VIENNA 2018 TRA
TRANSPORT RESEARCH ARENA
A digital era for transport
solutions for society, economy and environment

Proceedings of 7th Transport Research Arena TRA 2018, April 16-19, 2018, Vienna, Austria

ENABLE-S3 – Advanced V&V technologies and methods combined with simulation and testing environments enable the safe and secure development of Autonomous Vehicles

Horst Pflügl*, Andrea Leitner, Hannes Schneider

AVL List GmbH, Hans-List-Platz 1, 8020 Graz, Austria

Abstract

Highly automated and autonomous transport is a technology field that enables safer and cleaner transport and unburdens the driver from boring and/or error prone driving task. The development of automated transport features and vehicles will or have already led to new business opportunities in many technology sectors, like sensor technologies, SW-development or mobility services to name just a few of them. The highly precise sensors and communication technologies as well as the necessary computing power and algorithms within the vehicle plus the digital infrastructure that are necessary to realize the autonomous transport are developing very fast. But this goes also along with new heavy-weight challenges in terms of safety and security aspects. Extensive verification and validation efforts are necessary to make automated systems at least as safe as human-operated systems are nowadays.

The ENABLE-S3 project develops verification and validation technologies and methods that will help to tackle this challenge with reasonable efforts and high coverage of test-cases. 71 partners from different transport sectors (automotive, aerospace, rail, maritime, farming) and other industries are creating new knowledge in the areas of testing and simulation methods & technologies as well as the required testing platforms and environments.

Research within ENABLE-S3 focuses on:

- Test and simulation environments supporting open standards (e.g. Functional Mock-up Interface, OpenSimulationInterface) wherever possible in order to run tests for automated transport seamlessly in different virtual and semi-virtual environments.
- Open standards for the definition, management and execution of test cases/testing scenarios like OpenScenario or OpenDrive and their relationship to other existing standards like ASAM-XiL.
- Investigation of testing methodologies which are necessary to reduce the number of test cases tremendously, among them are DoE (design of experiments), combinatorial testing, FMEA analysis etc.
- Development of sensor models as well as sensor stimuli (physical sensor signal generators).
- Generation of test cases out of existing recorded real-world data.

The developed methods are applied in different industrial use-cases. This paper will give an overview over the needed building blocks for testing AD functions, including scenario generation, test planning, and test execution and simulation that were already developed within the ENABLE-S3 project and will finally present a practical use case and the application of aforementioned methods to an ACC function of a vehicle.

The results gained so far in the project will show that the verification and validation methods combined with simulation and testing technologies for automated vehicles in transport play a major role in reaching the high safety

---

* Horst Pflügl. Tel.: +43-316-787-1587
E-mail address: horst.pfluegl@avl.com

and security levels that end customers and legal authorities will demand for this important technology in order to get acceptance and in order to provide a great step forward in reducing road fatalities and at the same time also CO2 emissions.

Keywords: ADAS; Automated driving; Verification & Validation; Test planning, DoE; Scenario generation; Environment simulation;

Nomenclature

ADAS	Automated driver assistance systems
AD	Automated driving
ADF	Automated driving functions
MIL, SIL, HIL, VIL	Model (Software, Hardware, Vehicle) in the loop
KPI	Key performance indicator

1. Introduction

Technologically the development of automated driving functions is satisfactorily understood and witnessed by several millions of test kilometres already travelled by automated cars on public roads. The ever-increasing demand and technological improvements in functionalities are leading to greater safety, lesser accidents as well as more efficient and environmentally friendly traffic. ADAS systems, and in the upcoming years Automated Driving Functions (ADF), have - respectively will - slowly become an irreplaceable part of the everyday driving experience. Nevertheless, Watzenig et al. [4] state that new validation methodologies, procedures, and laws are needed in order to successfully incorporate emerging technologies into traffic and thus improve safety, reduce emissions, provide traffic flow optimization and enhanced mobility. Some steps towards this goal have already been taken. The EU made legal obligations on new passenger cars to include certain safety-related ADAS systems (EPS, EBA) and the level of automation will further increase in the following years.

However, demonstrating the reliability, safety, and robustness of the technology in all conceivable situations, e.g. in all possible traffic situations under all potential road and weather conditions, has been identified as the main roadblock for product homologation, certification and thus commercialization. Winner et al. [5] as well as Wachenfeld et al. [6] predict that more than 100 million km of road driving would be required to statistically prove that the automated vehicle is as safe as a manually driven car.

This is not feasible with current verification methods as it would require several years of testing. Therefore, research is ongoing in several research projects to develop accelerated testing methods. OEMs currently mainly rely on proving ground or public road testing in order to validate their systems because of the lack of alternatives. The test scenarios are usually taken from collections generated by engineers, which include the complete scenario description together with the expected response of the automated system. However, it cannot be proven that all possible scenarios have been covered. Scenarios on the road are influenced by the behaviour of human drivers, other traffic participants, changing weather and road conditions, which are all not 100% predictable. Therefore, it is hardly possible to reliably calculate the test coverage. In addition, proving ground and real world testing is associated with high costs, low reproducibility and long validation times. Especially reproducibility in a real-world setup is challenging because of the difficulty to reach correct initialization, exact traffic behaviour, similar environmental influences, and so on. Furthermore, safety is a very important aspect and further limitations arise because some test cases could be dangerous or even impossible to be carried out by human drivers. All these limitations add up and influence the overall time needed to successfully validate an ADAS or ADF function.

