Politecnico di Torino

Master's Degree in Mechanical Engineering

Master's Thesis

A.Y. 2024/2025

Graduation Session: March 2025

Hardware-in-the-Loop for the Whole Vehicle Development Process

Supervisor:

Prof. Andrea Tonoli

Company tutor:

Giorgio Pochettino

Candidate:

Edoardo Besenval

Acknowledgments

It is my pleasure to sincerely thank all colleagues and managers at Italdesign, especially, but not only, my tutor Giorgio Pochettino. They have always been very inclusive and willing to support my growth and research throughout the internship.

My thanks also go to Professor Andrea Tonoli, for following me during the creation of this thesis project.

Last, but not least, my profound gratitude goes to all friends, classmates, and family, who have always been by my side during this challenging, yet empowering years that brought me here. Their friendship, closeness, and trust really helped me stay on course and, if I may say so, being a better and more complete person than I was.

Table of Contents

Acknowledgments 2
Table of Contents 3
Abstract 6
1. Introduction 8
1.1. Objectives 8
1.2. Motivation 8
1.3. Context 8
1.4. Personal Contribution 9
1.5. The needs in detail 10
1.5.1. Increasing Vehicle Complexity and Functionality 10
1.5.2. Challenges of Traditional Development Methods 11
1.5.3. The Need for Efficient and Reliable Testing Strategies 12
1.6. Introduction to Hardware-in-the-Loop (HIL) Testing 12
1.6.1. Definition and Basic Principles 12
1.6.2. Technical-Engineering Reasons for Implementing HIL 13
1.6.3. Project Management Reasons for Implementing HIL 15
2. Background 18
2.1. A Brief History of HIL 18
2.2. Applications of HIL in Various Industries 19
2.2.1. Aviation 20
2.2.2. Offshore and Marine 20
2.2.3. Railways 20
2.2.4. Energy 20

2.2.5. Electric drives..21
3. HIL in the Automotive Industry.......................................22
3.1. Function-Oriented Engineering (FOE) in Automotive Design..22
3.1.1. Function Definition...22
3.1.2. Trends..22
3.1.3. Transition to Function-oriented Vehicles 24
3.2. Function-oriented Vehicles Development...........24
3.2.1. HIL Implementation..24
3.2.2. Principles and Benefits of Function-oriented Engineering..25
3.3. Model-Based Design for Automotive Systems...26
3.3.1. MBD Scheme 27
3.3.2. MBD Process and Loops 27
3.4. V Model for Vehicle Development.......................31
3.4.1. Integration with MBD 31
4. Test Entities Implementation and Applications...........33
4.1. From Function to Test 33
4.1.1. Requirements..33
4.1.2. Test Catalogue 33
4.2. HIL Development 34
4.2.1. Rack Development...34
4.2.2. Software Development..35
4.2.3. Automation...35
4.3. Modeling 35
4.3.1. Environment Modeling..36
4.3.2. Models and their Use..36
4.4. Validation 38
4.5. Component HIL 39
4.5.1. Initiation..39
4.5.2. Testing 40

4.6. System HIL 42
4.6.1. Principles 42
4.6.2. Testing 42
4.6.3. System Examples 44
4.7. Multi-Bench Systems and Automation 45
4.8. System Integration Benches 45
4.8.1. Lab Car 45
4.8.2. Vehicle HIL 48
4.9. Vehicle in the Loop 53
4.9.1. VIL Test bench 53
4.9.2. The Reference Vehicle 54
4.9.3. Master Vehicle 55
5. Testing in the Development Process 57
5.1. Overview 57
5.2. Release Management 57
5.2.1. Test Plan 58
5.2.2. Function-Oriented Development Milestones 58
5.2.3. Global Releases 59
5.2.4. Roles 59
6. Virtualization Within the Overall Processes 60
6.1. Augmented Reality Design Validation 60
6.2. Mechanical Systems Engineering 62
6.3. Thermal-fluid Dynamics Optimization 62
6.4. Crash Tests 63
6.5. Structural Optimization 64
6.6. Production 65
7. Conclusion 65
Glossary 67
List of Figures 68
Bibliography 70

Abstract

Industries from many different fields are undergoing considerable technological advancement and their products are becoming every day more function-oriented and user-centered. Production processes are becoming more agile, automated, and virtualized at a pace that was never seen before.

The automotive industry is no exception, with the market pushing for always new features and strategic innovations in the field of advanced driver assistance systems (ADAS), autonomous driving (AD), electrification and increasing safety and environmental standards. As a result, original equipment manufacturers (OEMs) and suppliers face increasing pressure to follow market and regulatory demands within tight timeframes.

The development process involves intricate system engineering, requiring the integration of various subsystems, including electro-mechanical, electronic and software components. New virtualization tools are needed to cope and increase efficiency, flexibility, and comprehensiveness.

Hardware-in-the-Loop (HIL) is a powerful technique that bridges the gap between virtual simulations and physical testing. It allows us to develop, integrate, and validate electronic control units (ECUs) and integrated systems in a controlled, simulated environment and with a high level of automation.

This allows for exponentially increased comprehensiveness, reproducibility, and scalability. These benefits also come with increased safety and the ability to test systems which would be otherwise impossible to prototype.

Effective HIL implementation happens when it is included in Model-based design methods and synchronized with

the overall vehicle development process, considering V-shaped models for product development.

Being up to date with HIL and integration tests impact on business strategy is essential to get the most from their cost-effectiveness, risk reduction, and competitive advantages.

This work has been developed throughout a 6-month internship at Italdesign Giugiaro S.p.a., a design and engineering company based in Moncalieri, Piedmont, Italy.

Some of the processes and technologies hereby described are directly derived from real activities. Nonetheless, no one intends to represent the exact counterpart in detail and in its entirety.

1. Introduction

1.1. Objectives

The primary objective of this thesis is to provide a comprehensive analysis of the testing and validation processes within the automotive industry, with a specific focus on Hardware-in-the-Loop systems. This includes a detailed examination of the various test entities used, their roles, and their integration within the overall vehicle development process. The thesis aims to describe the evolution and implementation of HIL testing in the automotive industry, analyze the technical and managerial benefits of HIL systems, their implementation and the specific processes and technologies involved. The aim is to provide a comprehensive treatise to relate the technologies, the process and the practices.

1.2. Motivation

The motivation behind this thesis rises from the increasing complexity and functionality of modern vehicles, in conjunction with the tighter schedules and increasing need for flexibility. The pressure on automotive manufacturers can be partially relieved by introducing more efficient reliable and adaptable testing strategies and by increasing the virtualization level along the overall development process.

Providing an overview of the testing and validation processes is key to working effectively on the process as a whole, making the most of every solution by implementing them synergistically.

1.3. Context

The background of this thesis takes place within the automotive world, characterized by fast-growing dynamics. The specific focus is on Italdesign Giugiaro S.p.A., a renowned design and engineering company based in Moncalieri, Piedmont, Italy. Italdesign is known for its cutting-edge work in vehicle design, engineering, and production, with a strong emphasis on innovation and quality.

During my internship in the production project management department, I had the opportunity to follow the design and build of a new whole-vehicle Hardware-in-the-Loop (HIL) batch production. This project involved close collaboration with various departments within Italdesign, as well as external partners, providing a comprehensive view of the company's approach to system engineering and validation.

Italdesign's testing and validation processes ensure that all components and systems meet the highest standards of performance and safety. The company employs a range of advanced techniques, including HIL testing, augmented reality, and virtual simulations, to streamline development and reduce the need for physical prototypes. This approach not only enhances efficiency but also allows for rapid iteration and continuous improvement, key in today's market.

1.4. Personal Contribution

My personal contribution to this thesis is rooted in my active involvement in a design-build project for a batch of innovative whole-vehicle hardware-in-the-loop test benches. As junior project manager, I worked closely with various departments to gain a comprehensive understanding of the entire development and production process.

This thesis is a culmination of my efforts, providing valuable insights and recommendations for future developments.

1.5. The needs in detail

1.5.1. Increasing Vehicle Complexity and Functionality

The main driving factors for technological advancement are the following:

Safety standards and regulations are increasing, alongside the market demand for driving assistance and automation. Coherently, Advanced Driver Assistance Systems (ADAS) and Autonomous Driving (AD) technologies. The latter are rapidly evolving and require sophisticated software and the integration of various subsystems of different kinds, in order to function effectively and safely.

Secondly, regulatory bodies are constantly demanding innovative solutions to meet stricter emission regulations and environmental standards. This is making electrification a robust trend. The transition from internal combustion engines (ICE) vehicles to electric vehicles (EVs) requires the development and integration of entirely new powertrain systems and innovative power management solutions.

Finally, customers' desire for the most immersive user experience (UX) possible drives significant innovation concerning integrated functions, connected vehicles, on-board comforts, and complex infotainment, which must all be controlled by an ergonomic and intuitive user interface (UI).

These trends contribute to a significant rise in vehicle complexity and functionality, inevitably leaving past differentiators (such as powertrains) in the background in favor of the technology stack, a term that describes the layers of software and hardware that comprise a vehicle's operating system.

These trends are referred to as "ACES", for autonomous vehicles, connected cars, electrification, and shared mobility.

1.5.2. Challenges of Traditional Development Methods

The latter factors constitute several challenges for automotive manufacturers (original equipment manufacturers or OEMs), and suppliers. *.1*

1.5.2.1. Development time

New features and functionalities must be developed and integrated within increasingly tight timeframes to keep pace with market demands. If in the 60s it took about seven years for a new vehicle to go through the development process, from the first concept to the start of production (SOP), and about five in the 2000s, this time has more than halved, lately.

1.5.2.2. System integration

Having more complex vehicles also means having more systems to integrate as fast as possible. That highlights how ensuring seamless communication and coordinated operation between various subsystems becomes increasingly difficult.

1.5.2.3. Validation difficulties

All the previous aspects bring the need to highly increase the number and quality of tests to be performed to achieve validation. The issue is that traditional physical testing becomes too laborious, slow, and expensive. Moreover, it can become dangerous or impossible to test safety critical technologies and functions and some scenarios are difficult to replicate in a real-life setting.

1.5.3. The Need for Efficient and Reliable Testing Strategies

This is where Hardware-in-the-Loop (HIL) testing and validation emerges as a new paradigm to address these challenges.

HIL techniques focus on the testing and (depending on the specific case) validation of what physically constitutes functions, i.e., Electronic Control Units (ECUs), and their network, which includes sensors and actuators.

Everything is performed in a simulated and controlled environment, allowing for faster development cycles, more efficient integration processes, and more comprehensive validation compared to traditional physical testing methods.

1.6. Introduction to Hardware-in-the-Loop (HIL) Testing

1.6.1. Definition and Basic Principles

HIL is a technique used in the development and testing of complex real-time embedded systems. It is used where verification is complex, costly, and could damage expensive components, or where the real environment is dangerous or impossible to replicate.

It involves integrating real hardware components (i.e. sensors, actuators, and controllers) with a dynamic mathematical model of a system (or device) under test SUT (or DUT), often semi-automatically. The model is run at constant speed to accurately mimic the behavior of the SUT.

It's necessary to clarify which part of the system should be real or virtual, depending on what is being tested. In HIL the controller is real and possibly the environment, the sensors and the actuators are virtual. If a virtual controller is used, the evaluation is often referred to as rapid control prototyping (RCP).

Depending on the objectives of the test, on the depth required, and on the SUT knowledge, it must be defined whether HIL involves black, grey, or white box testing (B/G/WBT).

It must also be clarified whether the simulation requires a closed-loop or whether the open-loop may suffice, depending on the need to introduce feedback into the simulation.

