



CHALMERS
UNIVERSITY OF TECHNOLOGY



Evaluation of hardware-in-the-loop framework for intelligent safety systems verification

Master's thesis in Automotive Engineering

JUNHUA CHANG
YICHANG LIU

MASTER'S THESIS 2018:21

Evaluation of hardware-in-the-loop framework for intelligent safety systems verification

JUNHUA CHANG
YICHANG LIU



Department of Mechanics and Maritime Sciences
Division of Vehicle Safety
Accident Prevention Group
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Evaluation of hardware-in-the-loop framework for intelligent safety systems verification

JUNHUA CHANG

YICHANG LIU

© JUNHUA CHANG, YICHANG LIU, 2018.

Supervisor: Siddhant Gupta, Volvo Car Corporation

Examiner: Jonas Bårgman, Department of Mechanics and Maritime Sciences Chalmers
University of Technology

Master's Thesis 2018:21

Department of Mechanics and Maritime Sciences

Division of Vehicle Safety

Accident Prevention Group

Chalmers University of Technology

SE-412 96 Gothenburg

Telephone +46 31 772 1000

Cover: volvo car

Printed by Chalmers University of Technology
Gothenburg, Sweden 2018

Evaluation of hardware-in-the-loop framework for intelligent safety systems verification

JUNHUA CHANG

YICHANG LIU

Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

Traditional Active Safety Function Verification methods involve comprehensive physical testing which is both time consuming and expensive. The current trend in the automotive industry is moving towards validation and verification of these Active Safety Functions in the *Computer Aided Environment* (CAE). Based on the components under test, the CAE Environments can be classified as *Model-in-the-loop* (MIL), *Software-in-the-loop* (SIL), *Hardware-in-the-loop* (HIL) and *Vehicle-in-the-loop* (VIL).

The thesis aims at evaluation of the hardware-in-loop framework which constitutes a setup of virtual vehicle, scenario generation methods, simulation on real time platform and data logging tools for the development and verification of intelligent safety systems.

The HIL framework and tool-chain in focus here consists of Intelligent Safety Manager, constituting the active safety function logic, Camera and Radar (sensor fusion) units as hardware components under test along with a virtual vehicle modelled in different off-the-shelf CAE Environments. The evaluation of these CAE environments includes setting up the HIL framework with *Virtual Test Drive* (VTD) for Scenario Generation and chassis model from VEDINA, replacing the chassis model with a Functional Mock-up unit from Dymola and ultimately modifying the complete framework to replace with IPG CarMaker.

The evaluation of above mentioned framework(s) includes determination of important components and performance parameters of tool chains. Evaluation were carried out with ISO vehicle dynamic attribute tests and open-loop driver assistance use cases. The result illustrated that the IPG CarMaker HIL framework gave reasonably good for collision avoidance function verification because of the accurate dynamics response for steering and braking, while the VTD & VeDyna HIL framework had the big advantage in the verification of camera based active safety function because of its high 3D rendering quality.

Keywords: Hardware in Loop (HIL), Model in Loop (MIL), Software in Loop (SIL), CAE Environment, Vehicle Model, steering model, intelligent safety systems, Benchmark, driver assistance, IPG CarMaker, VTD, VEDINA, FMU-Dymola

Acknowledgements

This Master's Thesis has been conducted during the spring of 2017 by Junhua Chang, student at the M.Sc. program Automotive Engineering and Yichang Liu, student at the M.Sc. program Embedded Electronic System Design at Chalmers University of Technology, Göteborg, Sweden. To a great extent the research was performed at the Driver Assist & Active Safety CAE group of Volvo Cars Corporation, Göteborg, Sweden. Resources provided by Volvo Cars in form of technical equipment and test data made it possible to carry out this thesis. The outcome is hopefully a contributing step towards an improved research and development process within the department and company.

We would like to thank our supervisor at Volvo Car Corporation, Siddhant Gupta for his continuous support and valuable guidance which has been the key for this study to be carried out. Special acknowledgement and thanks to our examiner at the Chalmers University of Technology, Jonas Bårgman for making the plans and revising report for us. Yury Tarakanov, manager of Driver Assistance Active Safety CAE, Volvo Car Corporation is very supportive and encouraging. We would also like to extend our gratitude towards Richard Ferngren, Nadeem Afzal, Rasmita Samantary and Alexey Gurov, Volvo Car Corporation, for their great help and all cooperation during our thesis.

Junhua Chang and Yichang Liu, Göteborg, June 2017

Nomenclature

Abbreviation

<i>HIL</i>	Hardware-in-the-Loop
<i>SIL</i>	Software-in-the-Loop
<i>MIL</i>	Model-in-the-Loop
<i>VIL</i>	Vehicle-in-the-Loop
<i>ECU</i>	Electronics control unit
<i>ECM</i>	Engine control Module
<i>ADAS</i>	Advanced Driver-Assistance Systems
<i>IHU</i>	Infotainment Head Unit
<i>crc</i>	Constact Radius
<i>lssc</i>	Low-G Swept Steer
<i>occ</i>	On Center
<i>sinc</i>	Sine with Dwell
<i>hssc</i>	Low-G Swept Steer
<i>frc</i>	Frequency Response
<i>hssc</i>	High-G Swept Steer
<i>CAN</i>	Controller Area Network
<i>dof</i>	Degree of Freedom
<i>VTD</i>	Virtual Test Drive

Matlab variables & Simulink Signals

<i>SWA</i>	Steering wheel angle
<i>SWT</i>	Steering wheel Torque
<i>Ay</i>	Lateral Acceleration
<i>kph</i>	Kilometers Per Hour
<i>Vx</i>	Longitudinal Speed
<i>Ra</i>	Roll Angle
<i>Tbt</i>	Torsion bar torque
<i>ELA</i>	Emergency Lane Assist system

<i>ELKA</i>	Emergency Lane Keeping Assist
<i>LKA</i>	Lane Keeping Assist
<i>BLIS</i>	Blind Spot Information System
Physics Constants	
<i>g</i>	Gravitational Constant

Contents

List of Figures	xiii
List of Tables	xv
1 Introduction	1
1.1 Objective	1
1.2 Environment Frameworks for Virtual Verification of Active Safety . .	2
1.2.1 Model in the loop	2
1.2.2 Software in the loop	2
1.2.3 Hardware in the loop	3
1.3 Virtual Verification of Active Safety Functions	4
1.3.1 Emergency Lane Keeping Aid	4
1.4 Benchmarking Methodology	5
1.4.1 Purpose	6
1.4.2 Different methods	6
1.4.3 Benchmarking Process	6
1.5 Report Outline	8
2 Virtual Vehicle Setup in Model-in-Loop Environment	9
2.1 Modelling of Virtual Vehicle in MIL	9
2.1.1 Vehicle model in IPG CarMaker	10
2.1.2 Default vehicle model in SPAS environment	10
2.1.3 FMU-Dymola chassis model in SPAS environment	12
2.1.4 Steering system	13
2.1.4.1 The Mechanical Module	13
2.1.4.2 The Power Assistance Module	14
2.2 Tuning and Validating of the Virtual Vehicle	14
2.2.1 Vehicle Dynamics validation tests	14
2.2.2 Tuning and verification Process	15
3 Virtual Vehicle Setup in Hardware-in-loop Environment	21
3.1 Modelling of Virtual Vehicle in HIL	21
3.1.1 Tool chain 1 : VTD & VeDyna	22
3.1.2 Tool chain 2 : IPG CarMaker	23
3.1.3 Tool chain 3 : FMU-Dymola Chassis model	24
3.1.4 Vehicle Communication Network	25
3.2 Operation and Calibration Process of HIL Setup	25

3.3	Tuning and Validating of the Virtual Vehicle	28
3.3.1	Vehicle Dynamics Validation tests	28
3.3.2	Active Safety Function Verification - Emergency Lane Keeping Assist	29
3.4	Evaluation of Toolchains	31
4	Discussion	35
5	Conclusion	39
6	Future Scope	41
	Bibliography	43
A	Appendix A Simulation Plots on MIL for Carmaker VS FMU VS Invehicle Log	I

List of Figures

1.1	V-Model [1]	3
1.2	Examples of the Scenarios of ELKA function	5
1.3	Benchmarking Test Methodology	7
2.1	Architecture In MIL	9
2.2	Carmaker Model In MIL	10
2.3	Vehicle Handling Model	10
2.4	FMU-Dymola chassis model with driver steering controller	12
2.5	Steering System	13
2.6	Simulation results of <i>On Center</i> test manoeuvre	16
2.7	Simulation results of <i>Sine Shape with Dwell</i> test manoeuvre	18
2.8	Simulation results of <i>High-G Swept Steer</i> test manoeuvre	18
2.9	Simulation results of <i>Low-G Swept Steer</i> test manoeuvre	19
2.10	Simulation results of <i>Constant Radius</i> test manoeuvre	19
3.1	Model in the loop to Hardware in the loop	21
3.2	VTD & VeDyna (current HIL setup)	22
3.3	CarMaker HIL Setup with external Steering Model	23
3.4	Steering System Work Flow	24
3.5	FMU-Dymola Chassis model in HIL (modified HIL setup)	24
3.6	The instrument panel of ControlDesk	26
3.7	HIL setup including different sensor models [1]	27
3.8	Simulation results of <i>On Center</i> test manoeuvre	28
3.9	Simulation results of <i>Sine Shape with Dwell</i> test manoeuvre	29
3.10	Open-loop simulation instead of close-loop	30
3.11	CarMaker simulation results for elka function	30
3.12	image from the CarMaker vs. image from the camera	31
3.13	Evaluation results of three tool chains	33

List of Tables

2.1	Vehicle Dynamics Simulation Result	17
A.1	Results Catalogue A	II
A.2	Results Catalogue B	II
A.3	Results Catalogue C	II
A.4	Results Catalogue D	III
A.5	Results Catalogue E	III
A.6	Results Catalogue F	IV

1

Introduction

To achieve high safety ratings from independent organisations such as EuroNCAP[2], vehicle's *Electronic Stability Control* (ESC) systems and dynamic behaviour are evaluated against their provided test-protocols. For car industries, running these evaluations on prototype cars brings down the development speed and is a large cost. Thus, these tests are first conducted in *Computer Aided Environment* (CAE). The CAE's examples are *Hardware in the Loop*(HIL) [3], *Model in the Loop* (MIL), *Software in the Loop* (SIL) and *Vehicle in the Loop* (VIL). In HIL the tool chain is composed of various hardware, software, model blocks including the tools for measurement, logging, maneuver and environment creation etc. The setup provides to its customers the flexibility of testing and verifying a particular Active Safety function before it is tested in the prototype vehicle.

1.1 Objective

The main objective of this thesis work is to benchmark different toolchains in HIL environment. The architectures of these HIL setups are more profoundly explained in Section 3.1, but the main part of the work is summarized as following.

A basic HIL test bench setup consists of vehicle (dynamics) model, traffic environment (dynamic and static), and driver model (which controls the test vehicle according to previous specified scenario), sensors, actuators and controllers (eg. Intelligent safety manager) can be model/software/hardware. Since MIL/SIL and HIL testing performed in the company uses different CAE environments, it will be more efficient and necessary to use a common scenario description to shorten development cycles. The purpose of this thesis is to analyze available tool chains and speed up the testing process.

The currently employed HIL setup consists of VTD & VeDyna (tools which provide test scenario, traffic environment and vehicle model) as softwares, *Intelligent Safety Domain Manager* and Radar & Camera as Hardwares. In the second step, IPG Car-Maker will be used to replace VTD & VeDyna in HIL setup to reduce the number of tools, lower the cost and make the system easier to handle. At the last step, a FMU-based (Functional Mockup Unit implements a tool independent standard for model exchange and co-simulation.) vehicle dynamics model along with the actuator models from supplier will be deployed to modify the default setup.

After the toolchains are identified, operation process of HIL setup has to be completely understood and documented. The capabilities of the tool-chain are determined in this process. Then the benchmark tests will be specified and all these tests are based on both the objective and subjective criteria. Each test component will be assigned with a weight factor and all test results will be combined into a matrix.

1.2 Environment Frameworks for Virtual Verification of Active Safety

The validation phases in the development of automotive intelligent safety system can be summarised as different validation methods with short feedback circle.

1.2.1 Model in the loop

Model in the loop (MIL): In the early stage of development process, active safety function is verified and validated in closed-loop simulations with the vehicle dynamics, sensors, actuators and traffic scenario, all as mathematical models. Virtual bus signals are implemented to connect the ECU models and plant models. The MIL stage is applied to verify and validate the basic implementation architecture and the design of active safety function in an environment without the hardware[4].

At the beginning of this project, simulation was also run in MIL environment by using *Simulation Platform for Active Safety* (SPAS) and *IPG CarMaker*. SPAS is an in-house simulation platform for active safety at Volvo Cars, developed by using Simulink. It has a specific road generation tool. And the scenario is defined in an excel sheet including the parameters of objects and corresponding velocity profile (longitudinal movement) and offset profile (lateral movement). After initialisation, the scenario with corresponding roads information is loaded to the workspace and then running the simulation with other Simulink models (driver, chassis, actuators and sensors).

IPG CarMaker is a complete simulation environment for virtual test driving, provided by IPG Automotive GmbH. This virtual vehicle environment consists of virtual vehicle, virtual road and virtual driver. It is easy to create realistic test scenarios with this simulation environment. In addition, the Simulink platform of CarMaker allows external models to be integrated into the system.

1.2.2 Software in the loop

Software-in-the-loop (SIL) is applied to test the functionality of the software code compiled from the models after Model-in-the-loop provides satisfied results.[5] The hardware components are not involved and virtual I/O interface is implemented.

When the steering system or braking system from the supplier are integrated in SPAS and CarMaker, the environment then became software in the loop, since the steering or steering system has already been compiled into C code to test steering or braking function.

1.2.3 Hardware in the loop

Hardware-in-the-Loop (HIL)[6] involves the physical target electronics units and physical communication bus. Data is loaded to the ECUs to perform the real-time simulation. In contrast to road vehicle test, HIL is able to test complete feedback control systems since it provides a repeatable and stable laboratory environment without disturbances from other unrelated systems.

In HIL environment, *Toolchain*[7] is defined as a set of software and hardware tools that are chained together constructing a testing loop. Software tools create software models, define test scenarios and traffic environment, run simulation as well as log the simulation results.

The tools involved in different toolchains with this thesis are: *VTD*, *VeDyna*, *IPG CarMaker* and *Functional Mockup Unit* (FMU). Parameters such as accuracy of the simulation environment against the vehicle measurements, complexity and user friendliness of the toolchains and efficiency with benefit analysis is the scope of evaluation. Subcomponents of the vehicle model, like sensor models, steering model and chassis model along with traffic and road information in the CAE Environment will also be assessed, for example their fidelity and accuracy.

All mentioned ‘in-the-loop’ simulation techniques are increasingly being used for design and validation of *Advanced Driver Assistant System* (ADAS) to meet the requirement of the fast, flexible and repeatable tests. All these different testing methods construct a V-model 1.1 [1], which represents all stages of development process.

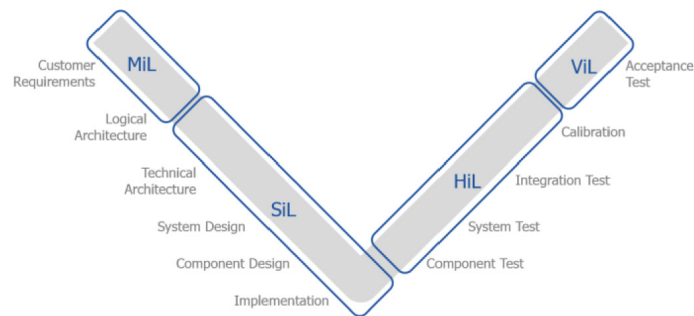


Figure 1.1: V-Model [1]

1.3 Virtual Verification of Active Safety Functions

Active Safety Functions [8], which in our case are *Emergency Lane Keeping Assist*(ELKA)[9], will be used as benchmarking function to be analysed across the toolchains. As for a comparing reference, in-vehicle validation tests were carried out according to Euro-NCAP defined protocol. The difference between ELA and traditional lane departure warning system[10] is that ELKA aims at providing an assist steering wheel torque to prevent dangerous lane departure, instead of forcing the vehicle keeping the lane.