New approaches are needed to reduce the effort required by today’s state-of-the-art practices by orders of magnitude in order to become economically acceptable and technically feasible.

2. Building blocks for testing highly automated vehicles or cyber physical systems

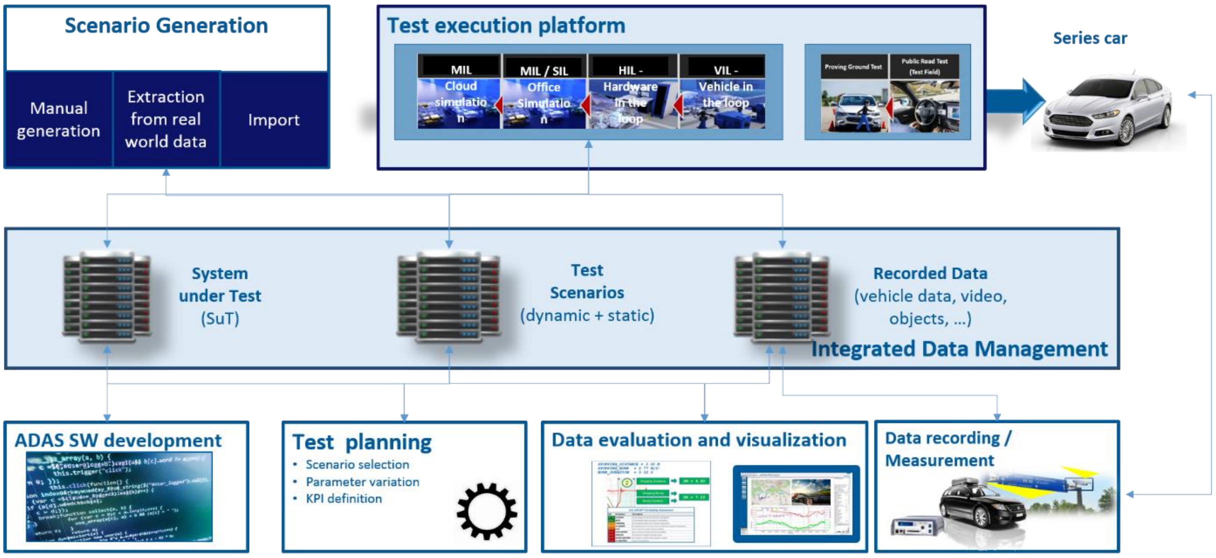

Figure 1: Building blocks for testing automated driving

The ENABLE-S3 project with 71 partners from different industries and academia has set itself the goal of developing a modular verification & validation framework, which consist on the one hand of the validation methodologies (including the test scenarios) and on the other hand of the validation platform (or tools) that is needed to carry out the tests that are needed. The methods and platform shall be developed along selected so called use-cases (like for instance a highway-pilot). As a result of the analysis and requirement elicitation in the first phase of the project, a generic test architecture (compare chapter 5) and the needed building blocks have been identified to get a common understanding of the challenge and base for discussions and collaboration. Figure 1 shows the main building blocks that are required for testing vehicles with ADAS and ADF functions for automated driving or highly automated cyber physical systems and shall be introduced shortly in the following paragraphs:

Scenario Generation: One main aspect for a scenario-based validation approach is the generation of respective testing scenarios which are the basis for creating test cases. Scenarios can be generated either manually or extracted from real world data. A reuse and therefore data exchange from or with other data sources or projects is also feasible. The scenarios include static aspects (road, environment / weather conditions, …) as well as dynamic aspects (own driving path, other moving traffic participants, …) and represent every information that is needed to reproduce a driving scenario in virtual, semi-virtual or real environment in an appropriate form. The scenarios could also be named test descriptions. However, the scenario information needs to be stored in an efficient way and format, and need to be transferable into various executable test environments (MIL - model in the loop, SIL – software in the loop, HIL – hardware in the loop, VIL – vehicle in the loop). If one would like to search in a scenario database for specific scenarios, there should also be relevant meta data or tagging data stored along with the scenarios, in order to ease the searching task.

Test execution platform: To make the validation work most efficient, it is expected to execute as many test cases as possible in pure virtual environments, which is the cheapest way of doing it. One main advantage in virtual environments is the fact that simulations can be executed faster than real-time and can significantly reduce validation time. Further there is still no need for expensive prototypes and there are no safety risks for involved traffic participants that or not real but only virtual. Nevertheless, a complete validation of the vehicle is not feasible in a pure virtual environment because there is always a lack of simulation accuracy or details to ensure 100% reproduction of reality. Therefore, various semi-virtual and real world tests are required as well to further close this gap. As mentioned above a VIL (compare Figure 2) platform could be useful for certain tests that are safety critical or dangerous for human drivers. In this case the complete vehicle is mounted on a testbed and the surrounding environment is simulated and physically applied by actuators to the vehicle. I.e. beginning from the missing torque to the tires of the car which is physically applied by means of e-machines directly mounted on the axles of the wheel and simulated with the environment simulation software. The same is true for steering forces to the front wheels which are also applied by means of e-motors or for radar reflection signals, ultrasonic signals and GPS signals which are applied with appropriate electronic devices. Even for the car2x connection or for the video camera there are appropriate devices installed at the testbed in order to simulate/stimulate the scenarios with best possible accuracy and details.