Note that HIL does not replace physical prototypes entirely but comes alongside them. It makes the testing process leaner, whenever physical interaction can be neglected.

1.6.2. Technical-Engineering Reasons for Implementing HIL

There are several technical reasons to implement HIL testing, with benefits on different aspects of the product development cycle. *.2*

1.6.2.1. Efficiency

Test standardization: HIL eases the creation of standardized test procedures that can be applied across different projects, increasing repeatability and results consistency.

Automation: the intrinsic standardized and digital nature of HIL, allows to automate repetitive and time-consuming test procedures, assuring better results while minimizing human error. Clearly, the effects become more evident in large-scale projects with numerous operations to be conducted.

1.6.2.2. Safety

Fault-injection: HIL allows to simulate how the SUT responds to faulty conditions and failures. This capability is crucial to ensure the system's reliability and compliance and would be hardly applicable on a physical prototype.

Early-stage debugging: HIL characteristics allow engineers to identify and correct bugs and issues early in the development process before they become more complex and costly to tackle.

1.6.2.3. Flexibility

Concurrent systems engineering: HIL supports the simultaneous development and testing of different system components, accelerating the development process by assuring proper interactions at an early stage.

Integration with different tools: integration with various software and hardware tools, ensures a very high comprehensiveness.

1.6.2.4. Adaptability

Customizability: depending on their complexity, HIL systems can be customized up to the tiniest details to return the most meaningful results.

Modularity: the modular nature of HIL systems permits to test different parts of the system to be tested both together and independently. Furthermore, components and functionalities can be added or reconfigured as needed, to meet evolving project requirements or adapt to new ones.

1.6.2.5. User experience

Human factor development: ensuring usability can be difficult without physical prototypes and design faults may emerge too late. HIL may also account for this aspect early in the development process, ensuring that the system responds correctly to real human interactions.

Ergonomics: by testing the UI and the interactions in a controlled environment, HIL helps ensure that the software is user-friendly. Physical ergonomics can also be simulated in a virtual environment, leading to products that are easier and safer to use.

1.6.2.6. Regulations

Regulatory compliance: HIL benches help ensure that systems meet regulatory requirements. Sometimes they can even be approved by the reference bodies to directly certify the developed products, without the need for further prototypes.

1.6.3. Project Management Reasons for Implementing HIL

The benefits of implementing HIL are not only technical, but also managerial.

1.6.3.1. Costs

Effort: HIL automated and streamlined processes, significantly reduce the manual effort and time required. This reduces direct labor costs and allows engineers to focus on more critical tasks, speeding up the overall development cycle.

Materials: The HIL simulation environment reduces the need for physical prototypes, enabling the testing of various updates with minimal to no waste of material resources. This reduction not only cuts labor hours and costs but also minimizes waste and environmental impact.

1.6.3.2. Duration

Tight development schedules: HIL supports development and testing parallelization, following strict schedules. Rapid iterations permit issue-fixing at a high pace, preventing further delays.

Time to market: This flexible infrastructure allows us to develop different products of the same line with few adaptations and a high level of standardization. This strongly reduces the time to market, making it possible to come up with a new model as soon as needed, maintaining a competitive advantage.

1.6.3.3. Quality

Comprehensiveness: by testing a wide range of scenarios and conditions, HIL ensures that the SUT performs reliably under various circumstances, increasing the overall product's quality. Rapid prototyping is possible via quick iterations, making it possible to test even smaller changes that would not be controlled otherwise.

Stability: testing must be repeatable and consistent to ensure the delivery of a robust design. It also contributes to deeper control of the production standards across different platforms.

Fidelity: nowadays, high-fidelity simulations replicate real-world conditions accurately. This makes it possible to prove the SUT's quality even deeper than with physical tests, especially if some conditions are actually difficult to replicate in real-life.

Controlled environment: testing in a controlled environment isolates the variables that could affect the results, ensuring that the test outcomes are reliable and can be used to make informed decisions.

1.6.3.4. Risk

Safety: testing in a virtual environment helps to exclude all of the risks related to the use of an object that is not completely safe yet. Even more if the simulation concerns hazardous conditions and faulty behaviors, or if the SUT is safety-critical by itself.

Quality: comprehensive testing and validation reduces the risk of defects and recalls, saving money and protecting the company's reputation and customer satisfaction.

1.6.3.5. Management

Documentation: HIL systems often include automated logs and reports for the performed activities, simplifying the documentation process and ensuring information integrity. It is easier for project managers to track progress and make informed decisions and the documentation is standardized and readable by all stakeholders.

2. Background

2.1. A Brief History of HIL

The most primitive examples of this kind of approach come from aeronautic applications: flight simulator specifically. This is a field where high quality and safety standards are mandatory: testing a plane or a training pilot too early carries too high risks.

In 1910 simulator by Sanders Teacher, the SUT is an actual airplane, strategically mounted on the ground. The pilot is the controller, and the simulated environment is the wind coming from a well-known direction at ground level.

During the 1940s this sort of simulator is enriched by the presence of analog computers and servo hydraulic systems.

While the first digital computers were applied to aeronautical and defense HIL systems in the 1950s, it was the significant increase in computational power of 1970s computers that led to the modern concept of HIL. This new concept is applied again in driving simulators, but this time for vehicle development.

The late 1980s bring the first examples of engine-in-the-loop (EIL) with electro-actuators controlled by a computer. Another key innovation developed thanks to HIL is the anti-lock braking systems (ABS).

Anyhow, vehicle ECUs and functionalities development are mainly carried out through physical tests on prototypes. This process is time-consuming, expensive, and limited by the availability of the prototypes themselves. On-vehicle testing has several disadvantages, such as the need to build the prototypes on purpose or disassemble and reassemble an existing one, expensive and slow implementation of corrections and a limited variety of tests.

During the 1990s HIL becomes widely spread and, in the automotive world, it starts to simulate vehicle dynamics.

In the early 2000s, HIL starts to be applied to the integration of many systems at vehicle level. However, it is only in recent years that we have started to see comprehensive whole vehicle HIL, mainly for system integration purposes, following the need for more robust systems to test more and more complex networks and features.

If once the code had to be manually compiled, the ECU installed on a prototype's network and road tested for each major update, now most of the work is done in an automatized way and on-bench. Prototype vehicles are adopted when it is necessary to consider physical interaction during operation, to assess usability and as a final step for milestones validation. When faults emerge, test engineers can easily try to fix them by adjusting the parameters at the integration HIL, before referring to the function or component owner.

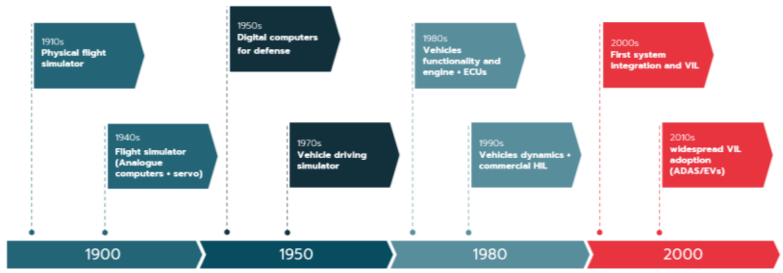

Figure 1: Main HIL innovations timeline.

2.2. Applications of HIL in Various Industries

HIL concept can fit into many different industries where testing and validation of complex mechatronic systems is required, and its strategic impact overcomes the considerable initial costs.

Here are some examples.

2.2.1. Aviation

As a first example, flight simulators, which are critical for pilot training, use HIL to provide realistic simulated environments that incorporate real-time data and responses and use actual commands in the cockpit as interface for the pilots. HIL simulation is essential to the development and testing of sophisticated control systems in aviation. HIL is used extensively in aircraft jet Full Authority Digital Engine Controllers (FADEC), whose test benches may cost around 200-500 K$, as opposed to 10 M$, when adopting prototypes. HIL has also become essential to develop fly-by-wire systems and for evaluating avionics systems' fault tolerance. In the military realm it is used for missile guidance systems, unmanned vehicles, and radar jamming.

2.2.2. Offshore and Marine

In the offshore and marine industries, besides testing huge mechatronic systems in a compact way, HIL helps in traffic management by monitoring and simulating traffic to guarantee effective and safe sailing. In addition, it makes it possible to evaluate communication and collaboration tactics for multi-vessels operations.

2.2.3. Railways

Power Hardware-in-the-Loop (PHIL) simulation is applied in the railway industry to develop complex regenerative breaking and electrification systems. At the infrastructure level, it can test how to ensure effective power distribution across the network.

2.2.4. Energy

Flexible AC transmission systems (FACTS) and high-voltage DC (HVDC) are tested using Grid Performance PHIL models. The impact of distributed energy resources (DERs), such as solar and wind power, can also be simulated with specific systems. Wind turbines and solar panels’ electronics themselves are also tested through HIL benches.

2.2.5. Electric drives

HIL simulation for electric drives and power electronics involves signal, power, and mechanical simulations, in order to guarantee the desired behavior in response to specific control parameters. HIL also helps evaluate the total energy residual (TER). Field-Programmable Gate Array (FPGA) integrated circuits can be integrated in the loop to foster accuracy. *.3*

3. HIL in the Automotive Industry

3.1. Function-Oriented Engineering (FOE) in Automotive Design

3.1.1. Function Definition

A function is a specific behavior of a vehicle designed to achieve a particular purpose which directly or indirectly changes the driver's and the road user's experience. It is made possible by one or more systems and activated by specific triggers. It has related functional requirements, such as legal, licensing, and specific regulations.

The link between different functions and subsystems to produce an extended behavior is referred to as functional network.

3.1.2. Trends

Following commercial trends, the market is asking for innovative functionalities to be provided, and vehicles complexity is constantly growing. In particular, the leading trends regard vehicle connectivity, electrification, and ADAS/AD. Furthermore, regulatory bodies responsiveness to innovation is becoming considerable. Hence, the imposed boundaries must be taken into account while implementing those innovative solutions into a new model. .4

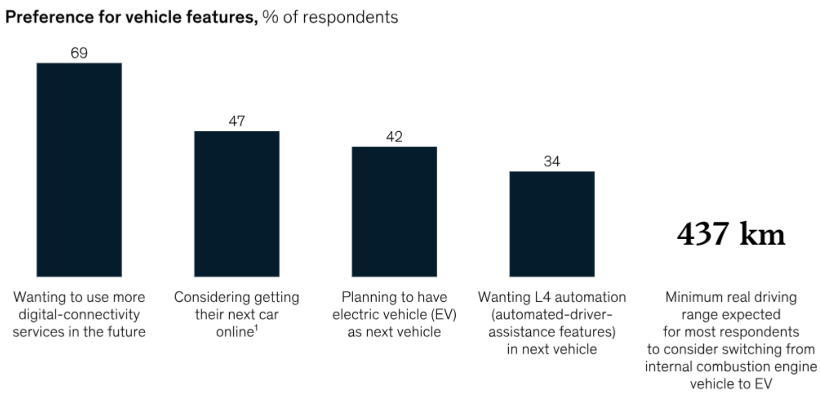

Figure 2: McKinsey consumer pulse survey, 27869 international participants, Dec 2022. .5

3.1.2.1. Connectivity

It is a fact that vehicles are always more connected. Routine over the air (OTA) updates, or even functionality upgrades are becoming standard. Smart infotainment functionalities are always requested more. Also, interconnected vehicles and smart mobility are foreseen for the near future, thanks to the advancement in internet of things (IOT) and 5G connectivity. This last aspect is fundamental for shared mobility and automated vehicles. *.6 .7*

3.1.2.2. Electrification

Driven by the market green wave and by new emission regulations, vehicles electrification is revolutionizing the industry, urging car manufacturers to come up with innovative engineering solutions and to stay up to date with the newest technologies. *.4*

3.1.2.3. Automation

Level 3 and Level 4 autonomous-driving capabilities add astounding complexity to a vehicle's network, including the need for a fully connected system, with specific sensors, actuators, and software with a strong AI base. *.8*

3.1.2.4. Automotive regulations

ISO/IEC 330xx norms, regarding Software Process Improvement and Capability Determination (SPICE), ensure that software and systems within vehicles meet OEM requirements. They cover software development, requirements elicitation, architectural design, verification, validation, and project management.