ELA detects the lane information by using a camera mounted behind the windscreen to take images of the lane mark, road edge, road signs. Objects like oncoming or faster moving vehicles are detected by the radar sensors. By sensor fusion, the lane position and objectives' relative condition would be extracted and filtered. Then the data would be sent to the ECUs to evaluate and decide providing assist steering wheel torque.

1.3.1 Emergency Lane Keeping Aid

Emergency Lane Keeping Assist (ELKA) Function, is a new active safety function built under the development of other conventional lane guidance active safety functions like Lane-Keeping-Assistant and Blind-Spot-Warning. It has a sensor module to detect the surrounding traffic and a threat assessment module that analyses and manages the triggers of this function. There have been quite a few discussions, research and implementations about Lane-Keeping-Assistant functions both in the academic and industry. Vehicles with Lane-Keeping-Assistant function are available on the market for years, implementation methods such as giving the driver an audio warning, or applying an opposite direction steering wheel torque to keep the vehicle in the lane etc.[11] However, there are two main problems, one is false alarm issue that the signal shown to the driver always pop up as long as the vehicle steps on lane markings. Although some warnings could be turned off if indicators are used, people have to be very careful for this issue which has a bad influence for driving experience. Another problem is misuse. In some cases, people tend to use this system as a more convenient system rather than a safety system. Since the vehicle can be kept in the lane automatically, the driver could release his hands and let other things distract him/herself. These problems can be reduced or solved by the ELKA function as it is using a threat assessment module. This module is to evaluate the risk level of scenarios and determine the time of the intervention of ELKA function, which means the ELKA function only interferes under the really dangerous circumstances.

There are some of the occasions when ELKA Function are activated. One typical version is that the driver is going to do a lane change or the vehicle is about to enter an adjacent lane. 1.2[8]. 'H' indicates host car. In the first subsection of this figure, there will be no intervention of ELKA, because there is no traffic in the adjacent lane which is considered no danger. But in the second, third and fourth pictures,

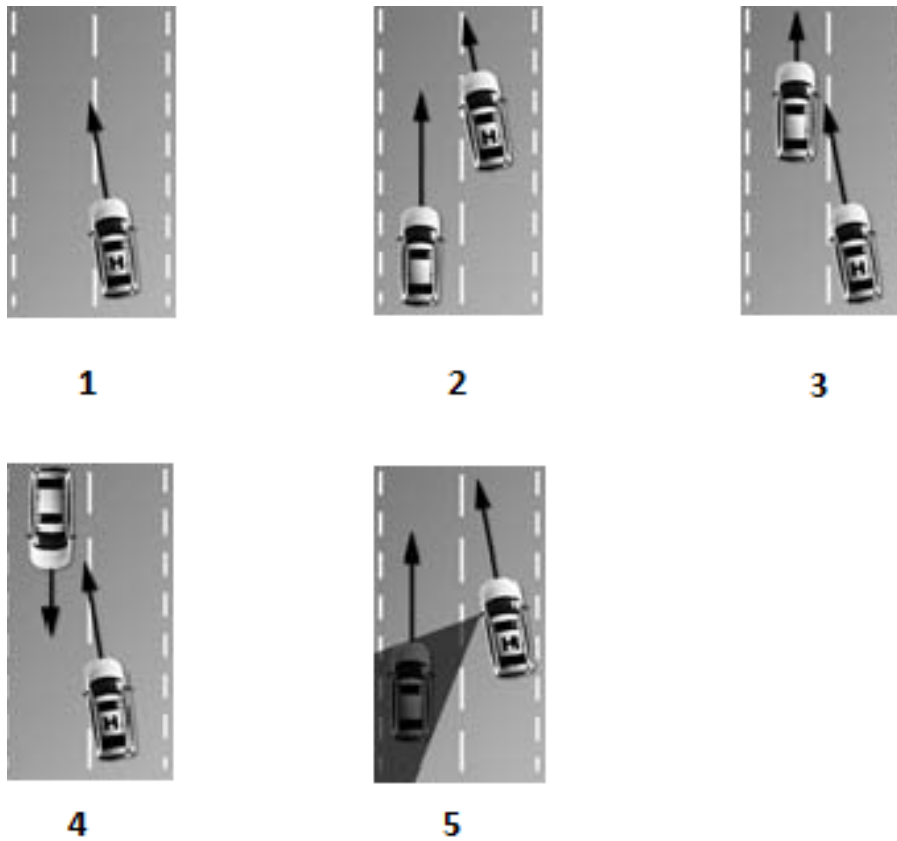


Figure 1.2: Examples of the Scenarios of ELKA function
[8]

there is another object coming from behind with a faster speed, going in front with a lower speed, or approaching towards the host car in front in the adjacent lane, these are when assessment module will consider dangerous by analysing the data sent from sensors and invoke ELKA, to take sufficient actions to avoid collision. Scenario in picture five is a specific circumstance of picture two, which illustrates that blind spot of the rear-view mirror would be the cause of this situation. [8] [9] For the reason that ELKA function is a quite complicated function that involves high fidelity vehicle dynamics, steering system actuation and both camera and radar as sensor module to trigger the function, it is then selected as the benchmark function.

1.4 Benchmarking Methodology

Benchmarking was firstly initiated by the Xerox Corporation for comparing the products and service from the competitors and then implementing the observed 'best practices' to improve the business. The ultimate goal of the benchmarking was not to compare the performance of different tools but rather an analysis of capabilities that is gained or lost by using this tool.[12]

1.4.1 Purpose

Apparently, benchmarking is needed to open up many more ideas to try to achieve the best performance in the industry and academic. There are several reasons accounting for it.

1. *Assessment and evaluation* of current employed products or working process can be achieved by benchmarking in relation to other organisations and learn from each other. It is always a good way of evaluate by comparing with others then seek for a chance to optimise.[13]
2. Benchmarking of a product or design process can simply broad the horizon. By applying the same design principles in another way could help *advancing the original design process* and looking for solutions for obstacles had before.[13]
3. Another reason of benchmarking is to clearly set a goal which has been approved to be successful. A company or a designer could improve his performance efficiently by learning some new and creative approaches.[13]
4. At some point, when a company is very familiar and skillful at its way of designing and working, which could then lead to a miss of better design solution or fading away of the ambition of workers. By looking through other companies and do comparison may enhance and overcome this problem.[12]

1.4.2 Different methods

There are different methods with individual process of benchmarking.

1. The basic type of benchmarking is *internal benchmarking*. It is one of the basic and starting benchmarking method that applied. The main purpose of doing internal benchmarking is to decide the internal performance standards of an organization. [13]
2. *Competitive Benchmarking* is another type which could compare within products and process of work of different competitors. This could help learning from others and knowing the situation of itselfes. [13]
3. There are another external benchmarking method which is *Functional or Industry Benchmarking*. The main object of this type is between collaborators sharing the same goal or technology, which makes them more willing to contribute more to the benchmarking. [13]
4. *Process or Generic Benchmarking* Benchmarking focusing on the best working process. [13]

1.4.3 Benchmarking Process

The major focus of the benchmarking in the thesis is to compare the IPG CarMaker tool chain and the current employed tool chain, VTD & VeDyna, against the real data collected in the real vehicle test. Some of the important components and capabilities of the tool chains will be evaluated. The benchmarking process can be summarised as following:

The first stage is the preparation process which consists of four steps.

- Firstly, the evaluated tool chains have to be identified.
- Then operation process of HIL setup has to be completely understood and documented. This includes the architecture and working procedures for simulations of the tool chains. After having a deeper view of all these things, the components and performance parameters of the tool chain, which will be tested and evaluated, can be identified.
- The benchmark tests will be defined for testing specific components or parameters.
- Last but not least, the data of the real vehicle test should be collected to be reference data during the benchmarking process.

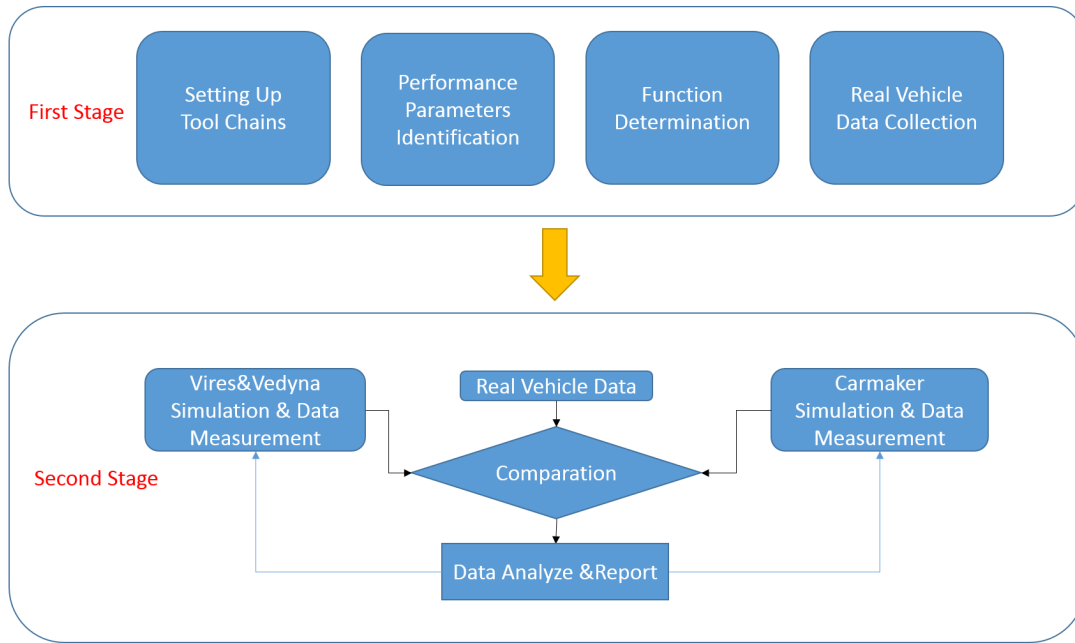


Figure 1.3: Benchmarking Test Methodology

In the second stage, several tasks are taken in parallel and series.

- Firstly, simulations with different tool chains on the basis of different functions are performed. Simulation results and data are stored for further analysis.
- Secondly, simulation results and data are stored for comparing with in-vehicle log.
- Thirdly, evaluating the components and parameters of tool chains and writing a benchmarking report. This step could help analyzing the pros and cons of the tool chains and what could be done to make the tool chains to have a better performance.
- Then finally, testing the promotion doing the simulation again.

1.5 Report Outline

The remainder of this thesis report is structured as follows. In chapter 1, the theoretical background and objective is introduced. Different frameworks for virtual verification and their validation phases in this thesis project is explained.

In chapter 2, details about setup of virtual vehicle in Model-in-Loop environment is described. This includes introduction of vehicle models in different CAE environments (IPG CarMaker, SPAS, FMU-Dymola library), steering system and its integration, vehicle dynamics validation tests, tuning and verification process of these three virtual vehicle setups.

Chapter 3 is to detail the setup of the virtual vehicle in the Hardware-in-Loop environment. Modelling of virtual vehicle in MIL environments is explained, along with some theory about vehicle communication protocol. All information related to operation and calibration of HIL setup is also included in this chapter. Tuning the setup is done by running the simulation to validate the emergency lane keeping assist function.

In chapter 4, how benchmarking matrix designed is explained and results are discussed.

Finally, chapter 5 concludes the thesis, summarises the results and brings forth possible further work in chapter 6 that can be optimised and improved to finalized the evaluation of different frameworks.

2

Virtual Vehicle Setup in Model-in-Loop Environment

In the Model in loop simulation phase, simulations run in *IPG CarMaker* and SPAS (in-house simulation environment for MIL) environment. In order to do comparison, the default chassis model in SPAS environment is also simulated. The FMU-dymola chassis model supplied by *Modelon*, is from *Complete HIL* team in the company. It is used to replace VeDyna chassis model and would be used in group *Active Safety HIL* in the next phase. According to information given by *Modelon*, there is no validation work of this model done. Therefore, it is necessary to do validation and research in the MIL environment in order to have a good understanding on it before it is used on the HIL test bench later.

2.1 Modelling of Virtual Vehicle in MIL

A general software architecture in MIL environment is shown in the figure 2.1. The vehicle model used in this thesis project is of Volvo XC60. The vehicle model is an multi-body system which consists of several subsystems, for example, chassis, powertrain, engine, aerodynamics and sensors. In the Model-in-the-loop environment, the entire vehicle is a mathematical model without any physical components involved. The parametrisation of the vehicle model is out of the scope of this thesis. Vehicle dynamics validation tests as stated in section 2.2.1 have been carried out to give good reference.

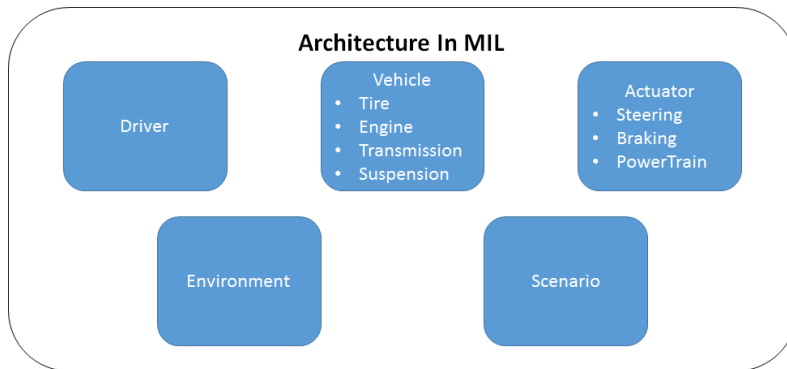


Figure 2.1: Architecture In MIL

2.1.1 Vehicle model in IPG CarMaker

The CarMaker MIL environment, which in this project comes from vehicle dynamics department at Volvo Cars, has a already validated vehicle data set with an external steering model. The setup of a CarMaker vehicle and actuator models in Simulink is shown in figure 2.2. In order to describe the vehicle body system, differential equations of different parts, like kinematics, forces and torque, kinetics and coordinates integration, are used to model the vehicle motion. Through the 'Vehicle Data Set' in the GUI, it is really flexible to set all the parameters related to the vehicle body.

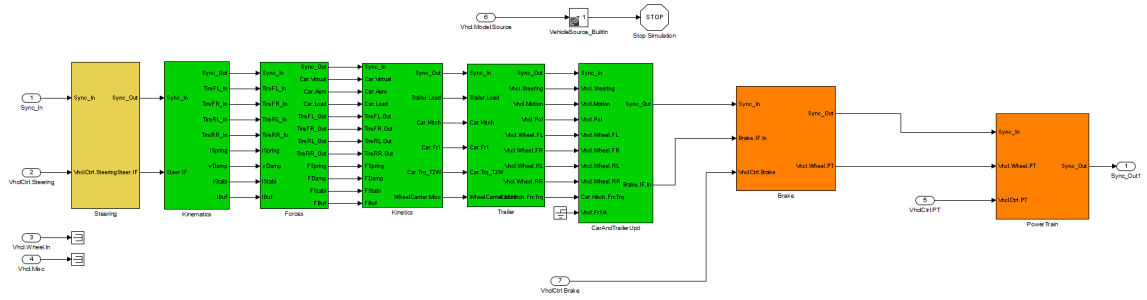


Figure 2.2: Carmaker Model In MIL

2.1.2 Default vehicle model in SPAS environment

The default chassis model in the SPAS environment, which is an in-house simulation software for verification and validation of active safety function for Model in the loop stage, has 7-dof (degree of freedom): longitudinal, lateral, yaw motions and the spin of each wheels. The chassis model is quite simple but also sufficient in a 2D simulation environment for early stage of verification. The figure 2.3 shows a similar model.