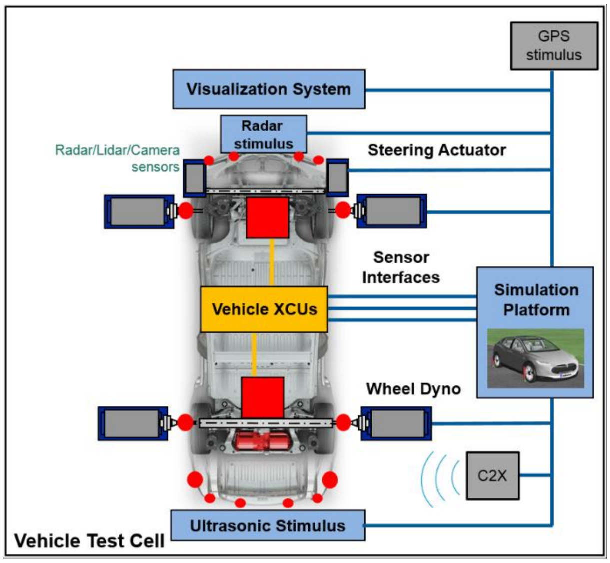

Figure 2: VIL vehicle in the loop test platform – also named DrivingCube™

Test scenario database/formats: One major goal in the entire verification process is the reuse of test descriptions (or test scenarios) as well as the involved models used for vehicle, environment, etc. within different if not every test platform. In order to achieve this efficiently, the use of open and standardized formats (e.g. OpenDrive, OpenScenario, Road5, etc.) is required. I.e. the scenario must be described in a way that it is useable by many different tools and can be easily exchanged with a standardized exchange format. The current existing initiatives in this direction are OpenDrive [2], which is a description format for static elements within scenarios (road network). The OpenScenario [3] standard completes the testing scenario by means of dynamic data of a scenario (traffic participants). Additionally, the test scenarios need to be stored along with appropriate meta-information to identify and search the scenarios based on specific attributes. The test planning engineer needs a functionality to search for specific test scenarios that are optimized or tailored or just necessary for the test of specific ADF/ADAS functions, so that the selection of appropriate test cases and scenarios can be done very efficiently.

Test planning: With test planning, we mean the process of selecting and planning the appropriate test cases (test scenarios) for testing specific ADF/ADAS functions. At this stage of the testing process the decision have to be made, how many scenarios are selected for the test and how many variations of parameters of the test scenarios are sufficient. Every identified scenario can be varied by changing parameters (like vehicle velocity, weather condition, varying number of traffic participants, number of road lines, etc.). This test planning approach needs to incorporate sophisticated methods to reduce the number of required test kilometres without compromising the test coverage and therefore accelerate the testing work itself. In order to achieve this reduction a lot of data from real world traffic observations is needed and needs to be investigated in detail, in order to reliably judge whether the automated vehicle that is tested with the selected, varied and reduced number of test cases is at least as safe as a manually driven car. Furthermore, it is necessary to define respective KPIs (key performance indicators) in order to evaluate the results of a test run based on these KPIs. The defined KPIs must/may cover safety, comfort, or efficiency criteria.

Database for recorded data: Testing of automated driving functions generates a whole bunch of data, which needs to be collected and managed respectively. The recorded data can include the vehicle data as well as dynamic ground truth data - recorded by reference measurement equipment - that is needed to validate the vehicle sensor perception. This data needs to be stored and linked together in a way that traceability and reproducibility of test cases is ensured. Coping with the huge amount and variety of data produced by an automated vehicle is still a challenge that needs further investigation and research activities. Besides the classical time series data there is additionally data like time stamped object lists coming out of sensors, Video-data, Radar-data and Lidar-data.

Data evaluation and visualization: The last step in this test process is the evaluation and visualization of test results. The big data amounts in the database need to be analysed, processed, evaluated and visualized, so that a test engineer can find proofs for correct function or malfunction of the concerned ADAS/ADF vehicle electronics and software for automated driving. The evaluation process is highly dependent on the respective ADAS/ADF functions and KPIs and therefore need to be selected carefully.

One of the main challenge throughout this testing process with the current simulation and testing environments and platforms is the lack of standardization, which makes the development of generic or common testing approaches extremely cumbersome. The ENABLE-S3 project therefore promotes the establishment of standards wherever possible and reasonable.

3. Scenario Generation

In the context of scenarios, we distinguish between two main aspects, the static and the dynamic content, as shown in Figure 3. The static content covers everything which does not change frequently such as the road network, traffic signs, buildings, and so on. The dynamic content defines the position and behaviour of all the traffic participants involved in such a test run, including the own “ego”-vehicle.