ISO 26262 norm about road vehicles functional safety aims in general to minimize the risk of accidents caused by system malfunctions in vehicles. That aspect is crucial considering ADAS and AD systems.

New regulations on cybersecurity and software updates and fast emerging laws, in particular the ones concerning

ADAS and electrification, constitute a matter of particular attention for manufacturers. *.9 .10*

3.1.3. Transition to Function-oriented Vehicles

Indeed, in the past few years we witnessed a clear transition that started from high-end vehicles and is now involving everyday city cars: the shift from conventional cars to automotive ecosystems. This transition implies a change in paradigm, where the vehicle is no more component-centric, but becomes function-oriented. The choice of components is therefore driven by their impact on the function in exam.

ECUs' architecture is no longer scattered but becomes a domain-focused system with central controllers, sophisticated software, and a proper network. Dedicated microprocessors replace embedded microcontrollers to enhance performance and control capabilities.

Consequently, OEMs and suppliers' know-how must shift towards software and system engineering.

3.2. Function-oriented Vehicles Development

One of the first aspects considered during the design phase of a new car is indeed the function list, which sets a reference point for the features and relative components to develop in order to achieve the desired product behaviors for the target customer.

Basic functions (technically called enablers) associated with single components are often part of a higher-level function related to a bigger, more structured system. Higer level functions usually operate across multiple systems.

It is therefore necessary to highlight that one component may take part in many functions and that a high-level function works on different components.

3.2.1. HIL Implementation

It appears now clear how HIL can be adopted both by engineering departments to develop and test ECUs, working mostly at the component level and by integration departments to validate system functionality overall, focusing on the functions.

Hence, HIL systems must provide high resolution, multi-level, and multi-domain simulations, synchronously providing sensors and actuators fusion unit with the same scenario from all interfaces in real-time. The complexity of the HIL structure and the distinction between what should be simulated and what should be a real component depend on the development phase and on the specific necessities of the project.

Usually, component HIL just needs the ECU as SUT and the few peripherals which it requires to function. Other inputs get simulated by a dedicated module and the test is conducted directly on the test engineer's pc. Integration HIL, on the other hand, demands more complex simulation, many ECUs, and different peripherals. Those can go from system level up to whole vehicle level, including all main ECUs and powerful testing modules.

3.2.2. Principles and Benefits of Function-oriented Engineering

Function-oriented engineering (FOE) rotates across the function and the component is designed with the only goal of supporting it. FOE promotes the seamless integration of various subsystems to ensure that they work together to satisfy the customers' needs.

Starting from around 600 existing enablers, a 300-350 elements function list emerges from strategic, user-centric objectives and the related requirements are drawn up with the same goal, ensuring that all functions meet the desired performance, safety, and regulatory standards.

A system designed around functions allows for their OTA updates throughout the vehicle's lifecycle, so as to improve or customize them. This improves after-sales experience for the customer and reduces development time for OEMs.

Focusing on functions means keeping ECUs at the center and, subsequently, their peripherals and network. Development and testing can therefore shift towards virtualized development and HIL testing, enhancing cost and process efficiencies, as previously discussed.

Clarification is now mandatory. Even though all engineering processes for developing a new vehicle apply a function-oriented approach to some extent, we are analyzing it with exclusive regard to system engineering. In particular, the overall system we consider is the one including all ECUs, their network, the sensors, actuators, and peripherals with all their electronic boards. *.11 .3*

3.3. Model-Based Design for Automotive Systems

FOE's emphasis on continuous Improvement and HIL implementation favors agile work methodologies over waterfall. This is why the whole function-oriented development process leverages Model-Based Design (MBD).

Model-Based Design (MBD) is a design methodology that streamlines the development process by using models as the central element of engineering, testing, and integration. Through simulation and virtual models, MBD enables iterative development and testing loops. These loops help identify and mitigate potential issues early and throughout the development process.

This approach is crucial for complex and safety-critical systems, enabling more in-depth product development, testing, and validation. This is achieved through various levels of simulation, including Model-in-the-Loop (MIL), Software-in-the-Loop (SIL), Hardware-in-the-Loop (HIL), and Vehicle-in-the-Loop (VIL). These models represent the system's behavior, control algorithms, and physical components and usually run on MATLAB/Simulink, dSPACE, or Vector environments. *.11*

3.3.1. MBD Scheme

The main steps in MBD include model creation, code generation, simulation and testing from lower to higher levels. The primary goal is to ensure that the system meets its requirements and performs reliably under various conditions. This involves iterative refinement and validation of models to detect and correct issues or faults early in the development process.

Earlier loops focus on lower-level aspects (such as software), while later loops address higher-level, integrated systems. One loop starts when the previous one has reached a sufficient maturity level and keeps existing in parallel with its successors as long as corrections are needed. This facilitates agile continuous throughout the development lifecycle.

Starting from the function list to be developed, the necessary ECUs, components, connections, and software to be developed are defined. The MBD approach consists in looping the model, the software, the hardware, and the system as they get developed.

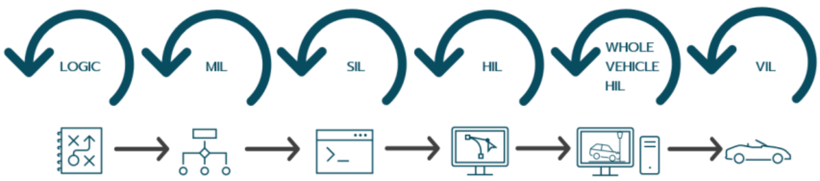

Figure 3: MBD process flow representation.

3.3.2. MBD Process and Loops

3.3.2.1. Logic

The first step is to define by major point what the function actually is and what it does involve, including requirements and parts involved.

During the system design phase, a draft of the overall logical flows is defined, based on the desired functionality, and starting from already developed technologies.

Models of control algorithms and decision-making processes embedded within the models are developed. Logic's robustness and system's requirements get finally checked.

3.3.2.2. Model-in-the-Loop

At this point, the theoretical function must be modeled, and its flow chart must be designed.

MIL validates preliminary control models against virtualized plant models within a software environment. It is used in the early stages of the process, detecting errors related to model behavior.

This helps to ensure that the model meets the requirements through iterated simulations, before moving to the proper coding phase and can be performed at system, subsystem or even component level, depending on the complexity of the final object.

3.3.2.3. Software-in-the-Loop

Once the model is set, the software must be developed. SIL focuses on testing and validating the entire control system, including the actual software, using virtualized models. Since the final hardware is not available yet, the software runs in a simulated environment, allowing for the validation of its behavior before it is even installed on the actual one.

SIL is used in the middle of the descending branch of the V-Model. Major issues at this stage indicate that something might have to be fixed within the model. Otherwise, the software version is stable and it's possible to proceed with the process until further refinement is requested by the steps below in the workflow.

At the final steps of SIL, a sort of simplified HIL bench might be adopted and to test the control algorithms before the actual ECU is available. In this case we talk about rapid control prototyping (RCP).

3.3.2.4. Hardware-in-the-Loop

At this stage, the virtual part of the function is pretty much defined, yet the actual hardware ability to run the function still has to be refined.

As the various releases of the software lead to its definitive shape, the latest version gets flashed (i.e. installed) on the ECUs or computers (if the component is simulated).

This phase permits us to test three key aspects. First, it ensures the software compatibility with the electronics' architecture itself, making modifications when needed. Secondly, it assesses proper communication between the software and the electronics, including the real sensors and actuators. Lastly, the desired functionality is tested before final integration.

HIL test benches main constituents are:

The ECUs, which are being tested and directly control the vehicle's functions.

The function-related software running on the tested components.

Actuators and sensors providing realistic feedback to the ECUs.

Real-Time simulation computer, such as dSPACE, Vector or National Instruments modules, to run the proper vehicle and environment simulation.

The fault injection units, responsible for the various perturbations and faults introduced into the system to test its response.

Input-Output (I/O) interfaces to make communication within all parts possible and to control the data exchange thanks to communication buses such as CAN, FlexRay, Ethernet or LIN.

A physical structure with all the needed power supply and data connections.

HIL can focus on signals, power-related components, mechanical interactions effects on the system, and many others, depending on the necessity and the system to test.

As the tip of the V is reached, various back and forth between SIL and HIL to refine the software and hardware integration allow to produce the definitive software and hardware, making a function (or part of it) perform smoothly.

As we proceed through the development, HIL benches pass from focusing on the component's functions, to integrated functions, incorporating all main ECUs, and simulating the whole vehicle dynamics and its systems in a realistic simulated environment. *.2*

3.3.2.5. Vehicle-in-the-Loop

Once the integration tests are performed on a bench in a simulated environment, there is still the necessity to assess the whole system's safe behavior when installed with all physical interaction and seamless integration in real operational scenarios. Those tests are just not possible unless on an actual vehicle.

In addition, a complete prototype constitutes a perfect entity to demonstrate the project status to the management and to the client.

This is where VIL comes into play. VIL testing involves connecting a real vehicle to a computer and a testing module. The main objectives are to test systems response to final user interactions, real driving condition and electrical stress tests. Depending on the test objective, the vehicle can either be stationary or in motion on the road.

The first prototypes are usually derived from previous models and adapted with new components. Pre-series prototypes only appear in a later phase. Clearly, on-road testing is possible only at the final stages, where a certain safety level can be guaranteed.

3.4. V Model for Vehicle Development

The V model is particularly suitable to represent the systems engineering process.

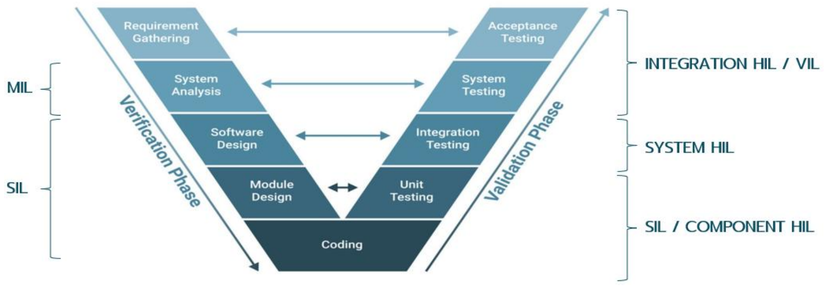

Figure 4: Automotive V-Model diagram.

As shown in figure 4, the V-model is characterized by two branches: the left, descending one represents the design and verification phase; the right, ascending one considers the validation and testing part of the process. Furthermore, the higher steps occur at the system level: requirements of the overall product and all architecture processes are defined. Then the whole vehicle is developed with increasing focus on subsystems and components at the lower levels of the V. In the end, the focus expands again, and system integration is validated and tested.

3.4.1. Integration with MBD

The curly brackets on the sides highlight how different steps in the V process are related to MBD phases.

As the system's requirements and structure are defined, the model gets built as well (MIL). The associated software is then programmed with the SIL, and coding is performed at a component level to develop the various enablers included in the ECUs.