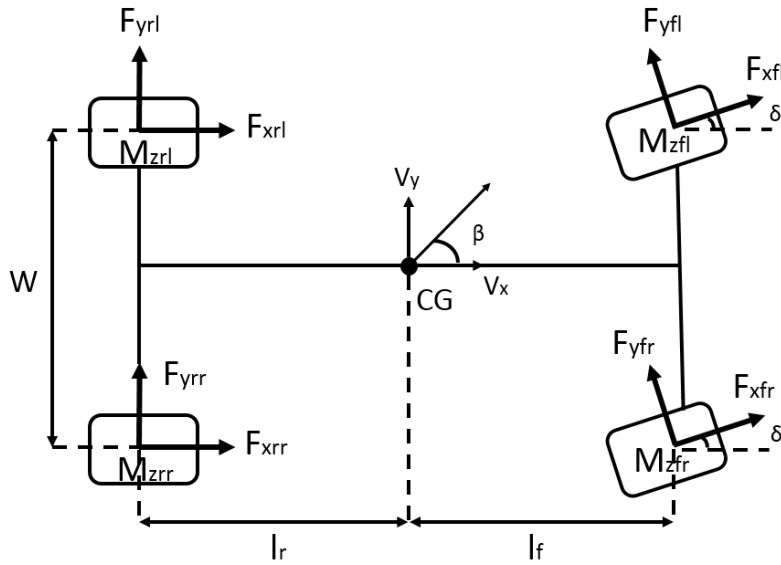


Figure 2.3: Vehicle Handling Model

Inertial acceleration at the center of gravity in the x direction is calculated as:

$$a_x = \dot{v}_x - v_y\dot{\psi} \quad (2.1)$$

The equation of forces in longitudinal direction:

$$m_i * a_x = F_{xfl} \cos \delta - F_{yfl} \sin \delta + F_{xfr} \cos \delta - F_{yfr} \sin \delta + F_{xlr} + F_{xrr} \quad (2.2)$$

Inertial acceleration at the center of gravity in the y direction is calculated as:

$$a_y = \dot{v}_y - v_x\dot{\psi} \quad (2.3)$$

The equation of forces in lateral direction:

$$m_i * a_y = F_{xfl} \sin \delta + F_{yfl} \cos \delta + F_{xfr} \sin \delta + F_{yfr} \cos \delta + F_{yrl} + F_{yrr} \quad (2.4)$$

The lateral side slip angle at the front tire rear tire can be defined as:

$$\alpha_f = \arctan\left(\frac{v_y + l_f\dot{\psi}}{v_x}\right) \quad (2.5)$$

$$\alpha_r = \arctan\left(\frac{v_y - l_r\dot{\psi}}{v_x}\right) \quad (2.6)$$

The speed of front tire is defined as:

$$v_{wxf} = V_{tr} \cos \alpha_f \quad (2.7)$$

And the longitudinal velocity of the front tire can be given as:

$$V_{tf} = \sqrt{(v_y + l_f\dot{\psi})^2 + v_x^2} \quad (2.8)$$

The speed of rear tire is defined as:

$$v_{wxr} = V_{tr} \cos \alpha_r \quad (2.9)$$

And the longitudinal velocity of the rear tire can be given as:

$$V_{tr} = \sqrt{(v_y - l_r\dot{\psi})^2 + v_x^2} \quad (2.10)$$

The longitudinal slip of the front tire is calculated under the braking condition:

$$S_{af} = \frac{v_{wxf} - w_f R_w}{v_{wxf}} \quad (2.11)$$

The longitudinal slip of the rear tire under braking:

$$S_{ar} = \frac{v_{wxr} - w_r R_w}{v_{wxr}} \quad (2.12)$$

Yaw motion is the left to right motion of the car on its vertical axis and is given by the following equation, where M_z is the aligning moment.

$$\begin{aligned}
 J_z \dot{\psi} = & \left[-\frac{W}{2} F_{xfl} \cos \delta + \frac{W}{2} F_{xfr} \cos \delta - \frac{W}{2} F_{xfl} + \frac{W}{2} F_{xrr} + \frac{W}{2} F_{yfl} \sin \delta - \frac{W}{2} F_{yfr} \sin \delta \right. \\
 & + l_f F_{xfl} \sin \delta + l_f F_{yfl} \cos \delta + l_f F_{xfr} \sin \delta + l_f F_{yfr} \cos \delta - l_r F_{yrl} - l_r F_{yrr} \\
 & \left. + M_{zfl} + M_{zfr} + M_{zrl} + M_{zrr} \right]
 \end{aligned} \tag{2.13}$$

The spin of the wheel is represented by the wheel angular speed. The following equation describes the torque balance for each wheel:

$$\begin{aligned}
 L_w \dot{w}_{fl} &= T_{dfl} - T_{bfl} - F_{xfl} R_w \\
 L_w \dot{w}_{fr} &= T_{dfr} - T_{bfr} - F_{xfr} R_w \\
 L_w \dot{w}_{rl} &= T_{drl} - T_{brl} - F_{xrl} R_w \\
 L_w \dot{w}_{rr} &= T_{drr} - T_{brr} - F_{xrr} R_w
 \end{aligned} \tag{2.14}$$

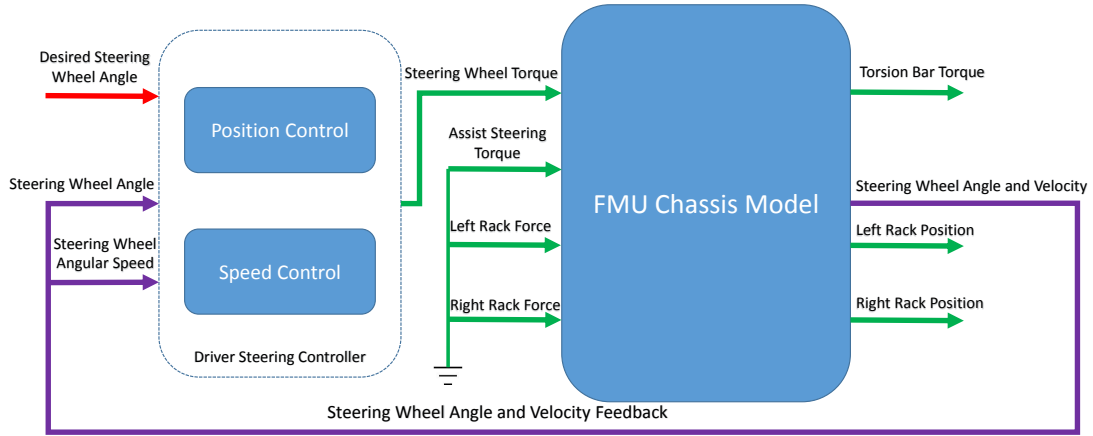


Figure 2.4: FMU-Dymola chassis model with driver steering controller

2.1.3 FMU-Dymola chassis model in SPAS environment

The validation of FMU-Dymola chassis model was also done in SPAS environment. The default chassis model along with the steering model was replaced by the FMU chassis model. The interface of these two models were quite different. While the default chassis model required steering wheel angle as input, the inputs of FMU model were the steering Wheel torque and assist torque along with the rack forces. The outputs with respect to steering were the steering wheel angle and velocity, torsion bar torque and rack position. The steering model (mechanical part) and tire

model were already integrated inside FMU model. Since the interface of the chassis model was changed, which means steering wheel torque should be the input, a driver steering controller from complete HIL team was implemented to transfer the steering wheel angle into steering wheel torque. In the meantime, assist torque (no assistant force from electrical system) and rack forces were grounded.

2.1.4 Steering system

The EPS steering system used in this thesis is shown in figure 2.5. It consists of mechanical part, which includes steering wheel, steering column, torsion bar, rack and pinion, and electrical part including servo motor, controller and sensors. This model is provided by the supplier and is compiled into a S-function model (An S-function is a computer language description of a Simulink block written in MATLAB). Therefore, it is compatible in both MIL (SPAS and CarMaker) and HIL (CarMaker) environment, but there is no chance for us to look inside the model to do some modification.

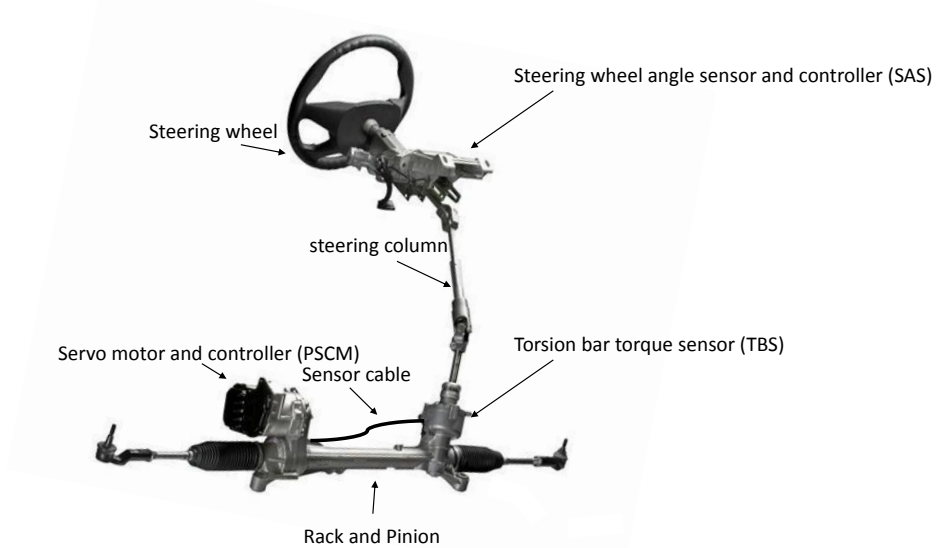


Figure 2.5: Steering System

2.1.4.1 The Mechanical Module

The mechanical module of the steering system supports the input to be both steering angle and steering torque. Usually the steering angle output from the driver (or driver model) gives a stable signal to the steering system. On the contrast, signal of steering torque is quite noisy. However, when the vehicle is taken over by the intelligent safety manager, which constitutes the active safety function logic, the system has to shift to steer by torque to ignore the signals from the driver and to enable the intervention by the ECU.

2.1.4.2 The Power Assistance Module

The power assistance module represents the electrical (or can be hydraulic) system that reduce the steering effort from the driver. In this case, it is an electrical motor that apply the assistant force directly to the rack with a belt and ball nut mechanism.

2.2 Tuning and Validating of the Virtual Vehicle

In this section some representative results regarding vehicle dynamics simulation in MIL environment are presented and analysed. All information about the test manoeuvres and plots of simulation results can be found in Appendix A.

Steering wheel angle and longitudinal velocity from real test data are extracted and input into the CarMaker and SPAS to simulate with different chassis models. Lateral acceleration, yaw rate along with the roll angle are the parameters compared with the real test data and give an overview of the performance of chassis models related to lateral movement.

2.2.1 Vehicle Dynamics validation tests

In order to have a brief overview of the handling characteristic of vehicle model with response to the steering command, the following test manoeuvres are selected.

- ***On Centre Weave***

This test is used to characterise the steering performance of a vehicle at a low steering frequency and low to moderate lateral accelerations. The vehicle is given sinus steering as input to produce a series of head angle changes and achieve certain lateral acceleration. The tests are conducted at different velocities and lateral accelerations. The Results catalogue can be found in the table A.1 in Appendix A. [14]

- ***Sine Shape with Dwell***

This test is used to objectively determine a vehicle's transient response behaviour (yaw stability and response, ESP performance) under closely controlled test conditions similar to lane change manoeuvres in real traffic. The initial speed is 85-90 km/h with released throttle and neutral gear then the vehicle coasts to 80km per hour. At the moment, the driver begins to use a steer pattern of a sine with dwell shape. The shape is defined as a sinusoidal wave with frequency of 0.7 Hz, and the steering pauses for 0.5 seconds after reaching the second peak. Sine Shape with Dwell is to evaluate the vehicle lateral stability and responsiveness. [15]

- ***High-G and Low-G Swept Steering***

These tests are Steady-state cornering test. The vehicle makes a turn adequate enough to achieve the maximum lateral acceleration with a constant speed. The steering wheel angle input increase linearly. For high-g swept steer test, 80kph is used for the speed while lss test use 80 and 120kph. The maximum

lateral acceleration of high-g swept steer is 0.88g, for lss are 0.49g for 80kph and 0.57g for 120kph. It is used to evaluate the steering response at large and small steering wheel angle. The tests will obtain and analyse steady state directional and roll dynamics, both left and right, to evaluate the road holding capability of different chassis setup due to different body and road variants.[16]

- **Frequency Response**

This test is used for correlating roll dynamics, yaw dynamics and general dynamic response characteristics. The vehicle performs a continuous sinusoidal steering wheel angle sweep input with frequency decreasing with constant speed. During the test, different constant speed (80 and 120kph) with different steering wheel angle amplitude are tested respectively. [16]

- **Constant Radius**

The Constant Radius is a Steady State Cornering test means that the vehicle drives on a path with a constant radius for 40 meters after a straight acceleration to 3.505 m/s. In our test, the vehicle is fed in increasing longitudinal speed up to above 18 m/s and steering wheel angle up to above 2.4 rad/s within 127 seconds. The test are both executed for left and right turn. The test mainly focus on data acquisition of the yaw rate as well as the SWT and lateral acceleration, and the behaviours of steering system.[17]

2.2.2 Tuning and verification Process

Figure 2.6-2.10 show some representative results of *On Center*, *Sine Shape with Dwell*, *High-G Swept Steer*, *Low-G Swept Steer*, and *Constant Radius* test manoeuvres. In all the plots, steering wheel angle is the input of FMU-Dymola chassis model and it maintains a high co-relation with the in-vehicle measurements (for CarMaker and default chassis model, steering wheel angle is the input as well as the longitudinal speed).

The yaw rate of FMU-model is close to the measurement and almost has the same performance as that of CarMaker: the biggest difference between in-vehicle log and simulation result is 9.2% in *On Center* scenario (Figure 2.6, bottom-left panel); 11.9% in *Sine Shape with Dwell* (Figure 2.7, bottom-left panel); 5.7% in *High-G Swept Steer* (Figure 2.8, bottom-left panel); 5.2% in *Low-G Swept Steer* (Figure 2.9, bottom-left panel); and 2.2% in *Constant Radius* (Figure 2.10, bottom-left panel). By comparison, the default chassis model in SPAS has a bigger bias in yaw rate. (For example, the deviation is 78.9% in *Low-G Swept Steer*.)

Another important parameter is roll angle, which also shows good performance of FMU-model. The biggest deviation between in-vehicle log and simulation result is only 4% in *Sine Shape with Dwell* (Figure 2.7, bottom-right panel) and 5% in *Low-G Swept Steer* (Figure 2.9, bottom-right panel).

On the other hand, the curve of lateral acceleration has a offset comparing with measurements and CarMaker. This offset is more obvious in the *Sine Shape with Dwell*, *High-G Swept Steer* and *Low-G Swept Steer* manoeuvres, which can be seen

2. Virtual Vehicle Setup in Model-in-Loop Environment

in figure 2.7-2.8. The lateral acceleration is in disagreement because of being calculated in different inertial frame.

In addition, the simulation result of the torsion bar is much higher than the measurement from real test log. The reason is likely that the effect of servo force is also applied to the torsion bar since the steering model, including the rack, is already integrated inside the FMU chassis model.

The result of FMU in *Constant Radius* test is unsatisfying. The plots of lateral acceleration and roll angle has a sine shape. This behaviour can be explained that the track of vehicle cannot maintain constant during the simulation. The change of the radius is exactly changed with a sine shape and that affects the lateral acceleration and roll angle afterwards.

Table 2.1 shows more details about yaw rate and roll angle result from the simulation. Since the lateral acceleration and torsion bar torque is quite noisy from the measurement (even after filtering), it is hard to do some statistic analysis. In addition, more thoroughly vehicle dynamics validation will be performed during the further development of FMU-model.