Generating scenarios can either be done synthetically (i.e. manually based on engineering methods like FMEA) or based on recorded data. This is true for static as well as for dynamic aspects. For dynamic aspects, the former one means to generate test scenarios manually based on safety/security analysis or using existing scenario description e.g. EURO-NCAP test scenarios and reproduce them in a scenario editor. The latter means to mine scenarios from recorded real world data. This approach will at the end lead to a more complete scenario database, since observations of the real world are systematically included in the database. Furthermore, this approach gives a good picture about what actually happens in real traffic. Also, extracting statistics about frequencies of scenarios and typical parameter values can give a good indication about the probability of certain occurrences in the real world and thus are an important input in the testing process.

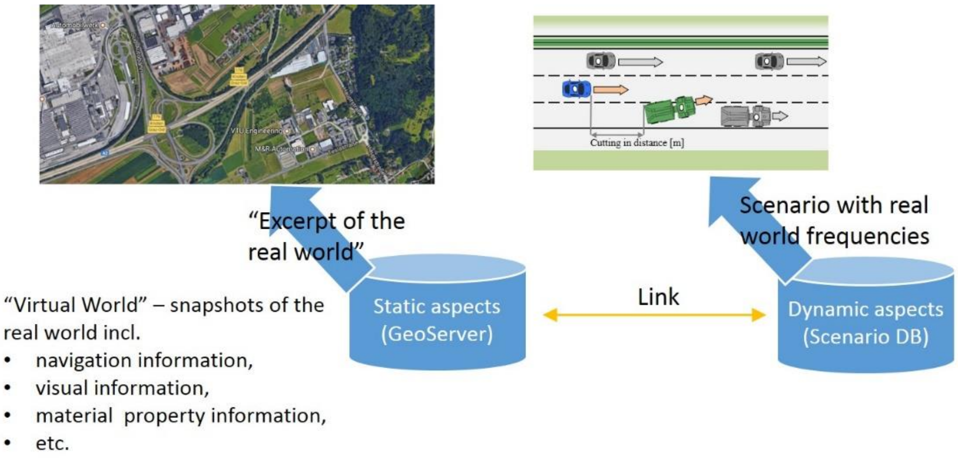

Figure 3: Distinction and relationship between static and dynamic aspects of scenarios

The same is true for static aspects. Virtualizing the real world supports the generation of realistic test environments and provides a possibility to virtually drive a huge amount of test kilometres before testing in a real environment. There are two main aspects: First, the logical description of the road including the curvature and elevation of the road, the lane information including the speed limit/traffic signs and the driving direction on a respective lane. This information is required for the simulator in order to actually navigate. Furthermore, visual information as well as information required by other sensors, such as radar sensors, is required. For all of this, it is required to extract the respective information from the recorded data and give some semantics to it. This means for example that in order to describe the road curvature it is required to “detect” the road in the recorded data and provide it with the respective label. The same is true for objects close to the street, which should be perceived by sensor models. Already this step is tidy and resource intensive since most of it needs to be done manually.

Not everything can be gathered by recorded data. A lot of existing data sources need to be included as well (e.g. GIS data, map data, pictures, etc.). This means that the various data sources need to be fused in order to get a comprehensive description of the real world.

Virtualizing recorded data furthermore provides a possibility to “replay” or “re-simulate” a situation experienced on the road. This might be helpful to reproduce situation where problems occurred after fixing these problems in the software. Reproducing exactly the same scenario in a real-world environment is almost impossible.

Once the data is captured, it needs to be transformed to a format which is understood by the simulation tool.

ENABLE-S3 promotes an open tool environment and work with different simulation tool providers. Thus, ENABLE-S3 partners prefer tools that support open formats and interfaces, such as OpenDRIVE® and OpenSCENARIO®.

OpenDRIVE® (see VIRES_OD [2]is an open file format for the logical description of road networks. It was developed and is being maintained by a team of simulation professionals with large support from the simulation industry. OpenSCENARIO® (see VIRES_OS[3]) is an open file format for the description of dynamic contents in driving simulation applications. The project is in its very early stage and is currently not officially supported by any tool.

The main advantage of a standardized scenario description is the reusability of scenarios in various simulation environments. This is especially important since the development of a comprehensive scenario database should be a joint effort by various players. Since each party should still be able to rely on its preferred development and simulation environment a common format is essential.

4. Test planning

Test planning is an important aspect to reduce the testing effort but guaranteeing the correct behaviour of an automated driving feature or vehicle. As mentioned before, 100 million km of road driving would be required to statistically prove that the automated vehicle is as safe as a manually driven car. Most of the time nothing interesting will happen. In fact, even after driving this huge amount of km it is not assured that all critical scenarios have been covered. The task of test planning is therefore to take the data available in a scenario database in order to plan the covered scenarios in a more efficient way. This means that critical or interesting scenarios need to be selected and combined without running the same scenario twice.

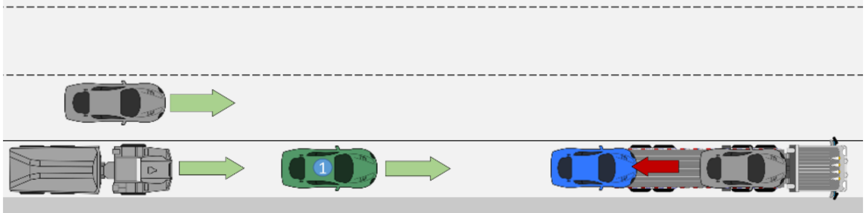

Figure 4: Car following a truck and approaching an obstacle

There are several researchers working on accelerated verification & validation methods, e.g. researchers at the University of Michigan state that they have developed accelerated evaluation methods that would need only 1000 miles of testing [16].