MBD is integrated also on the validation side. Components are indeed tested with HIL structures, such developed components are implemented within system HIL benches to validate subsystems and final integration tests are performed with all ECUs mounted on a bench first, and on prototype car later (VIL).

The horizontal arrows suggest how issues and faults discovered while testing at lower levels are attributed to lower-level engineering on the development side. The same thing occurs higher, at the system level.

Therefore, it is crucial to note that the later faults are discovered, the further back one must search for solutions, potentially requiring numerous downstream component readaptations and leading to unexpected increases in costs and time.

The V follows the technical process. Management and support activities, such as project management, release management, information management, supplier management or quality management are not included and will be discussed later on. *.3*

4. Test Entities Implementation and Applications

Let us now deep dive into the actual automotive system engineering process, highlighting in particular which HIL test entities exist, how they are positioned, and which other steps are related to their use.

4.1. From Function to Test

4.1.1. Requirements

As soon as the vehicle functions (VF) for a new project are defined, the primary step is to perform a Requirements Analysis. Many tools come in handy for this purpose. A common one is DOORS (Dynamic Object-Oriented Requirements System), a requirements management tool by IBM. It assigns a unique identifier to each requirement within its database, helping in tracking, managing, and linking requirements throughout the development lifecycle.

4.1.2. Test Catalogue

Starting with the use cases related to those functions, the related test cases are defined by the function owners and included in a test catalogue. The latter must always be maintained up to date, just like the testing toolchain requirements. If a particular function is a matter of homologation, a dedicated team carefully analyzes the requirements and constructs specific test cases, following the norms.

A test catalogue is organized with the following fields:

- DOORS ID: a unique identifier assigned to each test case within the DOORS database, ensuring traceability.
- Requirement ID: identifies a specific requirement that the test case is designed to validate.

- SUT: Refers to the tested ECU, subsystem, integrated system and so on.
- Test category: Common categories include functional tests, performance tests, safety tests, legal requirements tests, communication tests, wake up/sleep tests, quiescent current tests.
- Test type: indicates whether a test is manual or automated.
- Test objective: describes the purpose.
- Test specifications: Detailed criteria or steps for conducting the test, including setup, inputs, procedures, triggering events, needed manipulation, operative conditions, whether it is static or dynamic and more.
- Precondition: Initial conditions required before executing the test.
- Parameter / Signal: Refers to specific parameters or signals monitored during testing.
- Expected result defines the correct SUT's behavior of when functioning as expected. It is used to determine whether the test passes or fails.
- Country variant: Indicates if there are different versions of the test for different countries, depending on different regulations, standards, or market requirements.

4.2. HIL Development

4.2.1. Rack Development

As soon as requirements and test catalogue are available for a new project, component owners need HIL benches to start validating the developed or commissioned components and function owners need HIL benches to develop their products. Hence, the team evaluates the best solutions linking new necessities to known concepts, also adapting ideas from previous projects.

Years of technical know-how allow us to choose the most appropriate suppliers accordingly. Vector, for example, excels in real-time simulation but struggles with partialization. dSPACE performs better in parallelism and integration but is more complex to integrate. For component HIL, Vector might be a valid solution.

Components, wiring, and structure are finally confirmed, and benches' production can start.

4.2.2. Software Development

In parallel to the test definition phase, the simulation software gets developed. In particular, the test catalogue gets implemented into the test pc and automated. The simulation must also be modeled.

4.2.3. Automation

Depending on the scheduled test catalogue, HIL testing can be manual or automated. By now, up to 90% of test cases happen automatically to improve efficiency and consistency.

Current trends also include the automation of test results analysis to quickly identify issues, automatic generation of test reports to streamline documentation and automated validation procedures.

4.3. Modeling

In order to exploit the above-mentioned advantages of virtualization, the first step is to implement a model to describe the SUT or the virtual environment.

A model is a mathematical structure tailored to include the parameters needed for a specific test. It is installed on the testing PC for the HIL platform to access and execute.

4.3.1. Environment Modeling

In HIL bench tests, the environment is always simulated. Depending on the scheduled test cases, it may include many different virtual stimuli, made to trigger the SUT response. A complete replica of a physical set may also be designed to test the systems under complete virtual journeys.

In general, the parameters can be physical or logical. The physical ones may represent mechanical or electrical characteristics of a vehicle or component. The model can either be static or dynamic. The logical ones simulate the behavior of a sensor, actuator or other board which is not physically included on the bench but should be in connection with the SUT.

4.3.2. Models and their Use

4.3.2.1. Vehicle

The vehicle dynamics model simulates the overall mechanical behavior of the vehicle, including its motion and response to various forces and inputs. It is fundamental especially for virtual journeys, where a reliable digital twin allows us to validate all functions encompassing real world reactions.

These models are usually developed using software like MATLAB/Simulink. Two main options exist. For particular project necessities and fewer tests needing more elasticity, the parameters can be measured and implemented in a proprietary model by OEMs. If a more standardized approach is needed, for wider scale testing, the model's creation can be outsourced to the HIL supplier.

4.3.2.2. ECUs

Some test scope might not require certain ECUs, or those ECUs might be tested in a separate seat. In those cases, the specific ECU is not an SUT. Nonetheless, it might be emulated if its presence affects the behavior of other SUTs.

The software for those ECUs can be directly implemented in dSPACE SCALEXIO, Vector CANoe, or others.

4.3.2.3. Sensors, Actuators and Peripherals

When not physically present on-bench, sensors and actuators are simulated. This simulation consists of inputting signals to the same ECU's pins or ports the actual components would use.

A quantity that would be measured by a certain sensor is fed in with the desired intensity. A feedback signal that would derive from an actuator is provided with the desired message. All the outputs provided by the SUT, including the signal that would go from the ECU to the actuator, are collected into the log and analyzed by the test engineer.

In here, fault injection can occur to test if the SUT reacts properly.

The decision of whether to use a real component or simulate depends on a cost and feasibility analysis. For instance, some signals might be difficult and time consuming to replicate and some real sensors might be difficult to stimulate on bench.

A similar discussion applies to peripherals.

Finally, note how simulation can be not only virtual, but also include physical parts. Typically, for example, electrical loads are simulated by rheostats, capacitors, and inductors.

4.3.2.4. Network

When not present or simplified, the general network must be emulated with its characteristics.

For instance, an essential component is the gateway. It acts as a central communication hub to which all ECUs refer, sorting input and output data and translating them into the correct signal protocol. When the gateway is not present and communications between ECUs must be analyzed, it gets simulated.

For ECUs communication at vehicle level, REST bus networks are typically used. Representational State Transfer (REST) is a flexible architecture allowing ECUs to interact with each other using Hypertext Transfer Protocol (HTTP). Again, this is ideal where various components need to communicate efficiently, as in automotive applications. At HIL benches, a dedicated REST Bus simulation engine manages all sorts of communications.

4.4. Validation

Validating critical components involves a meticulous process that ensures both the logical and physical aspects of such components are accurately represented before testing.

When validating only the electrical and electronic parts with logical structures, the model's reliability level is significantly high because it involves an algorithm for validation working in tandem with another algorithm simulating the component's logic.

However, the process becomes more complex when the physical aspects of the components need to be considered. If the component is reused from past projects or is simple and well-known, its model might be directly implemented on the test bench. Otherwise, a cost-benefit analysis and technical difficulty assessment must be conducted, considering the HIL test objectives to determine whether to include the real component on the bench.

In all cases, it is essential to study and report from known real-world use cases, conducting preliminary bench tests on the vehicle's physical components when present, and making reasoned hypotheses about how a will behave when subjected to real use conditions, if still in development.

If the choice is to include the component in the test bench it becomes an active part of the SUT, providing lifelike signals to the related ECUs. Preliminary studies are essential to understand the response to expect.

If the decision is to simulate the component, the studies performed aim to create a validation model starting from its physical aspects, mechanical reactions, time constants, and dynamics. The model is then implemented in software like IPG CarMaker or similar, making the component part of the simulation.

Ideally, if costs and schedule allow it, the trend is to include as many real components as possible in the whole-vehicle HIL bench, as the development proceeds. Following this highest level of HIL complexity, the next step involves operating a fully assembled vehicle within a simulated environment, known as VIL.

4.5. Component HIL

Component HIL test benches are essential to perform functional tests and validation over isolated ECUs and to evaluate single SW releases between a global release and another and before integration tests.

In a component HIL, a real ECU is connected to a simulation environment via its inputs and outputs.

4.5.1. Initiation

The process often begins by setting up a REST Bus simulation, commonly using Vector's CANoe. More precisely, a ".dbc" file defines how raw data from the CAN bus should be interpreted, converting them into values understandable by the test engineer.

Next, the most recent version of the ECU software is flashed, and the ECU is coded with the necessary updates using specific proprietary software. Afterwards, a check for any active Diagnostic Trouble Codes (DTCs) is performed to ensure that the system is clear from any previous issue. The Quality Bit for all quantities used by the SUT is then set to "valid," and all signals are set to represent the 0 position.

Calibration of all sensors, actuators, and peripherals is crucial for the system to operate its functions properly. Then, it is ensured that the system can indeed detect all attached components and their operative limits and zero points.

A CANoe experiment can be created to test the system and understand whether it is necessary to modify some to achieve seamless functionality, since the system on the bench may not react exactly as it would when assembled in a full vehicle, due to the lack of some physical interactions. The CAPL (CAN Access Programming Language) node sets all the static signals required to avoid DTCs and to allow direct SUT stimulation via CAN.

4.5.2. Testing

Simulation software replicates the system's behavior over time, it usually uses CAN-based systems, developed with specific environments, depending on the provider. dSPACE has its own engine, National Instruments uses LabVIEW and Vector runs on CANoe.

Let us assume we use Vector. CANoe control panel features a graphical interface similar to a flowchart. Each control unit is represented as a square running a simplified version of C. A "trace" window analyzes messages, allowing to send and receive messages with the control unit and viewing the log.

The test objectives include the proof of a correct function according to the component specifications, the robustness against invalid or unspecified input values (in the form of electrical and logical signals) and a correct reaction, the correct configuration, the use of appropriate diagnostic function according to the specifications, the correct implementation of safety requirements.

The first test usually assesses correct communication between the SUT and the associated peripherals or input variables (BUS signals). The entire functionality of an ECU and its SW Clusters get then validated, following the test catalogue.

Various kinds of errors may merge at this point, especially at the first stages of the project. It is the test engineer's job to keep track of them in the appropriate documentation, understand their origin and, where possible, try to cope with them. A ticket is opened in the management system to report the issue.

Complex problems are deferred to the owner for resolution. If issues persist, the process returns to the component owner and might go back to the SIL stage.

The issues are finally fixed, the SW is validated and the process loops for all new software updates and components.

If necessary, the testing toolchain and test cases are updated to meet current needs or adapt to new use cases. *.3 .12*

4.6. System HIL

4.6.1. Principles

As the scope of HIL systems expands, passing from component to system, test cases operate at a higher level, delving less into individual enablers.

The testing SW follows this logic: High-level nodes work on Dynamic-Link Library (DLL) at a lower level, adding function packages to existing code. DLL files allow multiple programs to simultaneously access the same library of data and instruction.

Once the single components are validated through component HIL or after a global release arrives from integration tests, parallel development and issue correction can occur on system HIL, maintaining good cost efficiency. Indeed, Due to the limited number of complex and expensive V-HIL benches and prototype cars produced, the use of automated tests on system HIL is mandatory. This way, a 50 K€ can run simpler tests, while a V-HIL (varying from 1 to 2 M€) can be used for higher level tests.