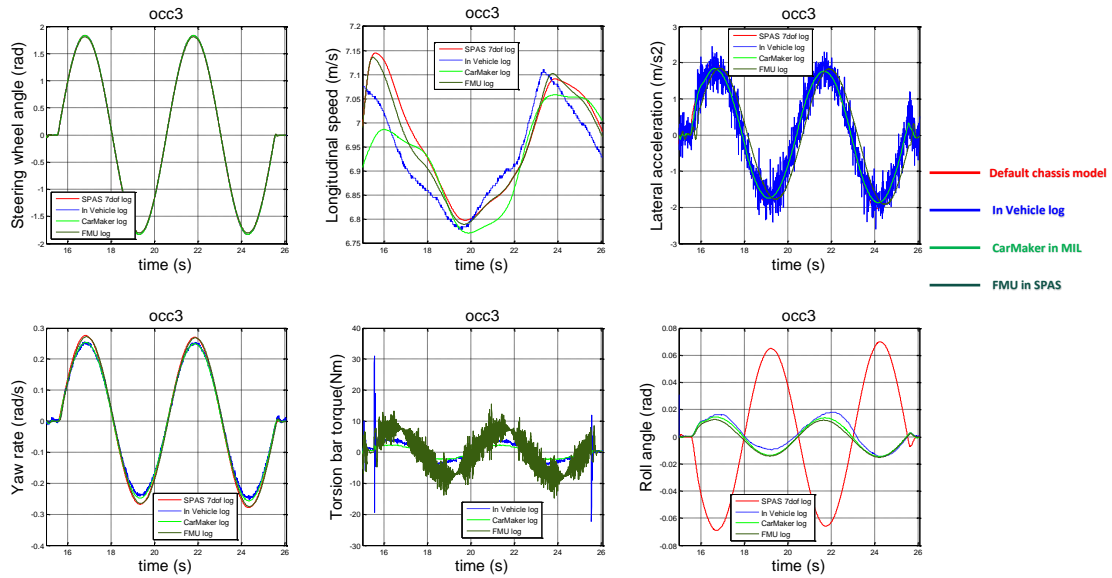


Figure 2.6: Simulation results of *On Center* test manoeuvre

Table 2.1: Vehicle Dynamics Simulation Result

Yaw (rad/s)	Rate	<i>Biggest deviation between in-vehicle log and simulation result</i>				
		<i>On Center</i>	<i>Sine Shape with Dwell</i>	<i>High-G Swept Steer</i>	<i>Low-G Swept Steer</i>	<i>Constant Radius</i>
<i>FMU in SPAS</i>		9.20%	11.90%	5.70%	5.20%	2.20%
<i>Default chassis model in SPAS</i>		8%	33.30%	42.80%	78.90%	44.40%
<i>CarMaker in MIL</i>		«1%	11.90%	5.70%	10.50%	<2%

Roll (rad)	Angle	<i>Biggest deviation between in-vehicle log and simulation result</i>				
		<i>On Center</i>	<i>Sine Shape with Dwell</i>	<i>High-G Swept Steer</i>	<i>Low-G Swept Steer</i>	<i>Constant Radius</i>
<i>FMU in SPAS</i>		11%	4%	71.50%	5%	x
<i>Default chassis model in SPAS</i>		x	x	x	x	x
<i>CarMaker in MIL</i>		5.60%	«4%	1.40%	<1%	2.50%

2. Virtual Vehicle Setup in Model-in-Loop Environment

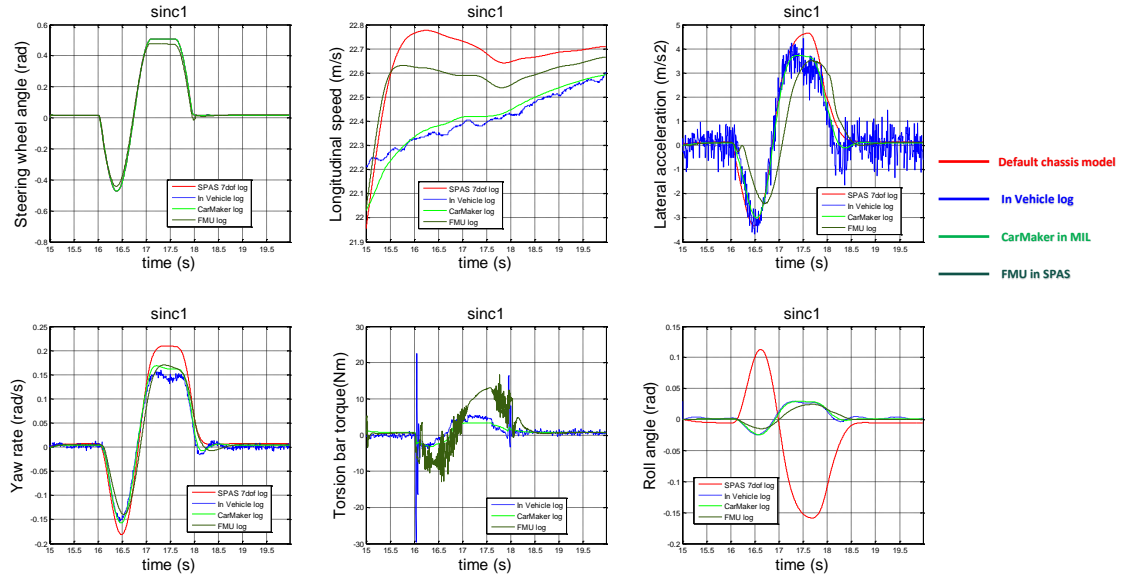


Figure 2.7: Simulation results of *Sine Shape with Dwell* test manoeuvre

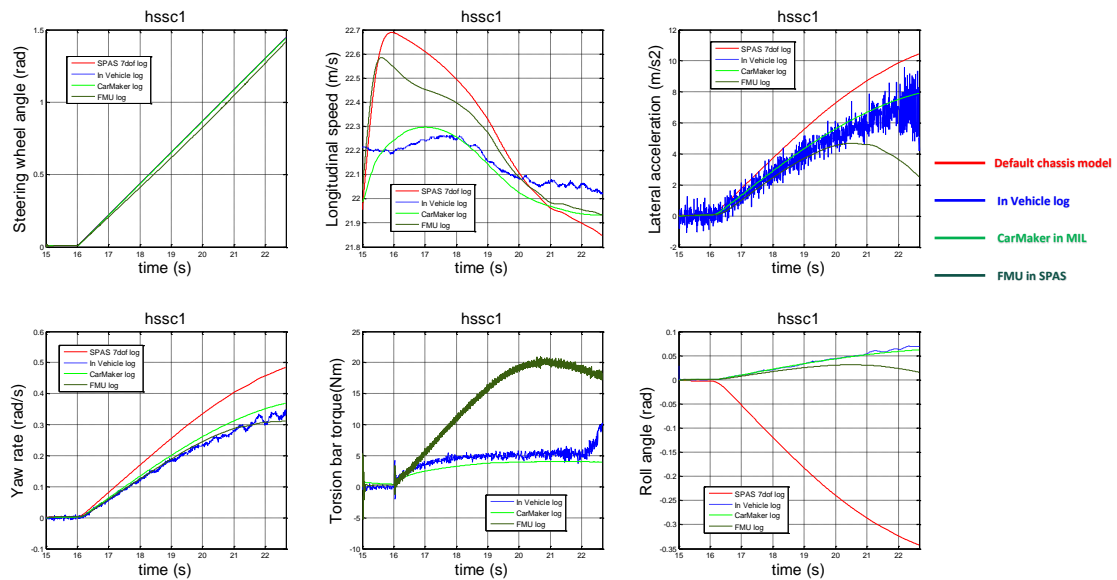


Figure 2.8: Simulation results of *High-G Swept Steer* test manoeuvre

2. Virtual Vehicle Setup in Model-in-Loop Environment

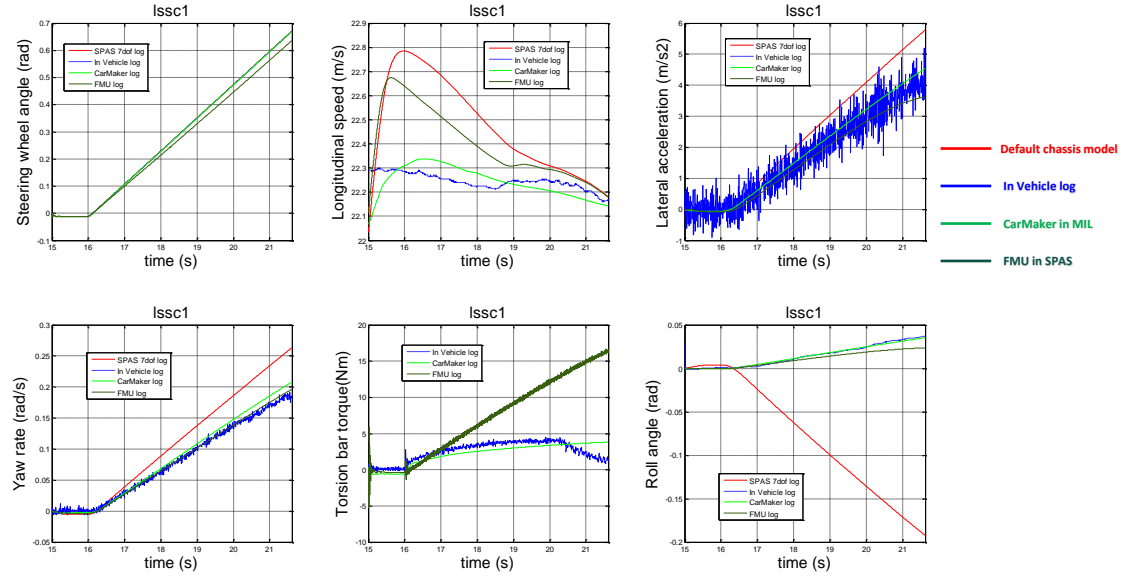


Figure 2.9: Simulation results of *Low-G Swept Steer* test manoeuvre

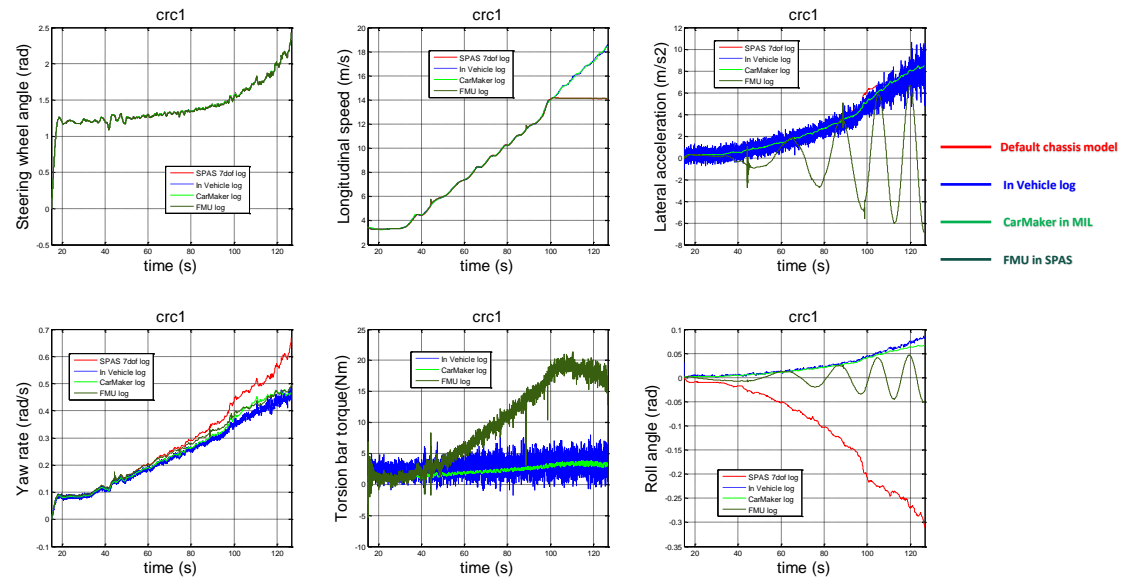


Figure 2.10: Simulation results of *Constant Radius* test manoeuvre

3

Virtual Vehicle Setup in Hardware-in-loop Environment

When designing verification methods and scenarios are decided, active safety function would be verified and validated with a mathematical models of the entire vehicle in the simulation, this is model in the loop step, which is already described in section 3.1. Next step is to include the physical target electronics units in the test environment. As showed in figure 3.1, with the implementation of hardware, simulation would be loaded to the ECUs to perform real-time simulation. This is called ECU HIL or component HIL. The horizontal arrow indicates hardware in the loop testing the outcome of the model in the loop and gives the feedback.

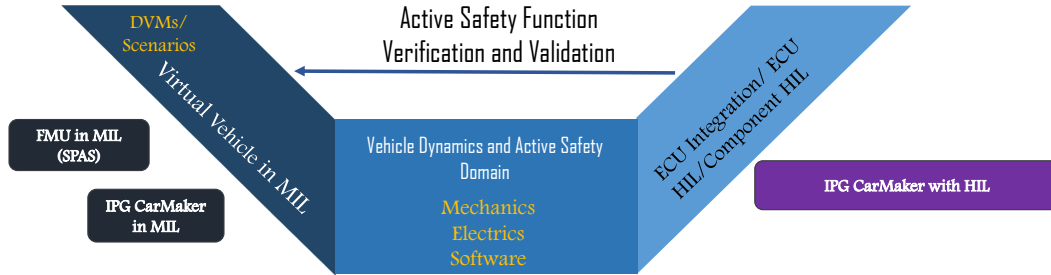


Figure 3.1: Model in the loop to Hardware in the loop

3.1 Modelling of Virtual Vehicle in HIL

In order to have a good understanding of HIL testing, much effort would be put in analysing the structure or architecture of currently employed HIL setup at the beginning. Then the default setup would be modified by other tool chains along with a benefit analysis.

3.1.1 Tool chain 1 : VTD & VeDyna

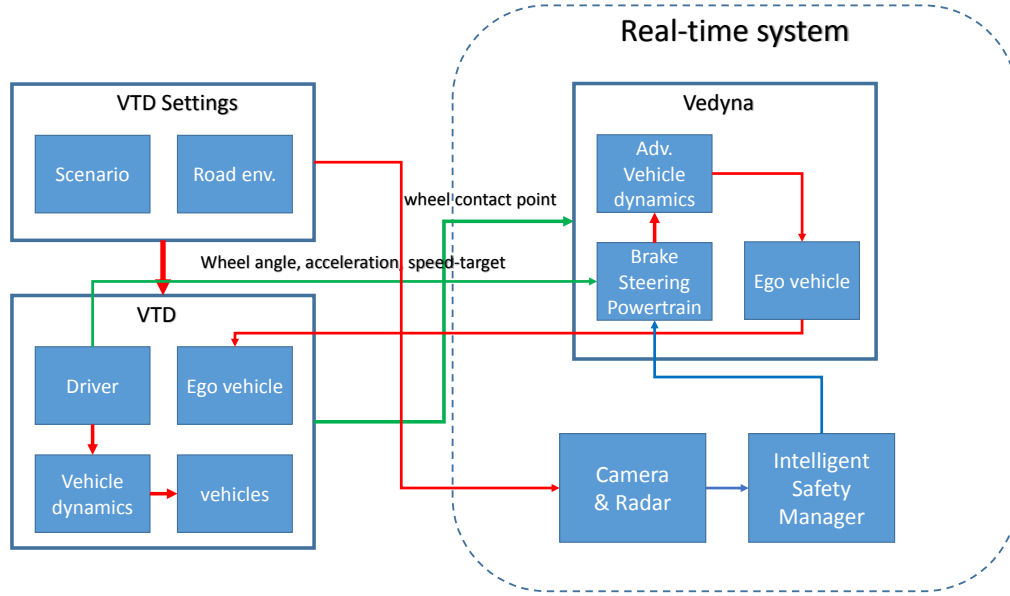


Figure 3.2: VTD & VeDyna (current HIL setup)

In currently employed HIL setup, the traffic environment(road, traffic objects and signs) and testing scenario(ACC, LKA, AEB...) are created or defined in *VIREs Virtual Test Drive* (VTD) while the vehicle dynamics model and the actuator models(steering system, braking system and powertrain) are created in VeDyna. The reason for using these tools is that VTD is a good software to create scenarios, import road information from real data and make professional 3D rendering. On the other hand, VeDyna provides a complete and detailed vehicle model that gives realistic characteristic of vehicle dynamics.

VTD is a standalone program run on a separate linux computer whereas Vedyna is executed inside a Simulink model uploaded to the real time platform. A solution of integration has to be implemented in the future to realize the co-simulation. In the current solution, the road and driver model is disabled in Vedyna. Therefore, the road information and driver manoeuvres have to be streamed in real time from VTD to Vedyna for vehicle dynamics calculation. This is done by:

1. Fetch the road data (geometry, friction...) of the Ego vehicle's wheels' contact points from VTD and stream into Vedyna.
2. Driver manoeuvres including steering wheel angle and throttle/brake are also feed into Vedyna from VTD driver model.
3. After Vedyna calculated the vehicle dynamics, wheels' speed, yaw, global position of the vehicle and other parameters will be feedback to VTD to realize the close loop control.
4. Since the reference coordinate systems are different in VTD (middle of rear axle) and Vedyna (middle of front axle), a transformation is necessary for the sending and receiving data.