The scenario generation and selection is one of the most discussed topic in automated driving. When do I have sufficient number of scenarios or tests, to determine whether my function behaves as expected. Usually this process is done by defining requirements and accordingly preparing test cases to check each of the requirements. The problem of automated driving is that requirements are often not complete or even not given or ambiguous. This is the case because there are too many influencing parameters into a scenario, which could have a significant impact to a scenario. Just consider a rather simple driving situation (compare Figure 4) of an automated car driving behind a truck on the highway. Then suddenly it becomes more complicated because a large piece of the load (e.g. a car) dissolves from the truck in front because it was not well fastened and falls onto the street in front of the automated car. An intelligent automated car could make an emergency break to stop, before it hits the more or less still standing obstacle in front of it. However, there could be another truck driving behind this automated car, which is so heavily loaded that it is not able to break early enough and hit the automated car in front of it with full or only slightly reduced speed. If the car would be more intelligent, it could decide to make an emergency evasive manoeuvre instead of a braking manoeuvre to avoid hitting the obstacle and to avoid to be hit by the following truck. In this case the automated car would have to check also, if the driving path for the evasive manoeuvre is free (there is another car overtaking on the second lane which might be on the critical path) – and it would have to check and adapting this continuously, because other traffic participants could also adapt their driving path and speed when they are watching the situation. With this example one can see that the number of possible expected reactions of an automated car within a driving scenario might be very hard to specify and hence to cover completely with appropriate test cases.

The definition of the right parameters and their values for the scenarios is done by corner case identification. The goal is to run several simulations with slightly different parameter variations in order to identify those cases which are potentially dangerous or have the potential to end with an accident. The procedure will find cases which can either be considered as safe, or which will for sure lead to an accident or which are very safety critical. For the potentially dangerous scenarios, one cannot make concrete statements just out of the simulation, since there are introduced uncertainties through the generalized models. These test cases need to be re-tested either in a vehicle in the loop environment, on a proving ground or on the public road. Therefore, another important output of the test planning is the assignment of test cases to different testing environments. The goal is to run as many test cases in purely virtual environment for the reasons mentioned above (cheaper, faster, …), however, there is always the necessity for additional tests in mixed virtual/real environment or pure real environment.

4.1 Parameter variations and intelligent reduction of test cases

Scenarios are typically described in a generic way including a set of parameters with a variable value range. Generating variations of test cases means to generate concrete instances of scenarios by fixing the parameters to specific values. E.g. the driving speed of the 3 vehicle in the first lane of Figure 4 could all be set to 95km/h; the distance of the ego-vehicle from the truck in front could be 80m and the car on the second lane could overtake with a speed of 105km/h and could be 70m behind the ego vehicle at the moment when the car dissolves from the truck-trailer in front. Testing all possible combinations of different values for all parameters is extremely time consuming when the number of parameters is high, since the number of test cases grows exponentially with the number of parameters. Therefore, it is required to select only the most interesting parameter combinations, e.g. by applying a DoE approach or combinatorial testing or any other method for intelligent test reduction.

5. Test execution platform - Simulation & Testing Environment

For the execution of test cases within a test execution platform, several technical communication and simulation aspects need to be fixed as shown in Figure 5. The figure acts as a generic architecture – the “ENABLE-S3 test architecture”. The ACPS (automatic cyber physical system, i.e. the automated vehicle in our case) represents the unit under test. Beside the driver (operator) there is control system and the system dynamics of the vehicle, as well as the connection to the environmental sensors and the optional communication unit (e.g. G5-communication). In case the vehicle operates autonomously, the driver is not directly in the loop but gets some feedback from the automated system. In case the SAE automation level is lower than five, the system needs a hand over functionality between the driver and the control system. The ACPS control system represents the actual automated driving function, which is responsible for decision making and trajectory planning based on the information it receives by the environment sensors. These sensors are required to perceive and virtually represent the environment to be used as input for the automated driving function in the control system. The system dynamics block is needed to calculate and simulate the vehicles behaviour dependent on the actuators (engine speed/gas pedal, gear, steering wheel angle, brake pedal). All that blocks can be available in virtual representation or in real. The same is true for the environment itself and the communication infrastructure. It is either represented by virtual simulation models of the environment and the communication infrastructure or in real.