4.6.2. Testing

As the SUT includes many ECUs in this case, system integration tests involve both the gateway and the components. For instance, pressing a button to lower a window generates a message that passes through a gateway, goes to the window network, and informs other ECUs. The message's path is checked on CANoe, verifying what ECUs receive. If the window does not open, the issue can be traced back to address the issue. Another example can be a steering wheel command to change music. It involves infotainment and button systems on two separate CAN networks. The command must pass between them, ensuring that a radio command on one network is correctly received on the other.

The ICAS is the main gateway that transmits information across networks, routing them and managing functions' priorities. Some of the ECUs can be simulated if not of interest for the test catalogue.

Understanding how control units communicate and what they need to operate is crucial. For instance, if a camera is involved, CANoe does not process video, and a specific board is needed. This highlights the importance of having the right tools and setups for different components within the HIL system.

Figure 5: Example of system HIL. .13

As said, test cases are usually executed automatically. For manual testing, logs are taken from the CAN or Ethernet network, and parameters are set via interfaces or VASTester for specific machine or control unit settings.

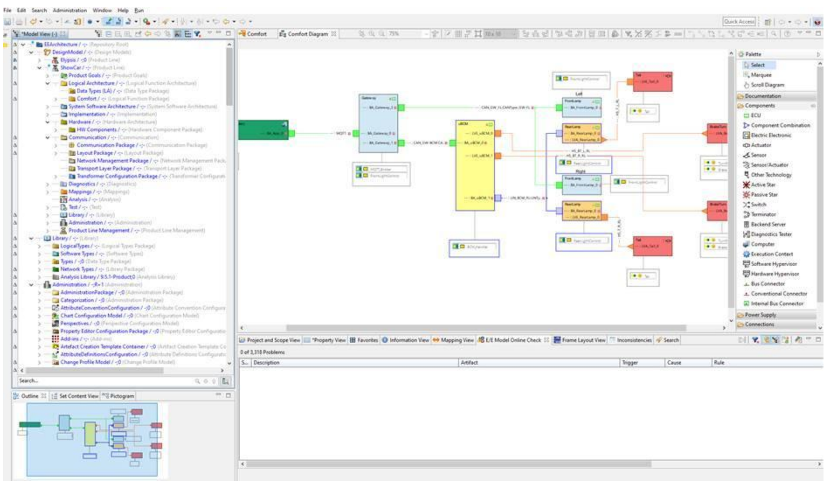

Figure 6: Example of blocks in a dSPACE control desk. .13

4.6.3. System Examples

System HILs are built for a specific system or subsystem to test and validate connections and basic functionality and smoke test.

Usually, this kind of HIL is built directly under the function owner department's request. Here are some examples.

4.6.3.1. Infotainment

specific desktop benches test infotainment and user-centered systems functionalities and UI usability. The system's ECUs are present, the others providing information for the driver onto the display are simulated. Benches like these also have a proper VIN number and a key, so that the system recognizes them as a real vehicle and is able to operate.

4.6.3.2. ADAS

ADAS also has dedicated benches for its many subsystems' validation. Cameras are usually simple to simulate through a custom FPGA solution, as it only returns video. Radar input must be processed before feedback is given. Many legal requirements are considered during these tests. For instance, simulating reverse gear insertion should activate the rear camera within a second. Even what may seem like a detail must be measured and verified.

4.6.3.3. Electric Vehicles

Specific benches are dedicated to EVs. Examples of SUTs include The Battery Management System (BMS), monitoring total and individual cell tension, temperature, state of charge (SOC), state of health (SOH), current flow, cell balance, and chassis isolation.

4.6.3.4. Combustion Vehicles

for ICE vehicles, Engine-in-the-Loop (EIL) systems are used to test and validate the integration of engine ECUs with other vehicle systems. These systems simulate real-world driving conditions and require highly detailed physical models allowing engineers to assess engine performance, emissions, and many operational parameters without the need for a physical engine, helping in complying with regulatory standards. *.12*

4.7. Multi-Bench Systems and Automation

System HIL's frontier allows us to use different benches across multiple global locations, enabling resource allocation based on availability, expertise, and market specifics. In addition, some HIL systems can work across many benches through a laboratory network, providing specific services. This can be done with radio and GNSS signals, for instance, providing their simulation to other HIL setups via Ethernet and specific cables for satellite signals.

Different test benches can be interconnected to perform integration tests on multiple subsystems, and they can sometimes be used for homologation purposes. This modular approach allows for flexibility and scalability in testing various automotive systems.

Automation trends include toolchain automation, integrating Automation Desk with software like IPG CarMaker for full test automation and automatic generation of sensor models.

Even though those innovations are very powerful, the most effective way to test complete vehicle integration is using a V-HIL. *.3*

4.8. System Integration Benches

The following test entities work at a higher level and are the core of system integration, not focusing on components or subsystems, but verifying the vehicle as a whole.

4.8.1. Lab Car

The lab car serves as a representative bench of the car's network, where all project ECUs, connection buses, and actuators are connected according to the real vehicle's architecture. In laboratory vehicles, the cabling of the bus systems is in fact implemented based on the current system circuit diagrams, following the networking topology, and made with comparable latency and thermal resistance to the real system, using all real control units and vehicle loads/parts.

4.8.1.1. Structure

All vehicle's electrical and electronic components are cabled on a standing aluminum frame and connected to the test pc.

Figure 7: A Lab car test bench. .14

4.8.1.2. Testing

The lab car is used as a first integration step to test the status of all SW releases coming from function owners on every ECU simultaneously to verify correct implementation and communication across the whole vehicle system.

Starting from the list of ECUs released by the component owners for a specific milestone, the required software is flashed. Verification ensures the absence of interferences and confirms both connectivity (messages are correctly sent and received) and communication (messages are correctly interpreted and used). Specifically, ECUs, possessing diagnostic addresses, are the components that communicate; other boards from certain sensors and actuators lack this capability, allowing only for connectivity checks.

A lab car also allows us to measure quiescent current, to verify correct loads and perform network wake-up and sleep tests.

No functional test is performed at this stage and no simulation takes place. For the same reason, there is no test catalog, just a standard set of tests to perform.

Test results are extracted from the bench data logger for analysis. Any emerging issue is identified, traced back to determine whether it affects the input or output ECU and reported into a dedicated ticketing system with logger trace attached. Since this integration test aims to ensure that the components function cohesively, working at low level, issues are usually a component owner concern. Rework is therefore usually required by the department, or the tier one supplier and component HIL validation follows.

These primary tests are fundamental to assess basic communication and safety of the system before further integration steps and flashing on prototype vehicles and tests take place. Lab car testing also constitutes the milestone to officially publish the global release, serving as a reference to state the development status and the latest software and network versions.

4.8.1.3. Test Management

Each lab car has a "vehicle manager". Since it serves different departments and integrates all possible vehicle versions, its users must reserve testing periods and book usage into the allocation plan in agreement with the vehicle manager.

4.8.2. Vehicle HIL

Modern cars feature up to 350 basic functions. V-HIL is a complex infrastructure enabling us to test all of these functions, grouped as clusters, within a simulated environment. The objective is to ensure seamless system integration and validation of networked functions during development and sub-releases, with the highest possible degree of automation and reproducibility. Particular attention is given to high level use cases even though communication between the systems and different kinds of multi-level and multi-domain tests can be performed. The tests are always functional, no component tests or diagnostic tests are executed and neither enablers are tested.

4.8.2.1. Tests

V-HILs' main point of strength are virtual journeys. In these tests, a complete model of the actual vehicle runs in real time inside a simulated and controlled environment. This is particularly useful when it comes to the validation of safety functionalities and with tests that would otherwise be dangerous or impossible to execute and replicate in the real world. This setup is also ideal for highly repetitive testing, as in the AD field. For example, Waymo's self-driving cars have driven over 32 Gm on public roads, but over 24 Tm have been driven in simulation.

After the tests, an overall maturity assessment attributes a grade to the system. It starts from 0 and reaches 100 when the development is finally completed.

4.8.2.2. Structure

The structure of a V-HIL can be modular and expanded or compact. The more compact the system, the less flexible it is, and the less troubleshooting and adaptability are possible.

Let us analyze in detail how its modular version is structured.

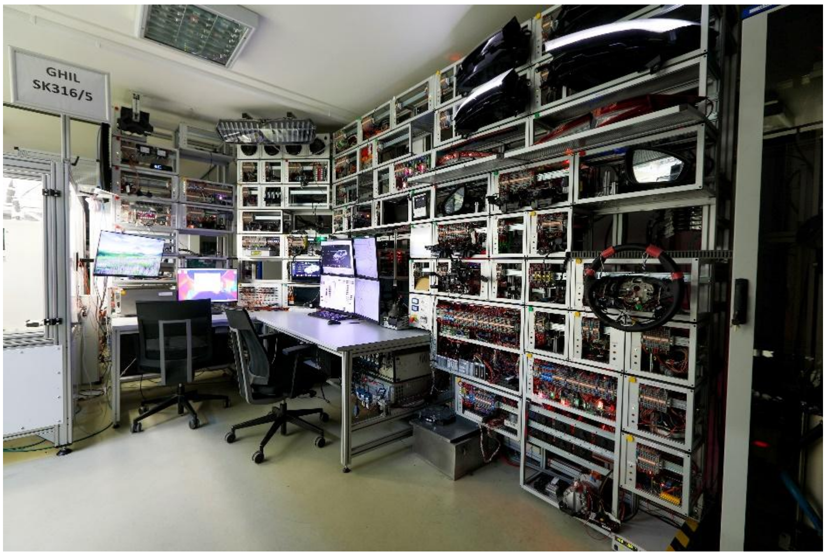

Figure 8: A V-HIL test bench developed for Skoda. .14

Figure 9: Within the modules, ECUs are wired to special probes. .14

The test pc containing the test management SW (automation desk) and the simulation management SW (control desk) is connected to the HIL platform (dSPACE, Vector, NI...) through a so called giga link, granting fast data transfer. The pc stores mathematical models of both vehicle and virtual scene and the HIL computer runs and manages the simulation. The two communicate to all ECUs thanks to mani I/O clusters, responsible for routing the signals, one per communication protocol used.

System partialization (to work only on the desired SUT) is allowed by particular data logger and bus panel modules, which aggregate clusters of other ECU modules (from the main vehicle systems) to make them work together.

For the sake of partialization and flexibility, ECUs are divided into modules. Each one of them is a small aluminum frame embedding the physical sensors, actuators, and peripherals necessary to test the SUT as well as the terminal blocks for internal wiring.

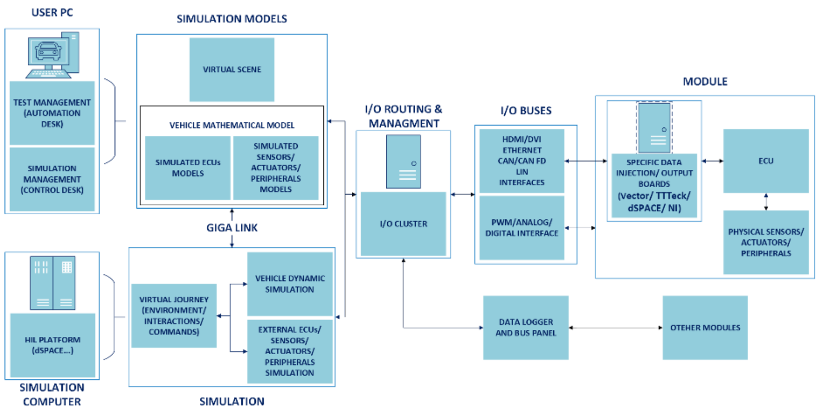

Figure 10: Logical scheme of a V-HIL structure and communication paths.