An overview of the current setup is shown in figure 3.2.

3.1.2 Tool chain 2 : IPG CarMaker

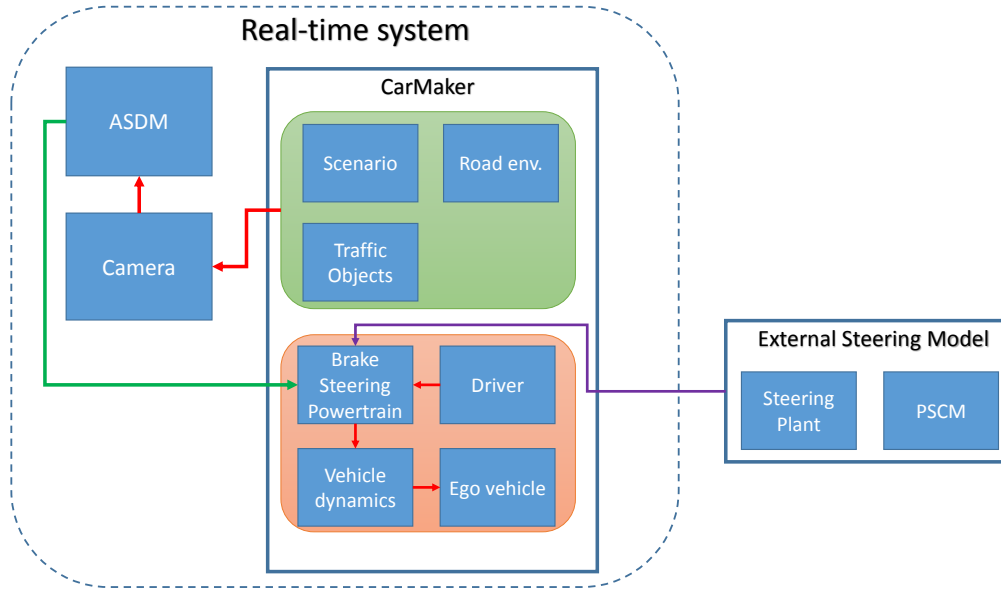


Figure 3.3: CarMaker HIL Setup with external Steering Model

IPG CarMaker is a virtual driving test tool supporting many applications from ECU and Subsystem testing to system networks testing for vehicles. The main advantage of using CarMaker is the possibility of reducing the involved tool chains since testing scenario, traffic environment along with all the models are created in CarMaker. Data does not have to be streamed from VTD to VeDyna, which means there will be no latency and packet loss in the communication from linux computer to the real time platform. Another advantage is that CarMaker is used in the vehicle dynamics department at Volvo Cars. Using experiences and some sort of technical support might be available. Modified HIL setup is shown in figure 3.3.

When emergency lane keeping assist is activated, in order to take control of the vehicle and make it turn back to the original lane to avoid the collision, an external steering model, which is the same as used in MIL environment, has to be used to replace the ideal steering model in CarMaker. The whole steering system, which is shown in figure 3.4 consists of steering plant and Steering Control Unit as ECU. When the steering wheel angle signal from driver is received by the steering column, it would be transferred into pinion angle and pinion angular velocity and forward to torsion bar. After that, the manual gear would transfer the torsion bar torque into mechanical force and finally feed it to the rack. On the other side, Steering Control Unit would receive the signals from Vehicle Dynamics Manager along with the torsion bar velocity to calculate the servo force. This servo force, produced by assist motor, would also be another force in the rack to balance the movement.

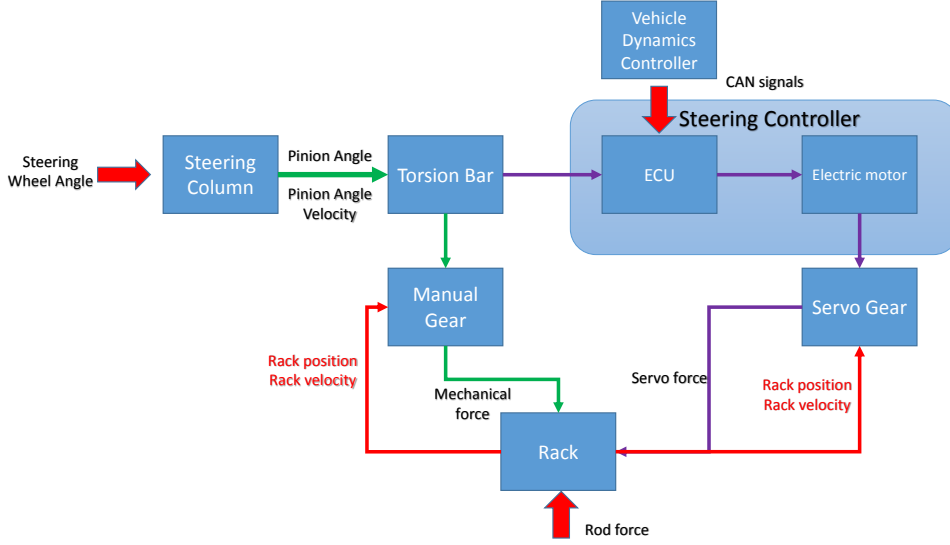


Figure 3.4: Steering System Work Flow

3.1.3 Tool chain 3 : FMU-Dymola Chassis model

The third tool chain is to use FMU-Dymola Chassis model (which we have already integrated and tested in our MIL environment) to replace vedyna vehicle model. The *Functional Mockup Interface* (FMI) [18], which is initiated and organised by Daimler AG, is a general standard designed to meet the need of integrating multiple tools and formats for simulation. It offers an possibility for standardised, tool-independent model exchange in collaborative system. Which means that the vehicle model will be easy to implemented in our HIL system and license free (reduce a lot of cost).

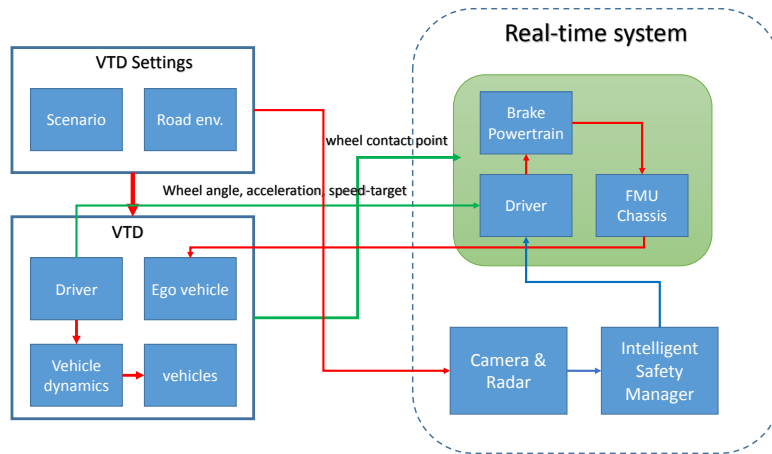


Figure 3.5: FMU-Dymola Chassis model in HIL (modified HIL setup)

In our case, a FMU-Dymola chassis model will be provided by the supplier. Our

mission is to use a FMU-Dymola chassis model to replace the VeDyna vehicle model in the Simulink. The communication structure will be the same as the primary one. The modified HIL framework with the FMU-based chassis model will be compared and evaluated with the other setups.

3.1.4 Vehicle Communication Network

The vehicle communication network, which is also called *vehicle bus*, is a central network in the vehicle to connect all the electronic control units. Each electronic control unit is responsible for one or more components inside the vehicle and communicate with other ECUs by exchanging data. Protocols including *Controller Area Network (CAN)*, *Local Interconnect Network (LIN)*, *Flexray* and others are used to deliver the message among the vehicle network. The I/O communication architecture of our test setup is based on CAN and Flexray.

Controller Area Network (CAN) is a serial communication protocol which provides a high level security when transferring the message[19]. The speed of message delivery is up to 1 Mbit/s. The standard format of message transfer is controlled by four types of frame: Data Frame, Remote Frame, Error Frame and Overload Frame. Each CAN data frame has a unique identifier and transmit different data.

FlexRay Protocol is a serial communication protocol which is faster than CAN with a maximum speed of 10 Mbit/s. The standard frame format consists of three segments: the header segment, the payload segment and the trailer segment.[20]

3.2 Operation and Calibration Process of HIL Setup

Before the determining the test criteria, all the steps in building the model, preparation of the HIL setup, performing of the simulation and analysis the results have to be listed in detail to have a good understanding of the requirements of the whole system. Thus, at the beginning, the focus was on operating and getting familiar with the currently employed HIL setup. The process of compiling and preparing the model was documented as well, after the task was turned to modifying the HIL setup by replacing *VTD & VeDyna* with *IPG CarMaker*. All these steps are explained as follow:

1. As is mentioned before, in HIL environments all the electronics control units are communicating with each other through CAN and Flexray protocols. The signals in the system are not virtual anymore like in MIL environment. In our case, the hardware, which includes Intelligent Safety Manager, camera and radar, has the physical connection with the real time platform and exchanges data with the model through CAN and FlexRay channels. Thus, in the first step the IO communication architecture of the model has to be built and modified to make sure the vehicle bus work properly. If the real-time controllers in the model are not corresponding with the physical harness of the real time platform or some default values of signals are not set correctly, some errors

3. Virtual Vehicle Setup in Hardware-in-loop Environment

might pop out and CAN or FlexRay signals from the model cannot be received or transmitted.

The complexity of communication architecture would affect the simulation performance and the time consumed for building the model. Therefore, it is a good point to consider when doing the benchmark later.

2. Integrating external (steering and braking) models is another important step after the basic structure of the whole model is set up. In active safety verification, Intelligent Safety Manager would send signals to the actuators to control the vehicle after it is triggered. For example, in lane keeping assistant scenario, requested pinion angle would be sent by Intelligent Safety Manager to steering control units to avoid collision. Thus, the steering system has to be able to shift to steer by torque mode to ignore the driver input and enable the intervention of assistant torque. In addition, the accuracy of the active safety function is highly dependent on actuator models.

However, the steering and braking models are black boxes which cannot be modified since they are provided by the supplier and have already been verified and validated. Thus, the major mission is to plug the electronic modules, (which are not real control units but models), into the communication network and make them be able to communicate with other modules.

3. After setting up all the models and debugging the signal system, the *Simulink* model, including the libraries of the database and the external models, should be compiled to generate and execute C code. The generated source code will then be loaded to the dspace real time platform for real-time application via *ControlDesk*. The consumed time for the compiling and building process is deeply affected by the complexity of the model.



Figure 3.6: The instrument panel of ControlDesk

4. The *ControlDesk* provides the possibility of control the simulation in real-time via a virtual instrument panel, which is shown in figure 3.6. It also offers a

inside view of the driver action and the features of the instrument panel can be modified by the user. When a new project (model) is loaded, the parameters corresponding to the buttons on the instrument panel have to be set to correct value. This process is to switch the virtual model (Intelligent Safety Manager & Camera & Radar) to hardware manually, but it can also be done automatically by Python script.

5. When the model is loaded and the communication network is checked without problem, next step is to calibrate the camera. In the current test bench, real sensors (a camera and a radar) are included and connected to the Intelligent Safety Manager. For the radar sensor, the virtual data of simulated objects gets injected into hardware as processed sensor data. The radar does not capture the physical signals. On the other hand, a screen is used to emit light and inject the rendering image directly into the physical channel of the camera. Thus, the position of the camera and the screen has to be set and adjusted to make sure the image injection process close enough to the reality.

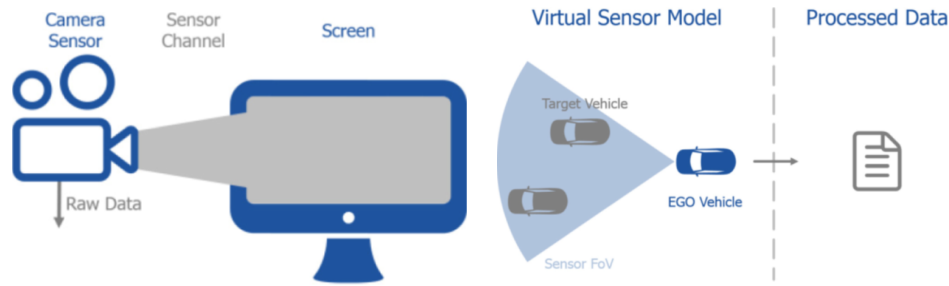


Figure 3.7: HIL setup including different sensor models [1]

6. The test scenario has to be prepared for driver assistance simulation. Firstly, the road network is generated based on the real road. This can be done by using a GUI or a script. Then host vehicle as well as the traffic environment are defined in scenario creator. Last step is to parameterize the driver model and manoeuvres. Since in our case, the simulation results of the active safety system will be against in-vehicle test. All the process mentioned before will be carried out based on the in-vehicle log.
7. Before performing the simulation, some signals from the infotainment head unit are used to trigger the specific active safety function. However, in our test bench, the infotainment head unit is a virtual model, which means all these signals have to be set manually via the *ControlDesk*.
8. GUI of the tools (VTD or CarMaker) is used to select the test scenario and start the simulation while the *ControlDesk* is able to monitor or track the change of the signals. In the current HIL setup, result or data of the simulation must be measured by the *ControlDesk* before it is stored.

3.3 Tuning and Validating of the Virtual Vehicle

After the CarMaker HIL setup was been prepared, vehicle dynamics validation tests were performed at the first step, in order to make sure that the integration and setting of the models were correct. Then driver assistant scenario (emergency lane keeping assist) was performed. This step was required all the physical communication properly working in the setup.

3.3.1 Vehicle Dynamics Validation tests

Vehicle dynamics validation tests were used to benchmark the vehicle model and steering model of the tool chains. For CarMaker, the same vehicle model and steering model were used in both MIL and HIL environment. Since we had already validated their performance in MIL environment, right now the mission was to integrate the same model in HIL setup. One of the difference needs to be mentioned is that the powertain model was not compatible for dspace environment. Therefore, a ideal powertrain model was used instead. The result is shown in the following figures.

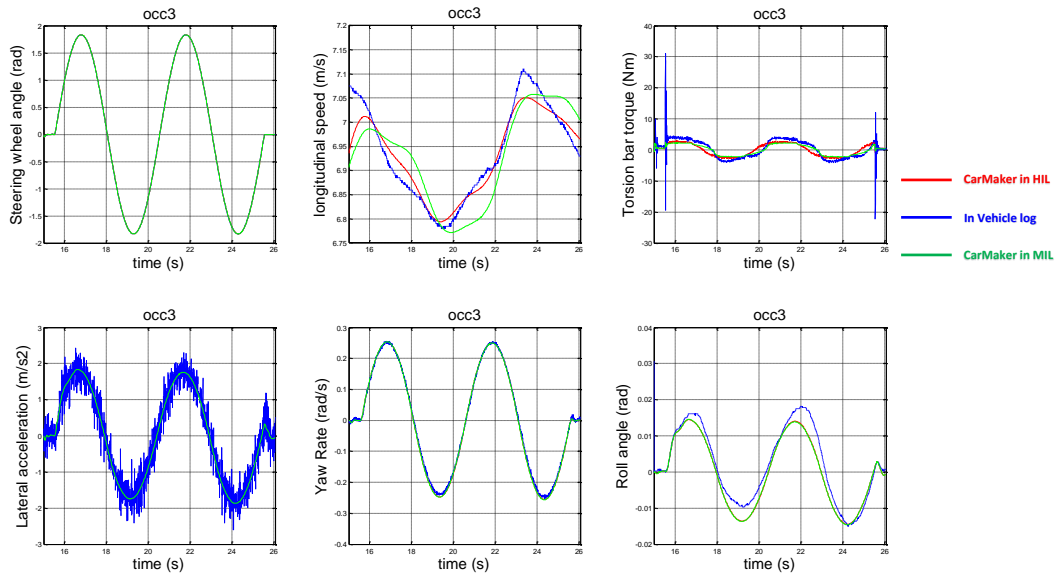


Figure 3.8: Simulation results of *On Center* test manoeuvre

Figure 3.8 shows the *On Center* test manoeuvre. Steering wheel angle (Figure 3.8, top-left panel) and the longitudinal speed (Figure 3.8, top-middle panel) are the input signals while lateral acceleration, yaw rate (Figure 3.8, bottom-left panel) and roll angle (Figure 3.8, bottom-right panel) are the parameters that are analyzed. Torsion bar torque (Figure 3.8, bottom-middle panel) is also evaluated as well. The results tested in HIL environment maintain a high co-relation with results in MIL environment and in-vehicle measurements.