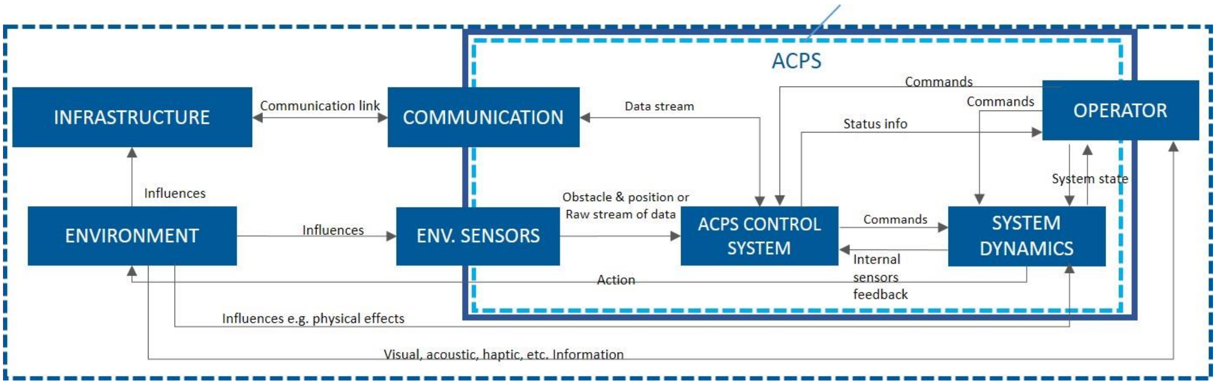

Figure 5: Generic “ENABLE-S3 test architecture” of the test execution platform

Depending on this representation, the test environment (or test execution platform) is either named MiL (model in the loop, i.e. when everything is virtually simulated) or HiL (hardware in the loop, i.e. when sensors and or the control system is in hardware and the rest virtual) or ViL (vehicle in the loop, i.e. the entire vehicle is real on the test bed and the rest is virtual). Proving ground with special environment and dummy traffic participants (pedestrians or vehicles made of carton fastened on remotely controlled, moving platforms) complete the test execution platforms, before it comes to tests on real roads with real traffic participants. At public roads the likelihood of failures needs to be already near to zero, otherwise it might get very dangerous in safety critical situations, even though specifically trained personnel is driving the automated vehicles and can intervene at any time.

The generic ENABLE-S3 test architecture has been defined in order to make different test set ups with different tools and different interfaces comparable with each other as a first advantage within the project. This is the best way of learning from each other, exchanging ideas and best practices with each other or even reusing developed assets or methods within different use cases. The second advantage of this generic architecture is the sound basis for standardization activities. If every team within ENABLE-S3 that is working on different use-cases and driving functions or other automated functions is using the same architecture and interfaces, the basis for standardizing interfaces, scenarios and other description languages is set.

The most important topics that are obvious for standardization are (compare also chapter 3).

- Scenario description language (e.g. OpenScenario, OpenDrive)
- Interfaces for co-simulation (e.g. OSI = Open Simulation Interface, FMI = Functional Mockup Interface, ACI = Advanced co-simulation interface)

E.g. the Open Simulation Interface is a specification that aims for a common exchange format for ground truth and sensor data over an interface.

6. Testing and optimizing the behaviour of an ACC controller

To demonstrate the procedures and building blocks described above, the test planning, execution, analysis and optimization of an ACC controller by means of a simple use-case shall be shown.

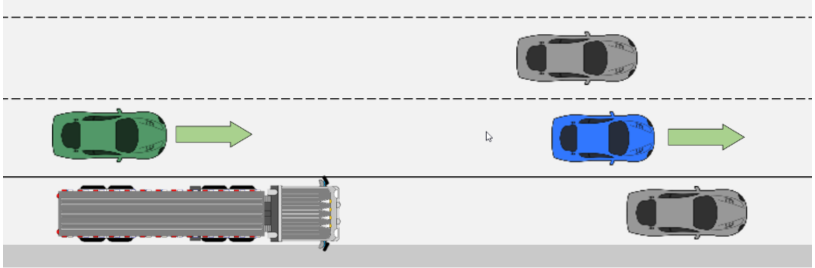

Figure 6: Scenario for corner case identification of an ACC function

It has already been explained in chapter 4 that the corner case identification plays an important role in test planning and hence test execution. The objective of corner case identification is the identification of relevant test cases for a given scenario, by varying parameters within appropriate range (i.e. this can also be seen as a sensitivity analysis of a function or the behaviour of a function in relation to the impact of some parameter variations).

The given ADAS function is an ACC (advanced cruise control) that controls the speed of a vehicle not only to a fixed set speed value, but also in accordance to the distance and speed of a vehicle in front of the ego vehicle. The distance and speed is measured by means of a radar sensor (and the corresponding reflection signal) in the ego vehicle.

The scenario should be given as shown in Figure 6. The green car on the second lane has a given target speed for the ACC and follows the blue car that is slower than this target speed. Because the third lane is occupied the green car cannot overtake, but must follow the blue car. It would be interesting to see what happens in the ego car, if the speed of the blue car changes over time. Is the controller of the ACC implemented in a way that it is safe (i.e. it keeps a certain distance to the blue car) and it is comfortable as well (i.e. it does not break or accelerate abruptly)?

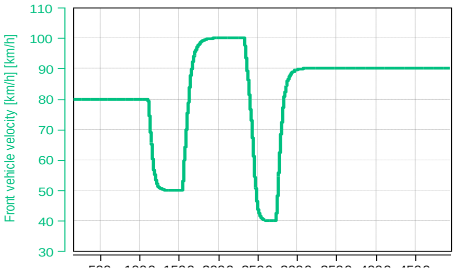

Figure 7: Reference speed profile for the evaluation of the sensitivity

Finally, it would be also interesting to see, if the fuel efficiency of the car has a significant dependency on the controller parameter settings. There is the request, that the fuel consumption shall not exceed a certain value when accelerating – this usually should go along with the need for a smooth acceleration one could argue. The target definition – we could call this also KPI (key performance indicator) – therefore consists of expected values for headway distance of the blue car, maximum acceleration (or jerk) and maximum fuel consumption.