4.8.2.3. Variants

Both in the lab car and the Whole-vehicle-HIL, if there are multiple versions of the same ECU that need to be tested separately the bench must be comprehensive for the full optional, but there is a difference. In the lab car, as an example, for antennas working on different frequencies for different regions, it is sufficient to mount the new component. However, in the Whole-vehicle-HIL, there is simulation, so the entire test needs to be re-parameterized.

4.8.2.4. Simulation

Each module reports all testing information to the simulation through its specific communication buses. CAN, CAN FD, LIN, and Ethernet may need complex boards for data injection, tracking and reading (e.g. Vector box, TTTech, dSPACE, NI, ETAS, Expleo, NOFTS). This is not needed with PWM (square wave) and simpler analog and digital electrical signals. Those boards are usually assembled on modules.

Whatever is simulated is not SUT. The choice depends on space, cost, and flexibility needs. If the simulation is too onerous and complex to be simulated easily, a real object is used. For example, the steering block is often real because it is difficult to simulate. Complex safety protocols instead, are difficult to simulate. For example, a variable geometry turbocharger can be very expensive to simulate, even more than the part itself.

The simulation may also include hardware parts, such as rheostats, resistors, inductors, and capacitors.

Simulation software solutions can also vary. For instance, CANoe uses event-based language, so a series of signals can be accessed, calculations can be made accordingly, and the output of my calculation can be reported instead of the ECU's output through a vector box. Exam, based on Python, can also be used, allowing interaction with various technology providers thanks to already developed libraries. In this case, SUT coordinates are given to a so-called method, and the instructions are automatically written in the appropriate low-level language.

The latest frontier is to enhance complex sensor fusion through AI algorithms, creating the highest quality emulation that allows to perform preliminary Euro NCAP tests before in a virtual 3D environment to achieve the best results in the actual one. *.15*

4.8.2.5. Autonomous Driving Testing Example

Let us consider, as a practical example, a small vehicle for autonomous driving experiments.

Before driving in a real environment, it was tested virtually.

Vehicle data were inserted with IPG CarMaker for the physical model, while the logical model was built on Simulink. The sensors were modeled from a generic model in this case but parametrized in the same way as the actual ones.

The environmental simulation was also specially designed to evaluate vehicle's reactions. It was a reproduction of

Italdesign's seat and a driving path with static and moving obstacles.

In cases like this, specific scenarios with predefined maneuvers to perform in response to a set of situations are described and must be validated during the test.

IPG CarMaker ran on the PC and provided inputs to Simulink.

The real ECUs under test and sensors were connected to the test within a dSPACE Compact V-HIL, providing data injection to make them interact with the simulated environment.

A virtual journey was performed. The molded vehicle moved in the virtual path and reported all sensor data. At the test user interface, the exact environment reconstructed through the sensors and the signals exchanged within the SUT could be monitored. They also were manually manipulated, and automatic tests could be executed by pressing specific commands from the dSPACE control desk. *.16*

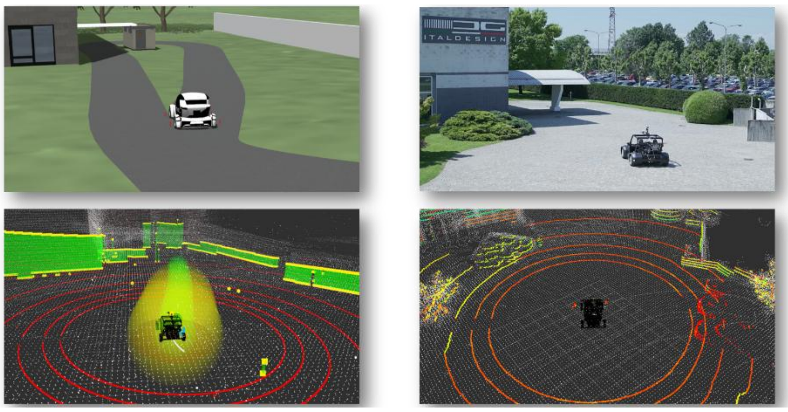

Figure 11: An example of virtual journey, the environmental simulation, sensor data injection and real-world environment are visible. 555555

At the bottom, LIDAR and radar virtual views are visible. The input was injected during the simulation.

The same path was followed afterwards by the prototype vehicle, and it indeed behaved as expected. *.12*

Figure 12: Prototype vehicle inserted in the loop at the next stages. .18

4.9. Vehicle in the Loop

As a final integration step in the MBD, the real vehicle must be taken into account to assure that the integration performs correctly even when physical interactions are present and to set a final status for each phase of the project. Of course, real prototypes are less flexible than a test bench: a single global release for a single version can be tested at a time and each phase requires a new testing vehicle due to wear and consequent unreliability of the results. At the first phases of the project one test car is sufficient and it is usually a "technology carrier", derived from dismissed passed models, but including new prototypal components. At the latest phases, more are needed in order to test different versions in parallel.

A fully equipped test vehicle can cost more than 1 M€. *.17*

4.9.1. VIL Test bench

As anticipated, the step occurring right after V-HIL is VIL. In these test benches, a complete and non-exploded vehicle operates in a controlled and simulated environment. The simulation, in this case, comes by physically stimulating the vehicle the same way a road journey would do, often starting from the wheels. To be precise, latest applications allow some sensors to be stimulated OTA. This facility is designed for comprehensive testing of conventional, hybrid, and electric vehicles and powertrains. It allows us to conduct standardized and custom test cycles, measuring performance, emissions, and energy consumption. This is crucial for ensuring the functional reliability of these systems over time, despite potential disruptive factors like aging materials and defective maintenance. *.19*

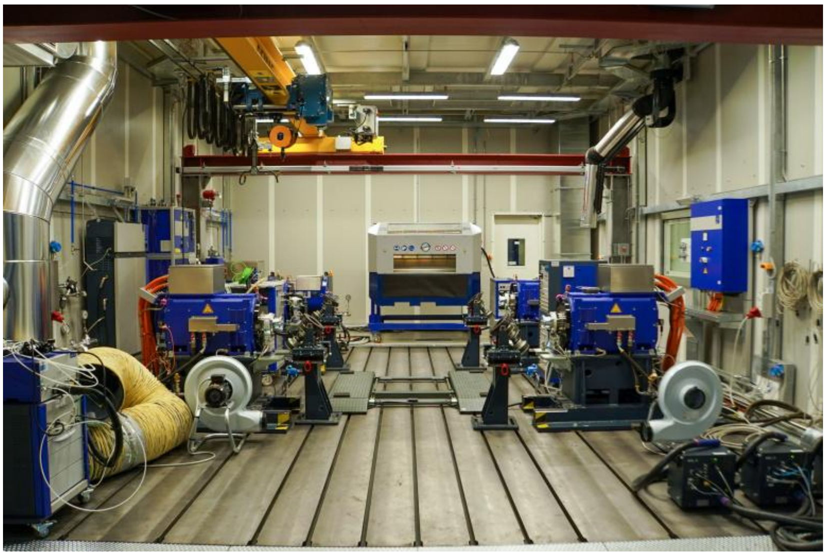

Figure 13: VIL-HIL testing facility at Center for Automotive Research and Sustainable mobility @PoliTO (CARS). .20

4.9.2. The Reference Vehicle

The reference vehicle is an actual car used for electrical and safety tests. In the reference vehicle, as in the lab car, all components are present, there is no simulation, and the validation occurs mostly in an automated manner.

The goal is to ensure safety and reliability for the next loops and, ultimately, for the customer.

4.9.2.1. Structure

Key features include an external, adjustable voltage source connected to the negative pole of the vehicle's battery, which acts both as generator and measurement system. There is a second turret with a PC connected to the OBD (on-board diagnostics) to read the entire vehicle at the beginning and end of each impulse, and another turret connected to the start button to automate the ignition as the main command. A climatic chamber is finally used for temperature related tests.

4.9.2.2. Tests

The reference vehicle checks a vehicle's electrical/electronic system in real-world situations and under stress and endurance operation.

Tests performed include over/undervoltage tests, cable-based disturbances, start voltage impulse, interference voltage, behavior at rest, quiescent current, generator malfunctions, and electrical network status, generator faults, reset behavior, starting voltage pulses, interference voltage measurement, temperature effects.

Figure 14: A refence vehicle being tested, AI generated.

4.9.3. Master Vehicle

4.9.3.1. Structure

The master vehicle is again a real car always updated with the latest software, making it suitable for management .12

and client presentations. It is used for both high-level smoke and dynamic tests and to review the status of functions throughout the project phases.

4.9.3.2. Tests

On the master vehicle, high-level functions are tested from the customer's perspective, only high-level functions with which the driver will interact are considered. For each global release, it is verified that everything declared is still present and functioning in the new release.

In this case, the test catalogue is mainly based on real-world interaction, even though a test pc is usually brought on board to control operational parameters and perform some fault injection tests. Errors might be challenging to track at this seat, especially if development is advanced. The single use case might indeed pass, but the concatenation of operations through different systems can lead to unexpected faults, which might even relate to a component that passed its use cases. When found, the DTC detected by OBD port and exact process to find the fault is inserted in a ticket for its resolution. Minor issues, like an incorrect tire pressure warning light, can be resolved with a key cycle. The problem might just be, for instance, that in certain cases an ECU requests pressure control but does not receive feedback, triggering the warning light.

DTC is also analyzed to verify the response to fault injections, ensuring a correct response.

Upon reaching a specified system maturity level and adhering to OEM procedures, component owners grant road-approval for their respective components. Once all safety critical components are approved, the new vehicle receives camouflage to preserve confidentiality road testing can finally occur.

Figure 15: A camouflaged master vehicle during track testing, some sensors are visible on the extern. .21

5. Testing in the Development Process

5.1. Overview

The complete development process for a new vehicle, from project start and feasibility study to the start of production (SOP) takes around five years. The testing and validation process takes place from around the 4th year to SOP. In particular, it starts as soon as the project concept is approved, and the functional packages are frozen.

From there, the complete list of the functions involved is indeed defined and each enabler and higher-level function is meticulously described. The function variants are organized in a tree-structure categorized firstly by function area, then by function family and lastly by customer vehicle function.

Using waterfall practices for overall vehicle architecture and agile teams for detailed requirements and component development.

5.2. Release Management

The global release management allows us to make project statements from functional freeze to SOP at synchronized times. Those milestones state the global network development status which, highlighting specific function maturity levels defined a priori.

Once those milestones are programmed, the next step is to create a test plan. Depending on the project needs, the number and specifications of test entities is defined. A global release schedule is then created. their number and requirements are planned from the beginning.

It is an advantageous method since it unequivocally indicates the objectives to reach, and which are the stable vehicles' versions to take into account during SW and HW development cycles. Frequent, agile, and predictable SW updates improve bug fixing. Eventually, it increases transparency and compliance and fosters commitment through clearly defined goals.

5.2.1. Test Plan

After MIL and SIL, the test plan includes diagnostic and flashing tests, component and system HIL, network testing, installation tests, Lab car testing, V-HIL, Reference vehicle testing, Master vehicle testing and pre-release tests. During the latter, a final maturity assessment is performed, and an overall network maturity report is prepared and shared with committees.

As discussed, global releases are published right after laboratory vehicle testing. It includes functional maturity comparisons, test results summary, updated HW/SW versions for all ECU for departmental tests.

The rhythm is quite regular, and they are spaced a few weeks apart.

5.2.2. Function-Oriented Development Milestones

The main milesotnes before SOP include:

- Technology carrier with non-specific components and basic functionality testing for early timeline functions.