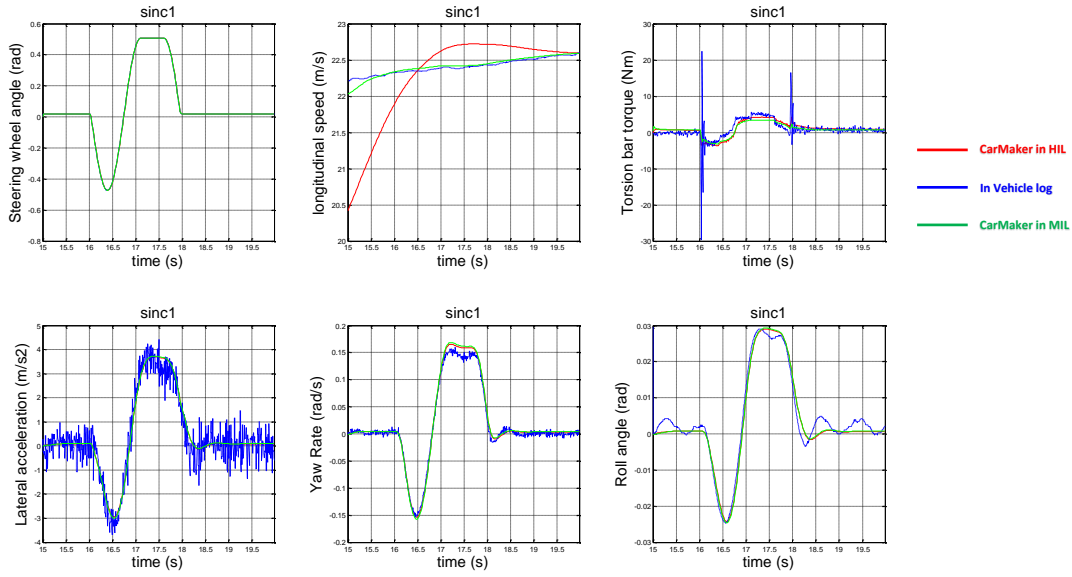


Figure 3.9: Simulation results of *Sine Shape with Dwell* test manoeuvre

Figure 3.9 shows the *Sine Shape with Dwell* test manoeuvre. All the parameters show the good performance of the simulation except that the longitudinal speed has a big offset when accelerating. This is caused by using ideal powertrain model as mentioned before. It cannot provide realistic acceleration ability.

3.3.2 Active Safety Function Verification - Emergency Lane Keeping Assist

Driver assistance system simulation met with a lot of trouble, from both hardware and software. The most critical issue was that the active safety function could not be triggered since Infotainment Head Unit and Vehicle Dynamics Manager models provided some wrong signals. The communication network required every nodes inside the system to work properly or send the correct message. Therefore, the close-loop system involving hardware was not possible anymore. In order to continue the project and do some evaluation, the close-loop driver assistance system simulation was changed to open-loop simulation. virtual signals from real in-vehicle test log, instead of physical signals from Intelligent Safety Manager, was used to control the steering.

Since the scenario was created based on the real in-vehicle test log, the speed of host vehicle and traffic object should be close to the reality. The data injection point was decided based on the distance of the host vehicle and traffic object.

Another issue was that in VTD & VeDyna setup, it was not possible to use the external steering model. The communication between linux PC and real time platform

3. Virtual Vehicle Setup in Hardware-in-loop Environment

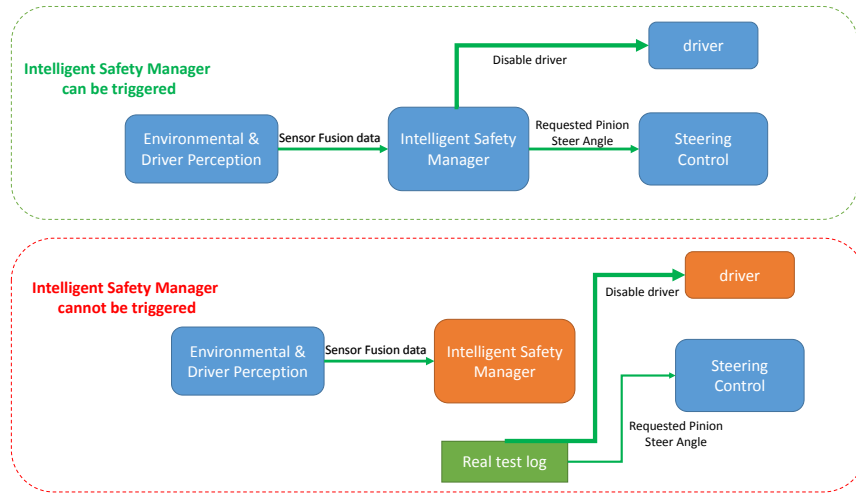


Figure 3.10: Open-loop simulation instead of close-loop

had critical latency issue due to co-simulation between various platform. Therefore, no results were available for default test setup. In addition, there was no point to use ideal steering model to do the comparison.

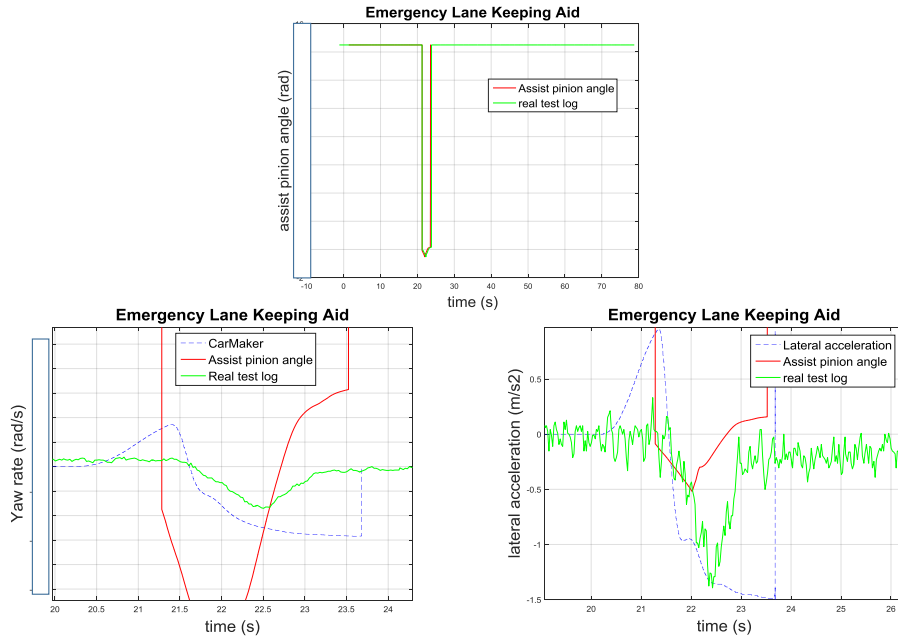


Figure 3.11: CarMaker simulation results for elka function

Figure 3.11 shows the simulation result of CarMaker HIL setup. The lateral acceleration and yaw rate shows an acceptable performance since no tuning for the function has been done for this setup. The scenario definition is not perfectly close to the reality and trigger time is also selected quite roughly.

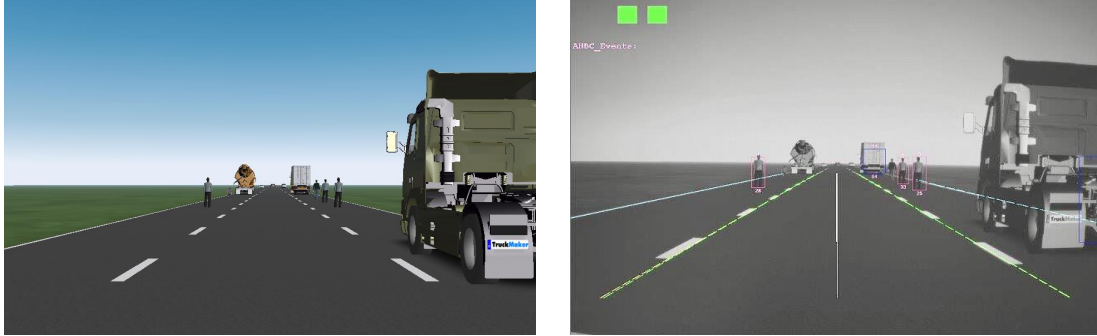


Figure 3.12: image from the CarMaker vs. image from the camera

3D rendering quality is a very important part that affects the image processing performance of the camera. The algorithm using the camera is able to detect the objects forward and also figure out the type, size and position of different objects. In the current situation, the hardware was not able to trigger the function and gives the output. However, the camera was still capture the image and did the image processing. Figure 3.12 shows that the trucks, pedestrians and road marks could be easily and correctly processed by the sensor. Therefore, it can be conclude that CarMaker provides sufficient 3D rendering.

3.4 Evaluation of Toolchains

As was discussed in the chapter 1.4.3, parameters and weight factors should be determined for the benchmarking. All the details about the selection is explained as follows:

There were several parameters selected for creating the benchmarking matrix. The behaviour and data of *Chassis model* and *Steering model* could be measured by running simulations on ISO tests as discussed at section 2.2.1. In this case, On Centre Weave and Sine Shape with Dwell test were selected since they evaluated the critical aspect of the models' performance.

Besides, it is always important to evaluate the *cost factor* of implementing the toolchains. It is obvious that companies and researchers spare no effort to minimise the cost during working process. These parameters can be obtained in the process of setting up, operating and calibrating HIL frameworks as well as documenting. The cost of hardware is not concerned since there will be no change in hardware for

different tool chains.

In order to save time and work load, an efficient and user friendly *Automation on Simulation Master* is a critical factor which should not be neglected. *Platform for Integration* illustrates how convenient it is to exchange the models or reconfigure the test setup.

A very important property is that the tool is easy for user to handle and user friendly, *User friendless* can be accessed during operating and validating process. One way to evaluate the quality of tutorials and documents. *Architecture Complexity* can be approached when the frameworks were built and all the models/hardware were integrated.

The *Fidelity* of the software is always on the top of the list. Especially in the active safety function verification and validation process, deviation has to be eliminated or decreased into an acceptable level. Latency, synchronisation problems, bias of the vehicle dynamics and steering controller are the focus.

Data Storage and Post-processing are also a conclusive parameters of the performance of tool chains.

Because hardware Camera is involved as sensors for Active Safety Functions, test case was simulated by using a separate screen to stream image to the camera. In order to create a virtual reality that is close to the real environment, variety of virtual objects and 3D-rendering quality is crucial. *Virtual Scenario Environment Creation* and *Visualisation* become an important factors.

Execution speed including compiling and building time measures the complexity of the frameworks.

Weight factors and score were set to evaluate the tool chains. parameters were assigned a numerical score to show the importance. Tool chains were assigned a numerical score to show the evaluation in different parameters. Some of them were based on objective criteria. For instance, financial cost of the license. But there were also some are based on subjective criteria. For instance, User friendliness.

We set the weight factors in four levels. The highest level is *Very important*, representing the key factor for active safety function analysis and verification in hardware-in-loop environment; The second level is *Important*, which should includes major components and features of test framework; Medium is for features that make the testing and analysis more efficient; Low means unnecessary components and features.

- Very important 10
- Important 6
- Medium 3
- Low 1

3. Virtual Vehicle Setup in Hardware-in-loop Environment

Evaluation of Toolchains						
Parameters and Rationale			Weight Factor	Weighted Score		
				VTD & VeDyna	CarMaker 6.0	VTD & FMU
Vehicle Model	Sensor model (Radar) Availability of different sensor models based on their fidelity		10	3	4	×
	Fidelity of the Chassis model		10	5	5	×
	Actuator Model (Steering model) Fidelity used in the tool chain		6	2	5	×
Environment Creation	Static Objects (Road, Lane Markings, Barriers, Intersections etc.)	Userfriendliness to create	6	4	5	×
		Support for Open Drive	10	5	1	×
	Dynamic Objects (Traffic Objects)	Types of traffic objects (library)	6	5	4	×
		Fidelity (o dof or bicycle models)	3	3	3	×
Scenario Generation	Maneuver generation for ego vehicle		6	2	5	×
	Maneuver generation for traffic objects		3	5	5	×
	Triggers		6	5	5	×
Simulation master for the Automation	Automation of test cases, simulation and postprocessing		6	4	5	×
Platform for Integration	Multi-domain integration(chassis, driver, actuators, sensors)		10	3	4	×
Commonality with MIL and SIL	Possibility to run the same setup on MIL/HIL		3	1	4	×
Visualization	Quality of 3d Rendering		10	5	4	×
Execution Speed	Compiling and building time		10	3	5	×
Complexity of architecture	Understanding of signal interface		10	3	5	×
Postprocessing	×		6	4	5	×
Driver Model	×		×	×	×	×
		Average		3.743802	4.297521	×
		STD DEV				×

Figure 3.13: Evaluation results of three tool chains

4

Discussion

In applying the vehicle dynamics validation manoeuvres to test different chassis models with external steering models in the SPAS and CarMaker platforms (both in the MIL environment), parameters like yaw rate, lateral acceleration, roll angle and torsion bar torque were analyzed. Since Figure 2.6-2.10 shows that the results from FMU-model are closer to measurement from real test log than those from default chassis model, it is concluded that the current FMU-model gives good correlation with in-vehicle test results. Statistical correlations wasn't done during the validation because the limited time and a detailed vehicle dynamics evaluation might be out of range of our scope. But if the FMU-chassis model would be further developed in the future, one approach[21] could be developing an error function, which based on the error between the simulated and real values of lateral acceleration and yaw rate, and performing the functional evaluations and iterations to tune the model.

Similar vehicle dynamics validation manoeuvres were performed in the HIL environment after setting up the IPG CarMaker HIL framework. The purpose was to evaluate that if all the models could be integrated and work properly in the real-time platform. (Vehicle model already validated in the Vehicle Dynamics Group at Volvo Cars.) Figure 3.8 and 3.9, top-middle panels showed that the powertrain model could not provide enough traction for acceleration since there were no correct powertrain models compatible for dSpace real-time platform.

In order to evaluate different HIL frameworks, driver assistance system simulation has to be performed in IPG CarMaker and VTD&VeDyna setup. In our case, close-loop driver assistance system simulation was changed to open-loop simulation since the virtual ECU models did not provide correct signals to trigger the functions. In addition, emergency lane keeping assist was the only test scenario because it was complex and a representative of collision avoidance function. In doing so, we succeeded in performing the evaluation test in spite of some shortcomings. One thing was that the result from open-loop simulation would have big difference between the real test log since the steering was actuated "manually" instead of being triggered by the ECU. Therefore, it was meaningless to do the statistic evaluation. If the HIL setup could be fixed in the future, it would be beneficial to run simulation on different scenarios with variable roads, vehicle speed and test cases[22].

The evaluation method implemented in this project to compare different HIL frameworks is a most common *Evaluation Criteria* method[23]. The process includes: identifying requirements of the system, defining evaluation test criteria, specifying

scoring system and performing the simulation. Many of the evaluation methods and techniques have also been widely studied and used to analyze and select simulation software. One advanced approach is *Analytic Hierarchy Process (AHP)*[24]. The main advantages of these methods are the simplicity and scalability since the problem is formed in a flexible hierarchy structure. Another approach can be *Two-phase evaluation and selection methodology*[25]. It divides the evaluation process into two phases: Feature Check and Quality Check. This methodology would be effective and reliable if more than 2 simulation software were evaluated.

It is obvious that the structure of the VTD&VeDyna framework is more complicated than that of the IPG CarMaker framework since the vehicle model from Vedyna has to be put in the VTD road/environment. The co-simulation causes inevitable latency in the system.

The *Scenario/Environment* of VTD, which includes *Road, Lane marking, barriers, Intersection and traffic objects*, has higher quality than in CarMaker because of better rendering quality. High quality virtual environment is beneficial for the camera. However, all the static and dynamic objects from CarMaker can also be detected by the camera.