The parameters (tuning parameters) that have an impact to this KPI's could be identified within 2 categories.

First category are the parameters of the assumed PID-controller for the ACC function that tries to keep the target speed and keep the minimum distance to the preceding car (headway distance).

Second category are some of the parameters of the aforementioned scenario, e.g. speed of preceding car (speed1), speed of following ego car (speed2), initial headway distance and others.

To identify the sensitivity of the KPI's in relation to the tuning parameters a reference speed profile of the preceding car shall be given (compare Figure 7). This could be the output of an observation measurement on a public road.

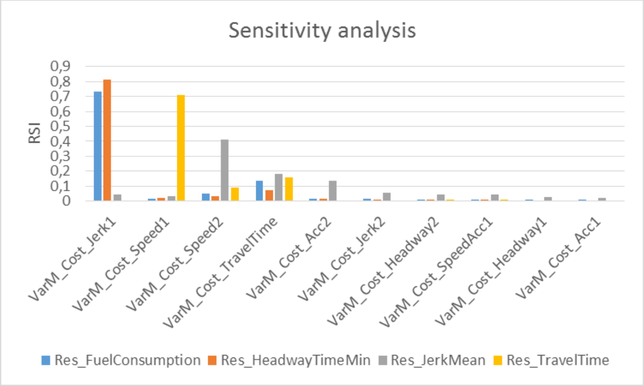

Figure 8: Parameter sensitivity of 10 tuning parameter with respect to 4 given KPI's

The test planning for the given ACC example was finally created with an interactive DoE approach (COR DoE, [9]) with 10 tuning parameters of the second category (since it was not clear from the beginning, how many parameters could have a significant influence to the KPI's on the one hand and the parameters of the PID controller should be fixed in first iteration and varied in an outer loop on the other hand, in order to find the best parameter combination of the PID controllers separately from the first analysis). 4 KPI's have been measured to identify the impact of the tuning parameters. The parameters of the PID controller have been varied in an outer loop.

The initial test design of 40 test runs was online automatically adapted to 520 test runs to get reliable sensitivity models in the area of interest. Figure 8 shows the Relative Significance Indicator (RSI) for the 4 KPIs with respect to each of the 10 investigated tuning parameters. The RSI is a statistically calculated value normalized between zero and one that indicates the influence of each of the 10 tuning parameters to the 4 KPI's. The higher the value of the respective KPI for a specific tuning parameter, the higher is the influence of this tuning parameter to the value of the KPI. The calculation of this influence was done by means of a Robust Neural Network (RNN, [14]) that is able to statistically evaluate the influence of input parameters to output of the neural network.

The first 5 parameters in this example have a considerable influence on the overall behaviour of the ACC function with regard to the chosen KPI's. For the outer loop of optimizing the parameters of the PID controller, the first 5 parameters have been varied and the rest of the 5 parameters have been kept constant.

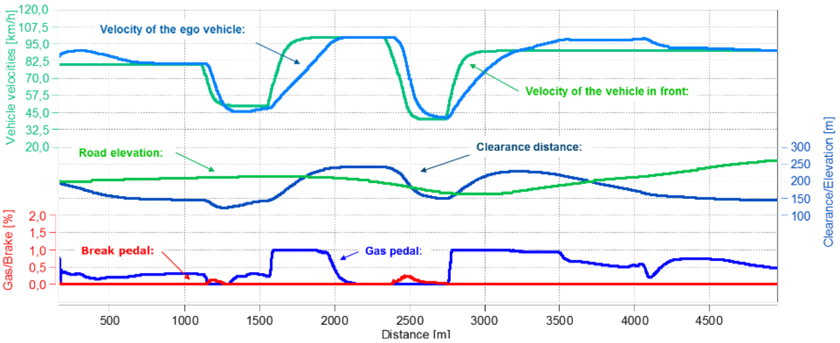

Figure 9: Vehicle signals of the optimized ACC controller, which fulfils the requested KPI's in the best manner for all selected corner cases

Without describing the well-known “model based tuning process” in detail [11, 12, 13], Figure 9 shows the final result of the of the optimized ACC-controller: The upper two signals represent the preceding and ego vehicles velocities respectively. The preceding vehicle’s velocity profile was fixed for the whole tuning task and the ego vehicle’s ACC controller was optimized in order to assure that the safety, fuel consumption and comfort KPIs are met. The next two signals represent the road elevation profile and the clearance between the ego and the preceding vehicle. We can see that the controller settings are chosen in such a way that robustness is assured and out-side disturbances coming from the road elevation are successfully handled. Finally, the two remaining signals represent the accelerator and brake pedal position percentages, and we notice that the controller settings minimize the braking energy and jerk, leading to a very efficient and comfortable ride.

For the evaluation of the optimized controller settings, we compared the behaviour and consumption of the ego vehicle with respect to the preceding vehicles’ velocity profile. The controller was able to achieve approximately 18% decrease in fuel consumption whilst still maintaining a smooth and safe ride and keeping the headway distance within the requested safe limits.