- Prototype components are then added and basic functionality is granted.
- All functions are installed.
- Pre series components and all functions implemented and testable on prototype.
- Series components and All functions suitable for broad validation.
- An analogous dedicated process follows homolologation-relevant components.

5.2.3. Global Releases

In between the global Network statements, the maturity planning is followed to publish global releases.

There different functional maturity levels are defined as follows:

- Function implemented with bugs.
- Function fully implemented with reduced number of bugs.
- Function fully and seamlessly implemented.
- As the single function maturity level increases, the overall system maturity increases towards 100%.

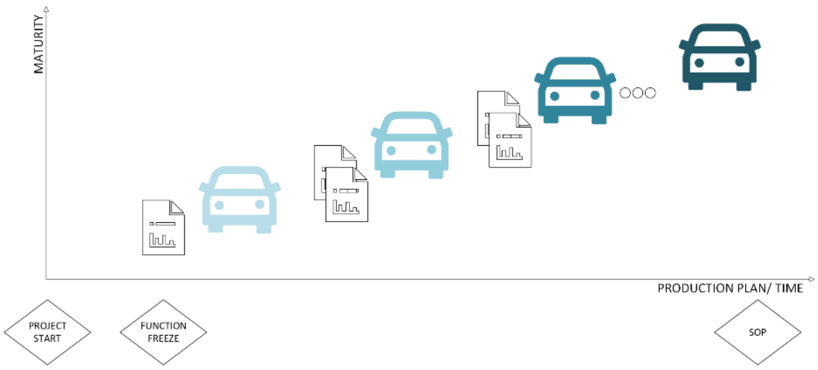

Figure 16: General ramp up of the development process through network milestones and general releases.

5.2.4. Roles

Main roles involved: functional project managers, functional test managers, component and function owners, component and SW developers.

6. Virtualization Within the Overall Processes

In this text, we analyzed system testing and validation process and entities, with a particular focus on how it is automatized and virtualized.

A brief overview of how other processes are digitized is useful to understand how the same logic - producing fewer physical prototypes for increased leanness and efficiency - is applied elsewhere.

6.1. Augmented Reality Design Validation

Augmented reality for instance, is an innovation with strategic applications in matter of reducing the number of physical prototypes. An excellent example can be found in interiors concept development and design proofing, at the early stages of the process. Interior ergonomics and related styling feasibility can be validated via specific tests in an adaptable and comprehensive augmented reality lab, without the need for a physical mock-up, which can be difficult to adapt to design changes.

Let's start with the styling feasibility. During the design process (especially if that’s for an entirely new vehicle) the laboratory translates design ideas into practical solutions, in collaboration with the styling department. Starting from a reference car, some “hard points” i.e., key positions for elements such as seats, steering wheel, and pedals, are defined. Pillars, central tunnel, and visibility angles are also evaluated. The concept lab integrates physical models only for the necessary parts, reducing costs, time, and waste. The rest of the information comes from the highly detailed 3D model, which is visualized and analyzed.

Considering ergonomics, the concept lab validates each improvement to ensure that the interior components’ positioning is ergonomically correct. This monitoring continues until the concept freeze, ensuring that the final design meets ergonomic requirements and is functional for the vehicle occupants. The operator evaluates ergonomics and establishes optimal positions with activities such as reachability tests and virtual measurements.

The system includes a VR headset, infrared cameras, and hand-tracking gloves. As for advanced tracking technologies, the concept lab uses an advanced tracking system to measure with millimeter accuracy the interactions between the markers on the user and vehicle elements.

Finally, the entire process is simulated in a virtual scene, using software such as VRED Autodesk. Quality is tested with a light beam, and the physical layout is certified with tolerances between 3 and 5 mm. Advanced testing sessions may include clinic testing.

Figure 17: A VR test bench during a driving test. .22

Figure 18: The simulated environment during a reachability test. .22

The creation of virtual models not only helps with complex electrical-electronic systems, but it can be applied also to mechanical systems, in order to understand their behavior before even producing mockups and because fast looped improvement can be implemented as the development proceeds and as new needs are found.

6.2. Mechanical Systems Engineering

It can be done for mechanical systems with complex interactions.

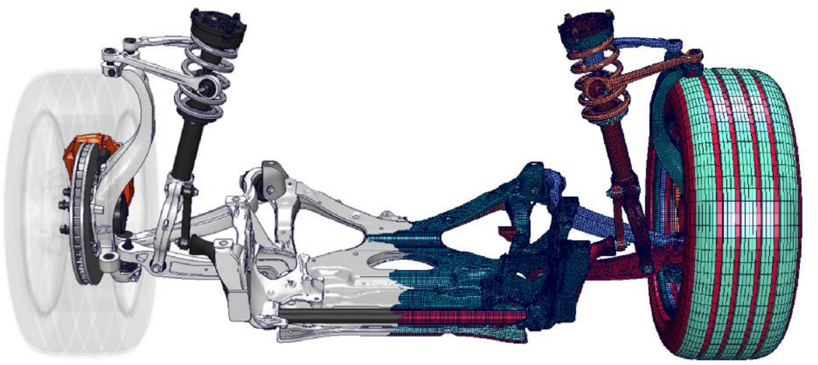

Figure 19: Model of vehicle suspensions. .23

6.3. Thermal-fluid Dynamics Optimization

It can be applied to aerodynamics and heat dissipation studies.

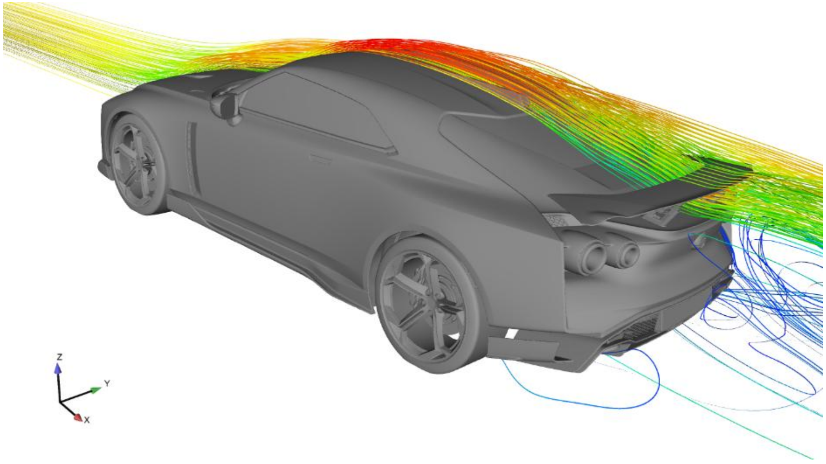

Figure 20: Aerodynamic simulation on a GT-R 50 .24

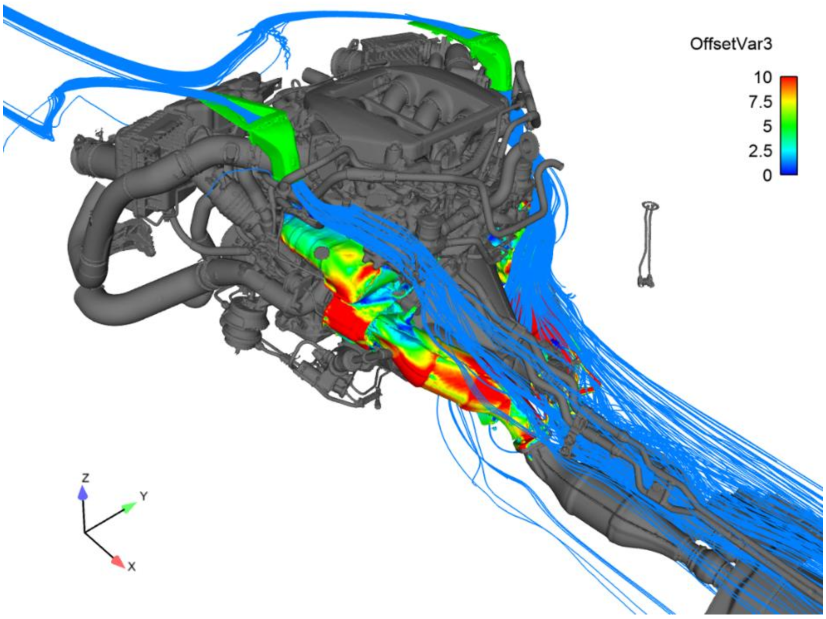

Figure 21: Fluid dynamic analysis on an engine to study convective heat transfer. .25

6.4. Crash Tests

The same idea of virtually validating safety critical systems before an appropriate maturity level is reached, is seen in the engineering of airbags, and retaining systems, for instance.

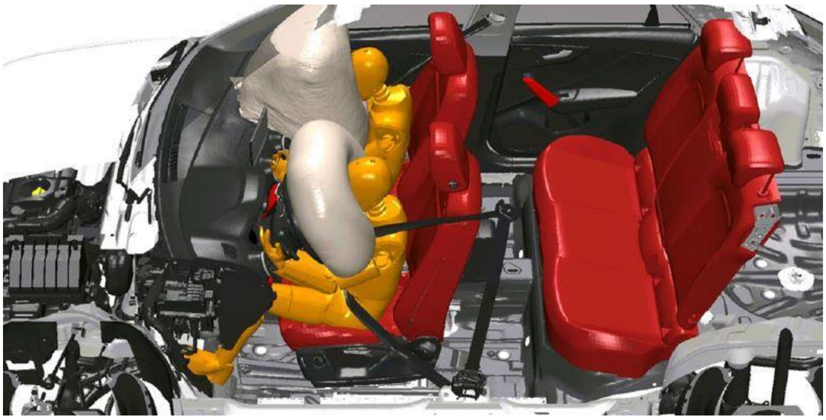

Figure 22: Virtual crash test example. .25

6.5. Structural Optimization

Structural optimization can be performed virtually to establish the best geometries to implement.

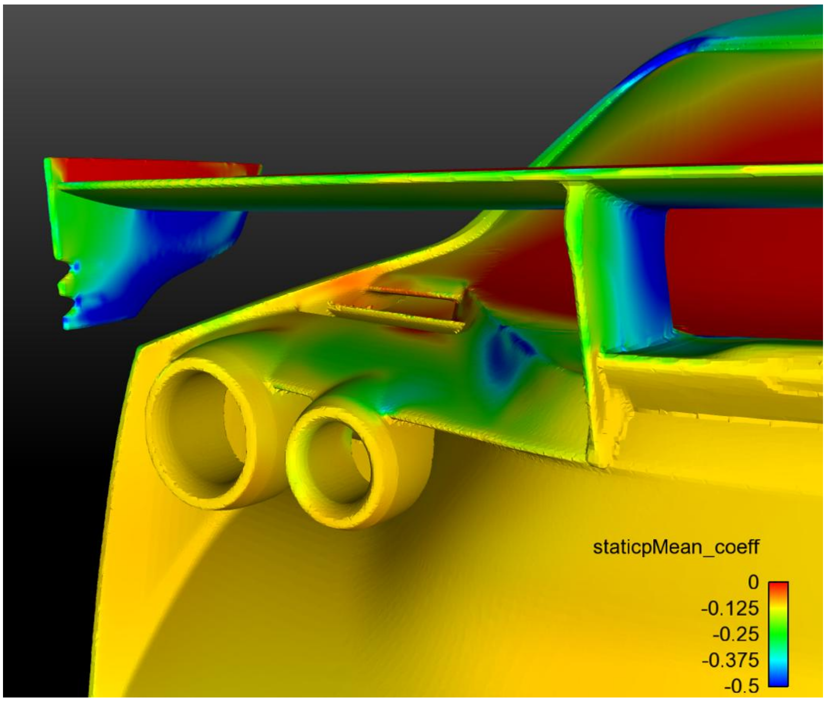

Figure 23: Virtual structural analysis. .25

6.6. Production

And finally, production itself can be simulated to maximize its efficiency.