Scenario creation in VTD provides much more features than IPG CarMaker, like setting triggers, motion of traffic objects, group objects selection and etc. Road design of VTD also supports OpenDrive (an open file format for the logical description of road networks). On the other hand, creating a scenario in CarMaker is simpler without these features. Compromises have to be made for different use cases.

The *vehicle model* from Vedyna was not tested separately during the thesis project. But according to the validation work, its performance is as good as that of CarMaker. Unfortunately, the steering model from the supplier was not integrated in the VTD&VeDyna framework yet. That makes the lateral response of the model in VTD&VeDyna not good as in CarMaker.

As *user friendliness* is concerned, the co-setup between IPG CarMaker and dSpace ControlDesk has several issues. CarMaker does not provide good interface for dSpace hardware and software. However, it also has to be mentioned that VTD&VeDyna setup has been used at Volvo Car for long time and the interface is kept developing during this time.

On the other hand, the script control and data trace features in IPG CarMaker make it efficient to run a batch simulation, analyze the data and do the post-processing. For VTD&VeDyna framework, test automation, data analysis and post-processing methods had to be developed by ourselves. Instead of driving test vehicles on the real road for thousands miles, one of the advantages of running simulation in virtual environment is to test the intelligent system with different cases day and night. And that cannot be done without test automation.

In addition, the time of compiling and building model is shorter in CarMaker. The whole process took 9 minutes while it took 17 minutes for VTD&VeDyna model.

Overall, the IPG CarMaker HIL framework gives solid performance for collision avoidance function verification because of the accurate dynamics response for steering and braking. It shows a promising future for the application in hardware-in-loop test. The VTD & VeDyna HIL framework has big advantage in the verification of camera based active safety function like *Traffic Sign Information* because of its high quality virtual environment. But shortcomings caused by co-simulation still have to be solved.

Using a CAE environment to verify active safety functions or autonomous driving is still under development and investigation. There are only few proven software or tools available in the market and companies (like Waymo, Uber) will not share their self-developed CAE software with competitors. Therefore, evaluation of CAE tool chains (especially HIL frameworks) for intelligent safety system is not being commonly reported in scientific work and/or being publically available. In this report, literature reviews had to focus on the different evaluation methods, rather than the results of the applications of such methods.

5

Conclusion

This thesis validated different vehicle models (IPG CarMaker, FMU-Dymola and default model in SPAS) with vehicle dynamics manoeuvres in the MIL environment. A lot of effort was put in setting up IPG CarMaker in the HIL environment and a benchmark test was developed to evaluate the IPG CarMaker framework and the VTD&VeDyna framework. Important components and parameters of the tool chains were compared separately or simulated as a whole system with emergency lane keeping aid function as focus.

By applying the open-loop driver assistance system simulation explained in previous chapter, it can be conclude that IPG CarMaker has good user friendliness and is easy to setup and debug. Since it provides the benefit of a single tool chain creating the virtual environment, scenario and vehicle model, open-loop simulation can be setup and run without any latency. Meanwhile, the 3D-rendering in the IPG CarMaker is good enough for object detection for active safety use cases. The script control and data trace features of CarMaker make it efficient to run a batch simulation, perform data processing and analysis. The time of compiling and building model is shorter in the CarMaker than that in the VTD&VeDyna framework.

As for the VTD & VeDyna HIL framework, it has good performance in closed-loop simulation with ideal steering model. The scenario/environment visualisation of VTD, which includes Road, Lane marking, Barriers, Intersection and Traffic Objects has higher 3D rendering quality. More importantly, VTD supports OpenDrive database and the framework of VTD VeDyna and dSPACE Controldesk serves smoothly for the users. However, this setup can not include high fidelity models for steering and experienced latency issues due to co-simulation between various platforms.

It is conclude that:

- VTD & VeDyna HIL framework is on the top list for camera based active safety function verification like *Traffic Sign Information* since VTD creates high quality virtual environment. But it also has limits when the function is activated and actuators start to take the control of the vehicle.
- IPG CarMaker HIL framework gives solid performance for collision avoidance function verification because of the accurate dynamics response for steering and braking. The 3D-rendering quality is also proven to be good enough for objects detection and works for emergency lane keeping assist function.

6

Future Scope

- The HIL test bench has to be fixed to make sure that the closed-loop simulation is possible and active safety function can be triggered. This step needs a lot of work since all the virtual electronics modules have to be checked and debugged.
- VTD & FMU HIL framework has to be prepared and implemented in the current test bench. Currently, the resources are limited. However, it is quite promising since the FMU-Dymola model already shows a good performance in the MIL environment. In addition, since the steering model is already integrated in FMU-Dymola model, driver assistance system including lateral control can be tested with this framework.
- Steering system, including the steering plant and the steering control unit, has to be implemented in the VTD & VeDyna tool chain in order to enable the emergency lane keeping assist function. This means that a solution has to be found for latency issue due to co-simulation between various platforms.
- On the other hand, brake control unit model has to be implemented in the CarMaker framework to enable the emergency braking function. This is feasible if the communication network should be fixed.
- More thorough research should be performed on the tool chains. Experts who are using these tool chains should be involved to give their opinions. Thus, the feature and performance of the tool chains can be better understood.

Bibliography

- [1] M. Feilhauer, J. Haering, and S. Wyatt, “Current approaches in hil-based adas testing,” *SAE International Journal of Commercial Vehicles*, vol. 9, no. 2016-01-8013, pp. 63–69, 2016.
- [2] C. Hobbs, P. Gloyns, S. Rattenbury, and J. Roberts, “European new car assessment program (euro-ncap),” 1997.
- [3] L. Heidrich, B. Shyrokau, D. Savitski, V. Ivanov, K. Augsburg, and D. Wang, “Hardware-in-the-loop test rig for integrated vehicle control systems,” *IFAC Proceedings Volumes*, vol. 46, no. 21, pp. 683–688, 2013.
- [4] V. Jaikamal, “Model-based ecu development—an integrated mil-sil-hil approach,” SAE Technical Paper, Tech. Rep., 2009.
- [5] O. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen, “Development of a driver information and warning system with vehicle hardware-in-the-loop simulations,” *Mechatronics*, vol. 19, no. 7, pp. 1091–1104, 2009.
- [6] O. J. Gietelink, “Design and validation of advanced driver assistance systems,” Ph.D. dissertation, TU Delft, Delft University of Technology, 2007.
- [7] <http://elinux.org/Toolchains>, *Toolchains*, 22 September 2016.
- [8] A. Eidehall, “Tracking and threat assessment for automotive collision avoidance,” Ph.D. dissertation, Institutionen för systemteknik, 2007.
- [9] A. Eidehall, J. Pohl, F. Gustafsson, and J. Ekmark, “Toward autonomous collision avoidance by steering,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 84–94, 2007.
- [10] S. Ishida and J. E. Gayko, “Development, evaluation and introduction of a lane keeping assistance system,” in *Intelligent Vehicles Symposium, 2004 IEEE*. IEEE, 2004, pp. 943–944.
- [11] T. Pilutti and A. G. Ulsoy, “Fuzzy-logic-based virtual rumble strips for road departure warning systems,” *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 1, pp. 1–12, 2003.
- [12] M. Ganley, J. Dawson, and D. McQuilton, “Benchmarking cae tools for emc,” in *EMC (Electromagnetic Compatibility) in Aerospace (Digest No.: 1996/243), IEE Colloquium on*. IET, 1996, pp. 8–1.
- [13] D. Elmuti and Y. Kathawala, “An overview of benchmarking process: a tool for continuous improvement and competitive advantage,” in *Benchmarking for Quality Management Technology Vol. 4 No. 4*. MCB University Press, 1351-3036, 1997, pp. 229–243.
- [14] A. Heathershaw, “Developments in on-centre steering evaluation and testing,” SAE Technical Paper, Tech. Rep., 2006.

- [15] U. department of transportation national highway traffic safety administration, "Laboratory test procedure for fmvs 126, electronic stability control systems," 1 2010.
- [16] A. ALBINSSON and C. ROUTLEDGE, "The damper levels influence on vehicle roll, pitch, bounce and cornering behaviour of passenger vehicles," master thesis, Chalmers University of Technology, 2013.
- [17] "Driveability test maneuver: Steady-state circular test," p. 1, November 2009.
- [18] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, H. Elmqvist, A. Junghanns, J. Mauß, M. Monteiro, T. Neidhold, D. Neumerkel *et al.*, "The functional mockup interface for tool independent exchange of simulation models," in *Proceedings of the 8th International Modelica Conference; March 20th-22nd; Technical Univeristy; Dresden; Germany*, no. 063. Linköping University Electronic Press, 2011, pp. 105–114.
- [19] C. Specification, "Version 2.0," *Robert Bosch GmbH*, 1991.
- [20] F. Consortium *et al.*, "Flexray protocol specification v3. 0.1, 2010."
- [21] M. F. Kinstle, D. Hassler, and B. S. Johnson, "Vehicle dynamics benchmarking and simulation," SAE Technical Paper, Tech. Rep., 2009.
- [22] F. Coşkun, Ö. Tunçer, M. E. Karsligil, and L. Güvenç, "Real time lane detection and tracking system evaluated in a hardware-in-the-loop simulator," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 2010, pp. 1336–1343.
- [23] R. A. Bauernschub and D. E. King, "Benchmarking mid-range cad tools in a diverse product environment: recommendations and results," in *Thermal and Thermomechanical Phenomena in Electronic Systems, 2000. ITherm 2000. The Seventh Intersociety Conference on*, vol. 1. IEEE, 2000, pp. 193–198.
- [24] L. Davis and G. Williams, "Evaluating and selecting simulation software using the analytic hierarchy process," *Integrated manufacturing systems*, vol. 5, no. 1, pp. 23–32, 1994.
- [25] Y. Alomair, I. Ahmad, and A. Alghamdi, "A review of evaluation methods and techniques for simulation packages," *Procedia Computer Science*, vol. 62, pp. 249–256, 2015.
- [26] carsim.com, *Sine with Dwell Test in CarSim*, December 2011.
- [27] C. Schwarz, "Technical note nads vehicle dynamics typical modeling data," *IFAC Proceedings Volumes*, p. 5, 2006.
- [28] T. DYNAware, "vedyna:more efficiency in component and controller development with comprehensive vehicle dynamics simulation," pp. 2–5, 2017.
- [29] *IPG CarMaker User's Guide*, 5th ed., IPG Automotive, Bannwaldallee 60, 76185 Karlsruhe, Germany.
- [30] M. Otter, "Functional mockup interface (fmi)," 1 2010.
- [31] A. KARLSSON, "Test procedures and evaluation tools for passenger vehicle dynamics," master thesis, Chalmers University of Technology, 2014.
- [32] H. Hanselmann, "Hardware-in-the-loop simulation testing and its integration into a cacs d toolset," p. 1, September 1996.
- [33] K. Mineta, K. Unoura, and T. Ikeda, "Development of a lane mark recognition system for a lane keeping assist system," SAE Technical Paper, Tech. Rep., 2003.

- [34] J.-F. Liu, J.-H. Wu, and Y.-F. Su, “Development of an interactive lane keeping control system for vehicle,” in *Vehicle Power and Propulsion Conference, 2007. VPPC 2007. IEEE*. IEEE, 2007, pp. 702–706.
- [35] J. Hwang, K. Huh, H. Na, H. Jung, H. Kang, and P. Yoon, “Evaluation of lane keeping assistance controllers in hil simulations,” *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 9491–9496, 2008.
- [36] M. Khoe-Fard and T. Lahdhiri, “Challenges in real time controls simulation (hardware-in-the-loop) in active safety for subsystem level software verification,” *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, vol. 4, no. 2011-01-0450, pp. 105–110, 2011.
- [37] B. Heißing and M. Ersoy, *Chassis handbook: fundamentals, driving dynamics, components, mechatronics, perspectives*. Springer Science & Business Media, 2010.
- [38] J. D. Setiawan, M. Safarudin, and A. Singh, “Modeling, simulation and validation of 14 dof full vehicle model,” in *Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME), 2009 International Conference on*. IEEE, 2009, pp. 1–6.
- [39] L. Stopczynski and M. Milford, “Collision mitigation system,” in *United States Patent*.

A

Appendix A Simulation Plots on MIL for Carmaker VS FMU VS Invehicle Log

Results Catalogue Occ		
Test	vx(kph)	ay(g)
occ3	25	0.2
occ5	50	0.2
occ6	50	0.4
occ7	80	0.2
occ9	80	0.4
occ10	100	0.2
occ11	120	0.2

Table A.1: Results Catalogue A

Results Catalogue Crc		
Test	Turn Direction	Radius(m)
crc1	left	40
crc2	right	40

Table A.2: Results Catalogue B

Results Catalogue Frc		
Test	vx(kph)	SWA Amplitude(rad)
frc1	120	0.184
frc2	120	0.2375
frc4	120	0.239
frc5	120	0.2379
frc6	120	0.2406
frc7	120	0.2559
frc8	120	0.2508
frc9	80	0.2515
frc10	80	0.33
frc11	80	0.3584
frc12	80	0.3501
frc13	80	0.3554
frc14	80	0.3503

Table A.3: Results Catalogue C

Results Catalogue hssc			
Test	Turn Direction	SWA at 0.3 g(deg)	
hssc1	left	27.14	
hssc2	left	27.51	
hssc3	left	27.25	
hssc4	left	27.56	
hssc5	left	27.31	
hssc6	right	27.63	
hssc7	right	28.64	
hssc8	right	27.73	
hssc9	right	27.21	
hssc10	right	27.20	

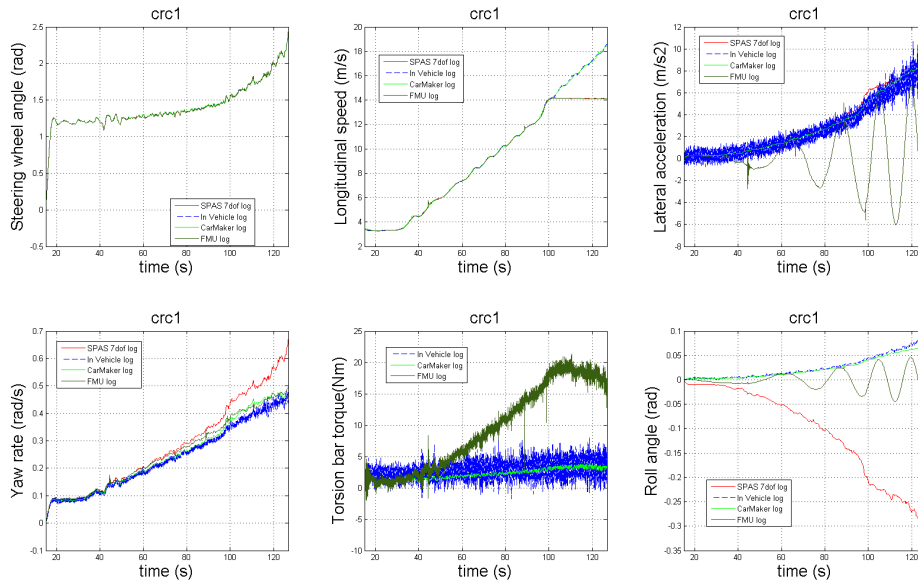
Table A.4: Results Catalogue D

Results Catalogue lssc			
Test	Turn Direction	SWA at 0.05 g(deg)	vx(kph)
lssc1	left	5.74	80
lssc2	left	6.13	80
lssc3	left	5.73	80
lssc4	left	5.96	80
lssc5	left	5.83	80
lssc6	left	5.22	80
lssc7	right	5.78	80
lssc8	right	5.53	80
lssc10	right	5.16	80
lssc11	right	5.59	80
lssc12	left	4.57	120
lssc13	left	3.76	120
lssc14	left	4.01	120
lssc15	left	4.77	120
lssc16	left	2.93	120
lssc17	right	4.21	120
lssc18	right	4.90	120
lssc19	right	4.45	120
lssc21	right	2.66	120
lssc22	right	4.20	120