7. Summary

The paper gives an overview about the building blocks that are involved for the verification and validation of highly automated vehicles and highly automated cyber physical systems. Some of these building blocks have been described in more details in order to give the reader the chance to get insights into the complexity of the development and testing process for automated vehicles. The applied methods and results gained so far in the project lay the basis for further investigation and collaborations in this sophisticated technology field. It can be expected that only common efforts of many different stakeholders – from OEM’s to suppliers, SW-industry, technology providers and academia, as well as legal authorities and certification organizations – will finally lead to automated vehicles and cyber physical systems that are at least as save as human operated vehicles and CPS – and may even go beyond that. The presented example also shows that developing and testing of these complex systems are often interwoven with each other and make things even more complex. But there is a high chance that combining and adapting well proven and existing V&V methods with new methods and technologies developed within this and other research project will provide effective means to tackle the problem and make huge steps forward in the direction of self-driving cars.

To enable the collaboration of the involved stakeholders it is also a prerequisite to establish sufficient standards that allow the exchange of ideas and methods amongst each other. For the exchange of models, testing scenarios, testing methods, sensor models, stimulation devices and test execution platforms, standardized interfaces, description languages and standardized exchange formats are needed.

This will have to be driven and supported not only by the 71 partners of the ENABLE-S3 project, but needs common efforts and motivation of the whole industry that deals with highly automated systems. Highly individual and isolated approaches will most likely fail to succeed in this ground-breaking new field of technology.

However, this project and many other ongoing joint research project show that the industry, academia and legal authorities are well aware of this potential risk and are willing to cooperate on a broad basis, share their insights and know-how and work together in a good spirit.

This common work will finally help to achieve the vision of safer and cleaner transport that is highly or fully automated.

8. Acknowledgement

This work has been conducted within the ENABLE-S3 project that has received funding from the ECSEL Joint Undertaking under Grant Agreement no. 692455. This Joint Undertaking receives support from the European Union’s HORIZON 2020 research and innovation programme and Austria, Denmark, Germany, Finland, Czech Republic, Italy, Spain, Portugal, Poland, Ireland, Belgium, France, Netherlands, United Kingdom, Slovakia, Norway.

9. References

[1] International Organization for Standardization, „ISO 19365:2016,“ Internation Organization for Standardization, 08 August 2016. [Online]. Available: https://www.iso.org/standard/64702.html. [Zugriff am 08 August 2017].

[2] VIRES_OD Simulationstechnologie GmbH, „opendrive.org,“ 08 August 2017. [Online]. Available: http://www.opendrive.org. [Zugriff am 08 08 2017].

[3] VIRES_OS Simulationstechnologie GmbH, „openscenario.org,“ 8 August 2017. [Online]. Available: http://www.openscenario.org. [Zugriff am 8 August 2017].

[4] Watzenig, Daniel, and Martin Horn, eds. Automated Driving: Safer and More Efficient Future Driving. Springer, 2016.
[5] Winner, Hermann, and Walther Wachenfeld. "Absicherung automatischen Fah-rens, 6." FAS-Tagung München, Munich 2013.
[6] Wachenfeld, Walther, and Hermann Winner. "Die freigabe des autonomen fah-rens." Autonomes Fahren. Springer Berlin Heidelberg, 2015.
[7] www.enable-s3.eu - European Initiative to Enable Validation for Highly Auto-mated Safe and Secure Systems.
[8] www.bast.de - Bundesanstalt für Straßenwesen (BASt).
[9] Dr.-Ing. Armin Engstle, Dipl.-Ing. Helmut Assmayr, Dr. Georg Schwab, Softwareentwicklung für autonome Fahrfunktionen,
[10] www.sae.org - Society of Automotive Engineers
[11] Rainer, A., H. M. Koegeler, and D. Rogers. "Iterative DoE—Improved Emission Models and Better Optimisation Results Within a Shortened Measurement Time." 2nd Biennial International Conference on Powertrain Modelling and Control, Brad-ford, UK. 2014.
[12] M.Sc. A.Ravi, Dr. H.M. Koegeler, M.Eng. Takayuki Miyata, M.Sc. A.Saroldi, Tool Chain for development of ADAS systems, CRF, Italy,
[13] Schick, Bernhard, et al. "Model-based steering ECU calibration on a steering-in-the-loop test bench." 6th International Munich Chassis Symposium 2015. Springer Fachmedien Wiesbaden, 2015.
[14] Keuth N, Koegeler H, Fortuna T, Vitale G. DoE and beyond the evolution of the model based development approach how legal trends are changing methodology. Conf. Des. Exp. Powertrain Dev., 2015.
[15] Conference of European Directors of Roads - CEDR, “Safe distance between vehicles”, 2010.
[16] Holzinger Juergen; Schloemicher Thomas; Bogner Erik, Objective assessment of comfort and safety of advanced driver assistance systems, FISITA 2016 World Automotive Congress, Busan, Korea
[17] http://ns.umich.edu/new/releases/24866-new-way-to-test-self-driving-cars-could-cut-99-9-percent-of-validation-costsClark, T., Woodley, R., De Halas, D., 1962. Gas-Graphite Systems, in *“Nuclear Graphite”.* In: Nightingale, R. (Ed.). Academic Press, New York, pp. 387.