Heave, Sway, Surge and Yaw. 2023. Available online: https://motionforsimulators.com/6dofmotionplatforms/ (accessed on 22 October 2023).
47. Gforcefactory B.V. About the EDGE 6D Motion Simulator. 2023. Available online: https://www.gforcefactory.com/edge-6d (accessed on 20 October 2023).
48. Shenzhen UNI Technology Co., Ltd. Custom 6DOF Motion Platform AASD15A Drive. 2023. Available online: https://unite-ch.net/product/diy-6dof-motion-platform-amc-aasd15a-controller (accessed on 20 October 2023).
49. Suzhou Fengda Automation Equipment Technology Co, Ltd. Hexapod 6-DOF Electric Motion Platform Stewart Platform Flight Simulator 6-Axis Motion Simulating. 2023. Available online: https://szfdra.en.made-in-china.com/product/pZATInqbIQai/China-Hexapod-6-Dof-Electric-Motion-Platform-Stewart-Platform-Flight-Simulator-6-Axis-Motion-Simulating.html (accessed on 21 October 2023).
50. BRUNNER ELEKTRONIK AG. 6DOF Motion 1000. 2023. Available online: https://brunner-innovation.swiss/product/6dof_motion_1000/ (accessed on 20 October 2023).
51. DOF Interact. 6 DOF Motion Platform. 2023. Available online: https://www.dofinteract.com/ (accessed on 21 October 2023).
52. Servos and Simulation, Inc. Six-Axis (6DOF) Motion Base Platform Systems 710-6. 2023. Available online: https://servosandsimulation.com/motion-base-platforms-and-control-loader-products/six-axis-6dof-motion-base-platform-systems/ (accessed on 20 October 2023).

53. Van Halteren Technologies. eMotion-1500. 2023. Available online: https://vanhalteren.com/wp-content/uploads/2025/01/DataSheet_VHT-eMotion1500_42024.pdf (accessed on 22 October 2023).
54. DOF Reality Motion Simulators. Professional Motion Simulator Platform 6-AXIS DOF. 2023. Available online: https://dofreality.com/product/racing/motion-racing-rig-6-axis-pro-p6/?srsltid=AfmBOor7fkPCRhZHFp-iA99iC7r9DrqbaQu3pz7_9YMuHpGZHci0v8lm (accessed on 20 October 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.