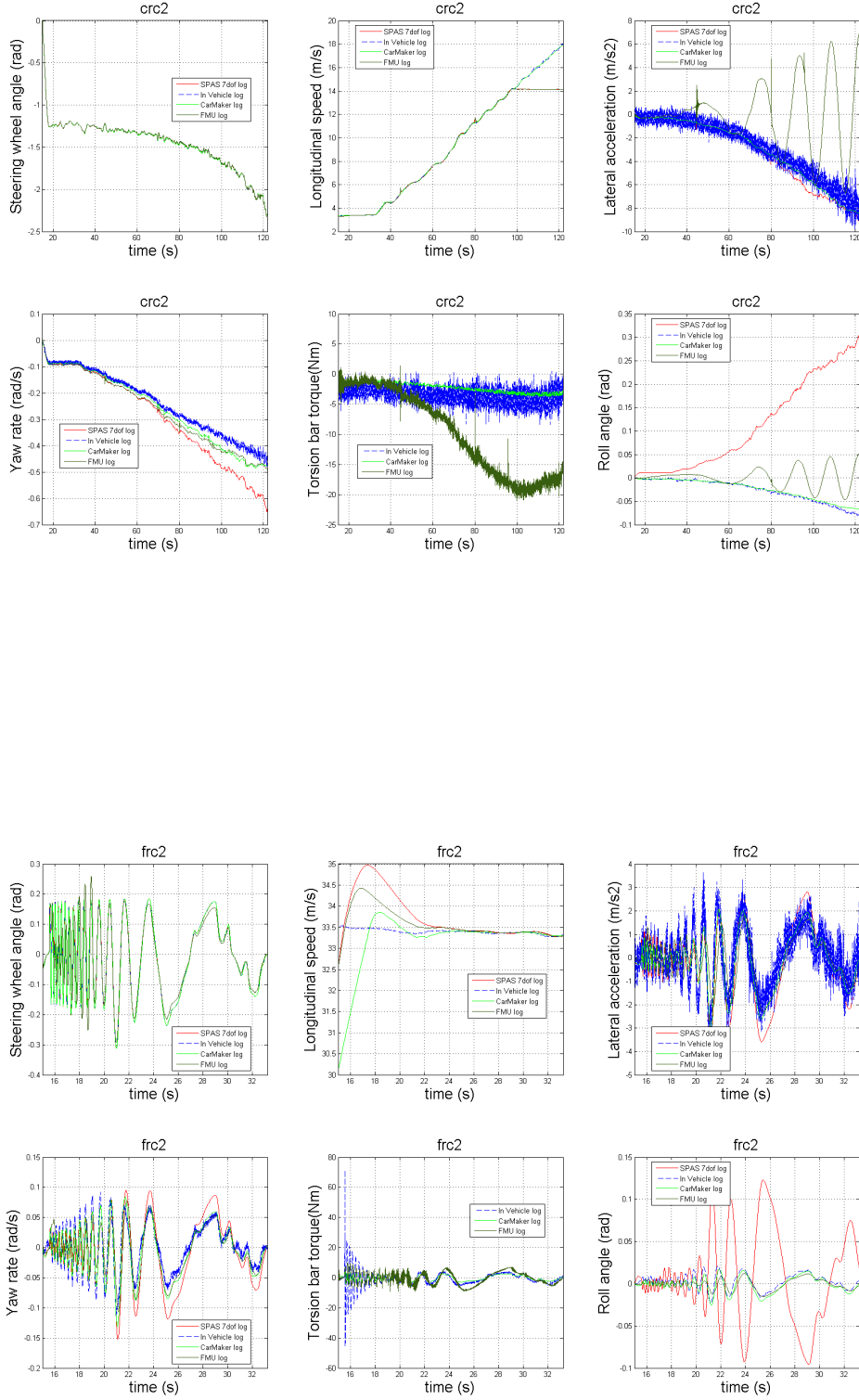
Table A.5: Results Catalogue E

Results Catalogue sinc					
Test	Turn Direction	Sideslip 1st Peak	Front	Sideslip 2nd Peak	Rear
sinc1	right to left	0.21		0.32	
sinc2	right to left	0.10		0.18	
sinc3	right to left	2.10		0.37	
sinc4	right to left	2.08		0.32	
sinc5	right to left	2.87		0.55	
sinc6	right to left	2.82		0.47	
sinc7	right to left	4.47		0.71	
sinc8	right to left	4.45		0.67	
sinc9	right to left	6.11		0.89	
sinc10	right to left	6.04		0.43	
sinc11	right to left	5.59		0.95	
sinc12	right to left	7.80		0.72	
sinc13	right to left	3.76		0.45	
sinc14	right to left	9.69		0.82	
sinc15	right to left	4.77		0.46	
sinc16	right to left	11.44		0.61	

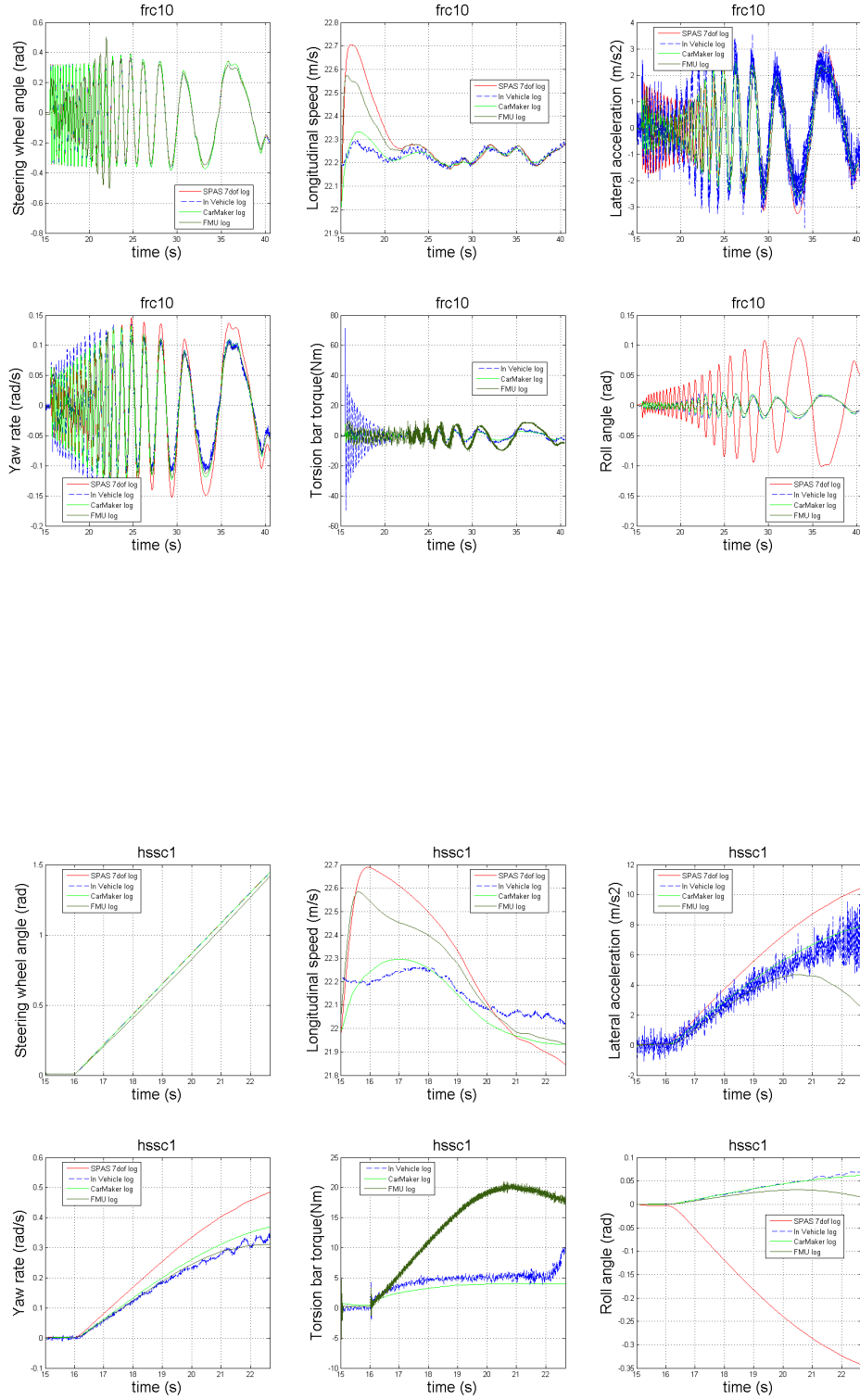
Table A.6: Results Catalogue F



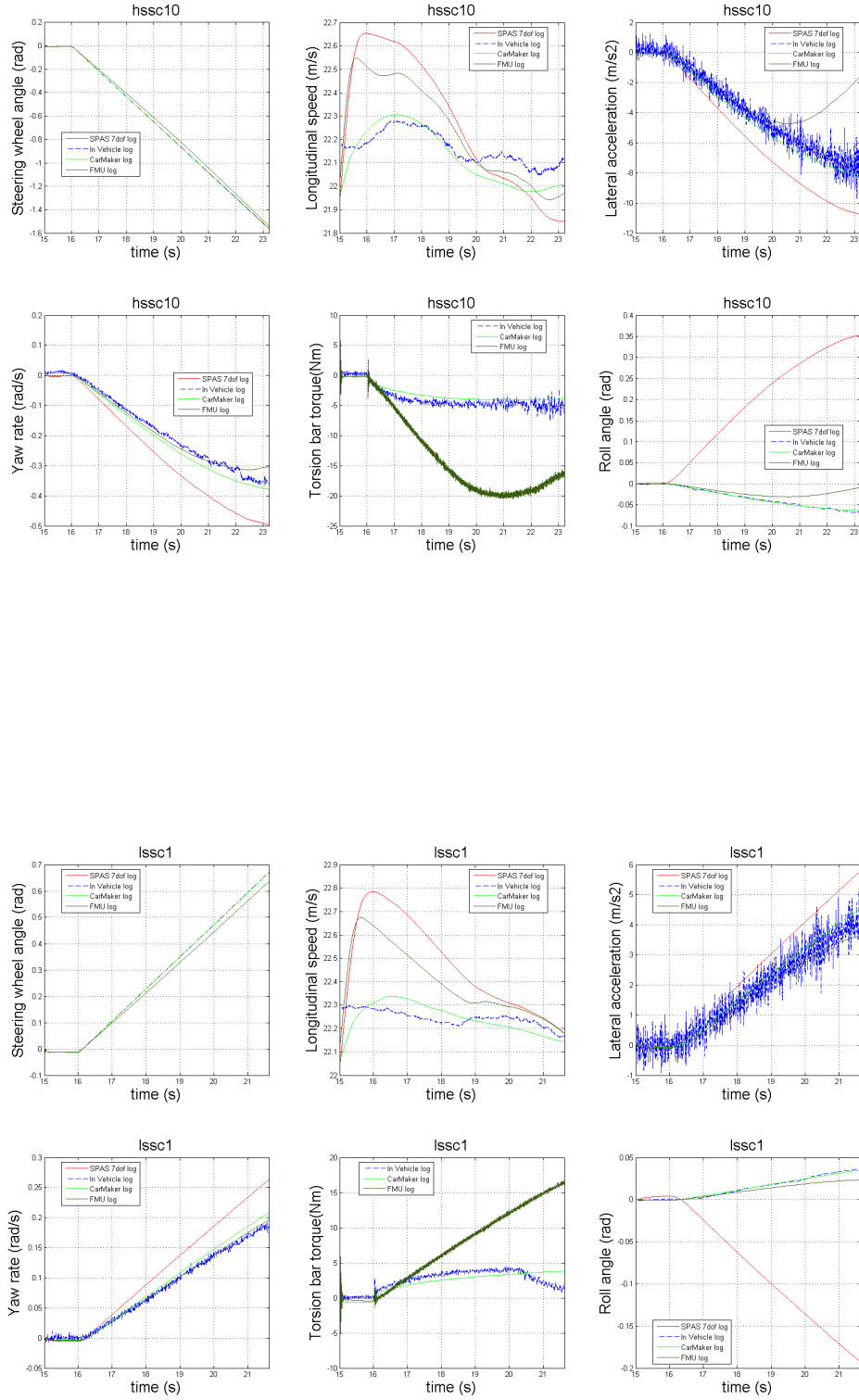
A. Appendix A Simulation Plots on MIL for Carmaker VS FMU VS Invehicle Log



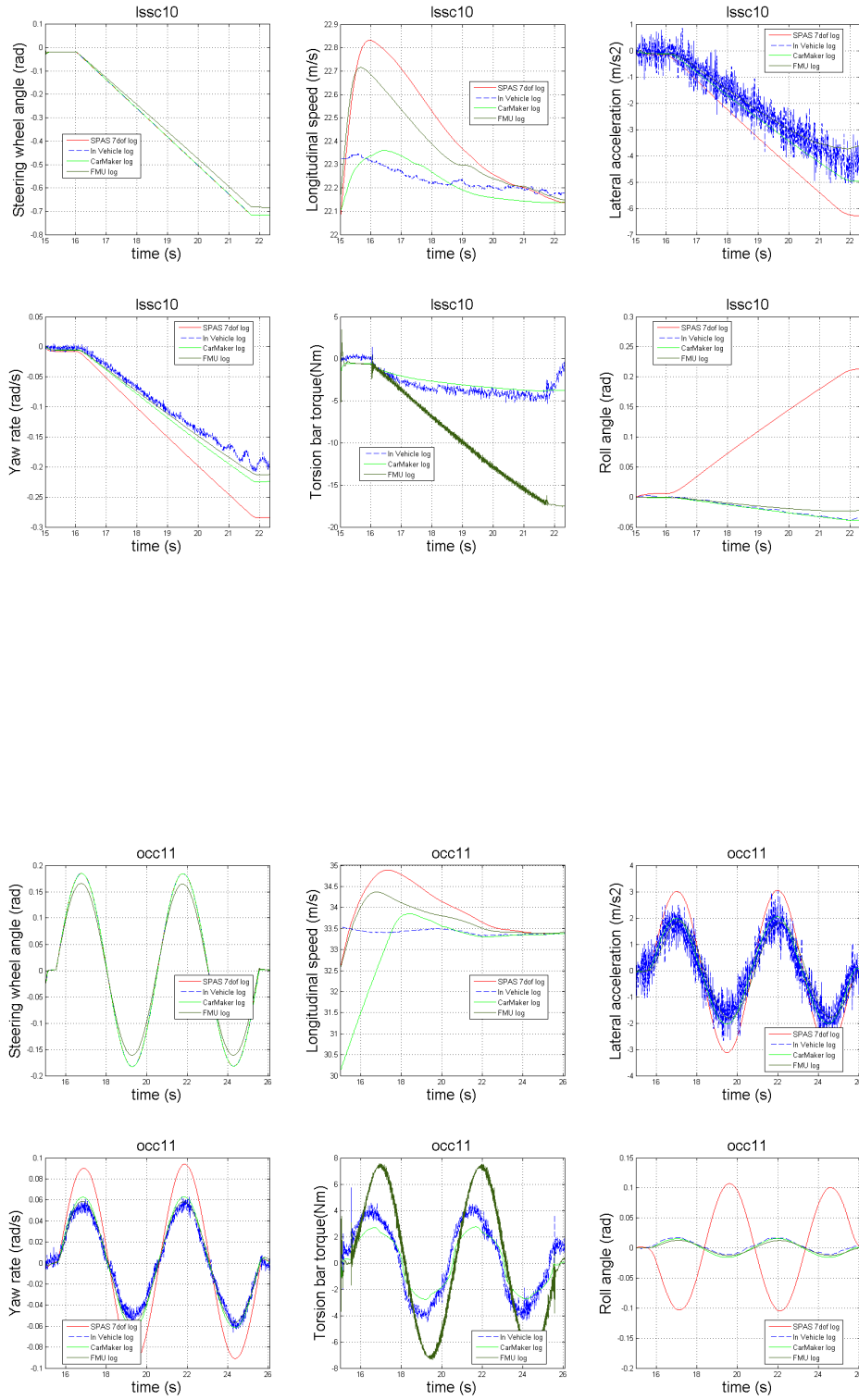
A. Appendix A Simulation Plots on MIL for Carmaker VS FMU VS Invehicle Log



A. Appendix A Simulation Plots on MIL for Carmaker VS FMU VS Invehicle Log



A. Appendix A Simulation Plots on MIL for Carmaker VS FMU VS Invehicle Log



A. Appendix A Simulation Plots on MIL for Carmaker VS FMU VS Invehicle Log