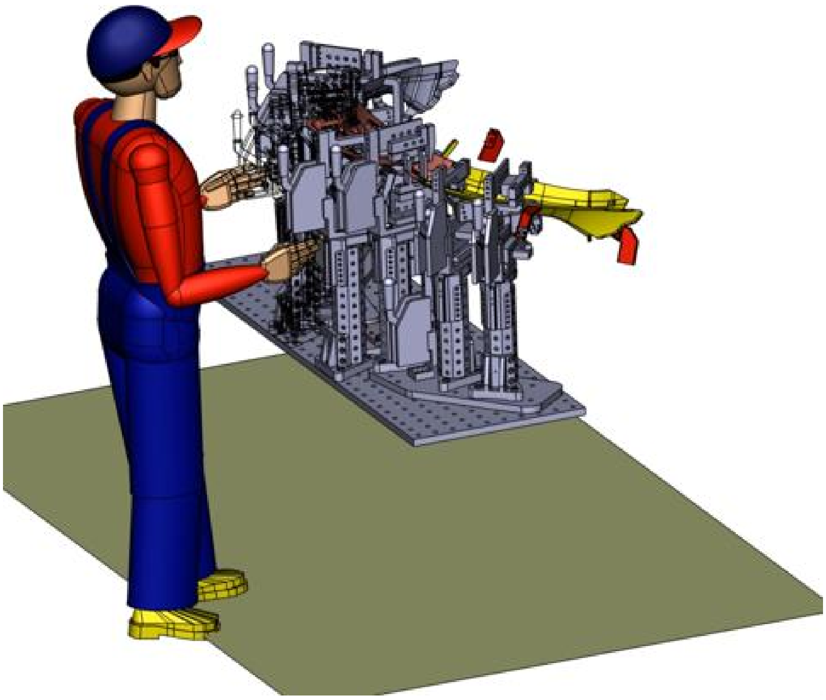

Figure 24: Simulation of production process at welding fixtures work station .26

7. Conclusion

In this thesis, a comprehensive analysis of the testing and validation processes within the automotive industry (with a particular focus on HIL) was provided. Objectives included describing the evolution and implementation of HIL testing, analyzing its technical and managerial benefits, detailing the specific processes and technologies involved at Italdesign Giugiaro S.p.A., and highlighting the connection between technical testing processes and project management activities.

We began by exploring the historical context and evolution of HIL testing, we then proved valid technical and engineering reasons for its implementation, emphasizing its efficiency, safety, flexibility, and user-oriented benefits. By integrating real hardware components with dynamic mathematical models, HIL testing allows for comprehensive, reproducible, automated and scalable validation of SUTs and seamless system integration, enhancing the overall product development cycle, from early-stage debugging to regulatory compliance.

Significant space was dedicated to the specific processes and technologies, demonstrating how many different technologies, integrated with a MBD logic into the V model, contribute to the same goal.

Finally, we saw how many virtualization techniques are applied across different stages of the vehicle development process, reducing the need for physical prototypes and enhancing efficiency.

In conclusion, this thesis provided a thorough analysis of innovative validation and testing techniques and their strategic role within the automotive industry.

Overall, this work has emphasized the importance of integrating technical testing processes with project management activities to achieve a holistic and efficient development cycle.

Glossary

ABS: Anti-lock Braking Systems.
AD: Autonomous Driving.
ADAS: Advanced Driver-Assistance System.
BWT: Black Box Testing.
DLL: Dynamic-Link Library.
DOORS: Dynamic Object-Oriented Requirements System.
DTC: Data Trouble Code.
DTC: Diagnostic Trouble Code.
DUT: Device Under Test.
ECU: Electronic Control Unit.
EIL: Engine-in-the-Loop.
EV: Electric Vehicle.
GWT: Gray Box Testing.
HIL: Hardware-in-the-Loop.
HTTP: Hypertext Transfer Protocol.
IOT: Internet of Things.
I/O: Input-Output.
ICE: Internal combustion engine.
MBD: Model-Based Design.
MIL: Model-in-the-Loop.
OBD: On-board Diagnostics.
OEM: Original Equipment Manufacturer.
OTA: Over the Air.
RCP: Rapid Control Prototyping.
REST: Representational State Transfer
SIL: Software-in-the-Loop.
SOP: Start of Production.
SUT: System Under Test.
UI: User Interface.
UX: User Experience.
VF: Vehicle Functions.
V-HIL: Vehicle-Hardware-in-the-Loop.
VIL: Vehicle-in-the-Loop.
VR: Virtual Reality.
WBT: White Box Testing.

List of Figures

Figure 1: Main HIL innovations timeline. 20

Figure 2: McKinsey consumer pulse survey, 27869 international participants, Dec 2022.. 23

Figure 3: MBD process flow representation. 28

Figure 4: Automotive V-Model diagram. 32

Figure 5: Example of system HIL. 43

Figure 6: Example of blocks in a dSPACE control desk. 43

Figure 7: A Lab car test bench, AI generated. 46

Figure 8: A V-HIL test bench developed for Skoda. 48

Figure 9: Within the modules, ECUs are wired to special probes. 49

Figure 10: Logical scheme of a V-HIL structure and communication paths. 49

Figure 11: An example of virtual journey, the environmental simulation, sensor data injection and real-world environment are visible. 50

Figure 12: Prototype vehicle inserted in the loop at the next stages. 52

Figure 13: VIL-HIL testing facility at Center for Automotive Research and Sustainable mobility @PoliTO (CARS). 52

Figure 14: A refence vehicle being tested, AI generated. 54

Figure 15: A camouflaged master vehicle during track testing, some sensors are visible on the extern. 55

Figure 16: General ramp up of the development process through network milestones and general releases. 56

Figure 17: A VR test bench during a driving test. 59

Figure 18: The simulated environment during a reachability test61

Figure 19: Model of vehicle suspensions. 61

Figure 20: Aerodynamic simulation on a GT-R 50 62

Figure 21: Fluid dynamic analysis on an engine to study convective heat transfer. 62

Figure 22: Virtual crash test example. 63

Figure 23: Virtual structural analysis. 63

Figure 24: Simulation of production process at welding fixtures workstation. 1

Bibliography

.1: NI. "What Is Hardware-in-the-Loop (HIL)?" Last modified July 4, 2024. https://www.ni.com/en/solutions/transportation/hardware-in-the-loop/what-is-hardware-in-the-loop-.html.

.2: https://e-university.tu-sofia.bg/e-publ/files/8260_Review%20of%20hardware-in-the-loop%20-%20a%20hundred%20years%20p.pdf

.3 Mihalič, Franc, Mitja Truntič, and Alenka Hren. "Hardware-in-the-Loop Simulations: A Historical Overview of Engineering Challenges." *Electronics* 11, no. 15 (2022): 2462. https://doi.org/10.3390/electronics11152462.

.4: Gao, Paul, Hans-Werner Kaas, Detlev Mohr, and Dominik Wee. "Automotive Revolution – Perspective Towards 2030." *McKinsey & Company*, January 1, 2016. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/disruptive-trends-that-will-transform-the-auto-industry/de-DE.

.5: Heineke, Kersten, Philipp Kampshoff, and Timo Moller. "Spotlight on Mobility Trends." *McKinsey & Company*, March 12, 2024. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/spotlight-on-mobility-trends.

.6: Heineke, Kersten, Alexandre Menard, Freddie Sodergren, and Martin Wrulich. "Development in the Mobility Technology Ecosystem—How Can 5G Help?" *McKinsey & Company*, June 27, 2019. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/development-in-the-mobility-technology-ecosystem-how-can-5g-help.

.7: Alius. "Driving Innovation: Navigating the Future of Car Technologies." *Technology Org*, March 8, 2024.

https://www.technology.org/2024/03/08/driving-innovation-navigating-the-future-of-car-technologies/.

.8 Chiao, Derek, Kersten Heineke, Ani Kelkar, Martin Kellner, Elizabeth Scarinci, Dmitry Tolstinev, and Johannes Deichmann. "Autonomous Vehicles Moving Forward: Perspectives from Industry Leaders." *McKinsey & Company*, January 5, 2024. https://www.mckinsey.com/features/mckinsey-center-for-future-mobility/our-insights/autonomous-vehicles-moving-forward-perspectives-from-industry-leaders.

.9: International Organization for Standardization. "ISO/IEC 33000 Family." https://committee.iso.org/sites/jtc1sc7/home/projects/flagship-standards/isoiec-33000-family.html.

.10: International Organization for Standardization. 2011. "ISO 26262-1:2011 - Road Vehicles — Functional Safety — Part 1: Vocabulary." https://www.iso.org/standard/43464.html.

.11: McKinsey & Company. "Smartphones on Wheels: New Rules for Automotive-Product Development." *Automotive & Assembly*, October 24, 2022. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/smartphones-on-wheels-new-rules-for-automotive-product-development.

.12: https://d2368tcediwknr.cloudfront.net/bkm/magazin_2010_02_en/index.html#page_6

.13: "Italdesign. 'E/E Testing & Validation.' Italdesign. Last modified October 13, 2023. https://www.italdesign.it/services-electric-and-electronics/ee-integration-and-validation/.

.14: Škoda Storyboard. "Perfectly Proven Electronics." Last modified November 8, 2023. https://www.skoda-

storyboard.com/en/skoda-world/perfectly-proven-electronics/.

.15: "ZF: AI-in-the-Loop - dSPACE." dSPACE. Last modified October 11, 2024. https://www.dspace.com/en/inc/home/applicationfields/stories/zf-ai-in-the-loop.cfm.

.16: Vector Informatik GmbH. "Scenario-based Testing with OpenSCENARIO." https://www.vector.com/it/it/know-how/virtual-test-drives/openscenario/.

.17: "Vehicle-in-the-Loop Simulation." dSPACE. Last modified October 2024. https://www.dspace.com/en/inc/home/news/engineers-insights/blog-vehicle-in-the-loop-sim.cfm.

.18: "Italdesign." "ADAS & Autonomous Driving." Italdesign. Last modified June 13, 2022. https://www.italdesign.it/services-electric-and-electronics/adas-and-autonomous-driving/.

.19: 10 dSPACE. "Regular Testing of Driver Assistance Systems." Last modified October 11, 2024. https://www.dspace.com/en/inc/home/applicationfields/stories/regular-testing-of-driver-assi.cfm.

.20: "Facilities." CARS. Politecnico di Torino. https://www.cars.polito.it/it/it/facilities.

.21: "Italdesign. 'Prototypes and Mules.' Italdesign. Last modified December 1, 2021. https://www.italdesign.it/services-assembly-and-construction/prototypes-and-mules/.

.22: "Italdesign Concept Lab." *Italdesign*. https://www.italdesign.it/services-labs/concept-lab/.

.23: "Italdesign." "Chassis." Italdesign. https://www.italdesign.it/services-vehicle-and-product-development/chassis/.

.24: Italdesign. "Aerodynamics." Italdesign. Last modified October 19, 2023. https://www.italdesign.it/services-vehicle-and-product-development/aerodynamics/.

.25: "Italdesign." "Virtual Validation." Italdesign. Last modified November 16, 2021. https://www.italdesign.it/services-vehicle-and-product-development/virtual-validation/.

.26: "Italdesign. 'Product & Process Validation.' Italdesign. Last modified October 19, 2023. https://www.italdesign.it/services-vehicle-and-product-development/product-and-process-validation/.

Other sources are not of public domain.