Model-based safety validation of the automated driving function highway pilot

Presenter:

Halil Beglerovic,
Research Engineer / Research and Technology

Co-Authors:

Abhishek Ravi,
Application Engineer / Calibration Applications

Niklas Wikström,
Simulation Engineer / Powertrain and Hybrid Systems

Hans-Michael Koegeler,
Principal Engineer / Calibration Applications

Andrea Leitner,
Project leader / Research and technology

Jürgen Holzinger,
Project manager ADAS

AVL List GmbH

Abstract

In the development cycle of an Advanced Driver Assistance System (ADAS) emphasis is always placed on passenger safety, which directly correlates to adequate and extensive testing and validation procedures. However, testing and validation of ADAS systems is not a simple task and cannot be performed in a similar manner as conventional testing approaches. The main challenges is the vast amount of scenarios and environment parameter variations that might occur during operation. Current validation and testing procedures mostly rely on real world tests conducted on roads; however, because of the cost and complexity, these tests are not exhaustive and compromises on the scenario types and the number of considered parameters are made. In this paper we propose a model-based validation approach performed in a SiL (Soft-ware in the Loop) environment to validate an ADAS system under various conditions and the proposed methodology is presented on a Highway Pilot case study.

Preliminaries

In this document we are going to use the following terminology:

- Test scenario: *A generic traffic situation i.e. crossing, roundabout, highway driving etc. (including variation points for certain environment parameters).*

  For example: Driving on a high way within one lane

- KPI: *Key Performance Indicators.*

- Test case: *traffic scenario with defined discrete parameters – e.g. the number of other traffic participants or the number of lanes + defined KPIs in order to assess the function. Continues variables like starting velocities and positions of the traffic participants are still given as valid ranges only. For example: Driving on a 2 lane road behind a proceeding vehicle in a distance between 80 to 130 m with velocities of the proceeding vehicle between 40 and 150 km/h with defined KPIs like minimum clearance distance happening in the next 5 km of driving.*

- Test run: *Test case with defined values for all (also continuous variable) environment parameters. For example: Driving on a 2 lane road behind a proceeding vehicle in a distance of 80 m with a defined velocity profile of the proceeding vehicle stepping from 80 to 50 km/h, during light rain and slippery road with defined μ of 0.5. The KPIs remains like given in the test case.*

- ADAS: *Advanced Driver Assistance System. Typically below SAE-level 3*

- ADF: *Automated Driving Functions.*

- ENABLE S3: *European Initiative to Enable Validation for Highly Automated Safe and Secure Systems.*
- ACC: *Adaptive Cruise Control,* LKA: *Lane Keep Assist,* LCA: *Lane Change Assist.*
- ESP: *Electronic Stability Program,* EBA: *Emergency Brake Assistant.*
- Highway Pilot: *Combination of ADAS functions like ACC, EBA, and others for level 4 automation on highways.*
- MIL: *Model in the Loop,* HIL: *Hardware in the Loop,* VIL: *Vehicle in the Loop,* SIL: *Software in the Loop.*
- SUT: *System Under Test.*
- DOE: *Design of Experiments.*

1 Introduction

Technologically the development of automated driving functionalities is satisfactorily understood and witnessed by more than 1 million 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 ADF, have slowly become an irreplaceable part of the everyday driving experience. Nevertheless, Watzenig et.al. [1] 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 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. [2] as well as Wachenfeld et.al [3] 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. 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 ADAS system. However, even when utilizing these collections of tests one cannot prove that the ADAS system will not fail in a test scenario that was not previously covered. 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 function.

Taking further into account the high number of vehicle variants and software versions, it becomes obvious that new approaches are required to validate automated vehicles within a reasonable time period at reasonable costs. 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.

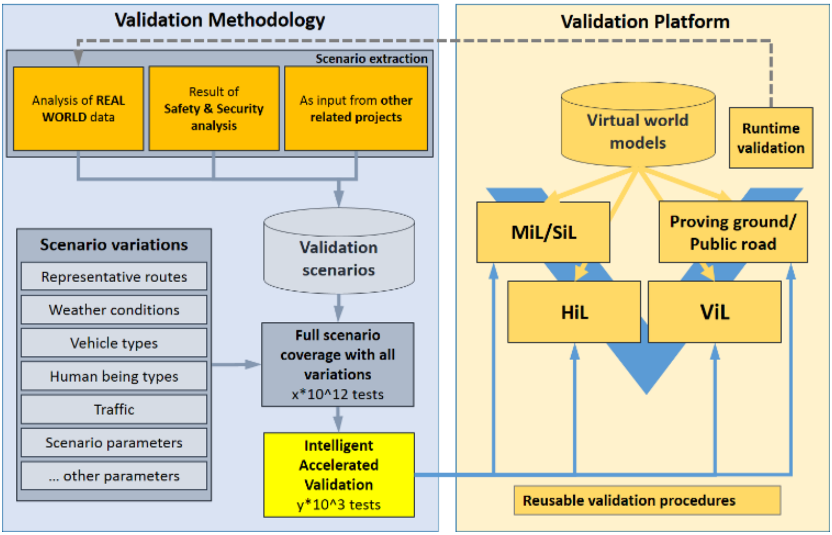

Figure 1: Modular Validation and Verification Framework envisaged in the ENABLE-S3 project [4]

Figure 1 shows the approach of a European research project that was started in May 2016 called ENABLE-S3 [4]. The main objective of the ENABLE-S3 project is to develop a modular framework for validation and verification of automated systems.

Because of the large scope and complexity the problem has been split into two parts. The validation methodologies on the one hand side describe the necessary steps and re-search on data acquisition and storage, scenario and metrics selection, as well as test optimization methods. The validation platform on the other side focuses on reusable validation technology bricks, which are able to seamlessly support various development stages (MIL, HIL, VIL, and Proving ground / Public road). By combining both parts and their respective technology bricks the project aims to achieve testing with at least 50% less effort compared to the actual established testing methods which are estimated with 10^{12} test runs.

This paper will show the first validation approach by selecting one of maybe 1000 relevant traffic scenarios for a Highway Pilot. By systematically varying scenario characteristics it is expected to achieve a better coverage compared to the classical approach. However, this approach is only feasible if adequate virtualization tools and MIL/SIL environments are available.

According to accident databases like “German In-Depth Accident Study” (GIDAS) [5], most highway accidents in recent years have occurred when driving in the same direction on the same lane, at sunny weather and dry road conditions.

Consequently, the first automated driving use cases focus on Highway Pilot functionality [6] – allowing a SAE level 4 automation [7]. In general, the Highway pilot is a combination of several complex ADAS functions such as: ACC, LKA, and LCA. Each of these ADAS functions individually and their integration presents a challenge for validation and verification. In the scope of this paper we will focus on the ACC part of the Highway Pilot in order to focus on the “following behind a car in the same lane” scenario with the highest accident probability. To keep the confidentiality of manufacturers, we put ourselves in the position of an ADAS-Function supplier and develop it from the ground up starting from the development to the first tuning and validation. The ACC control strategy will be discussed in more detail in the following section.

2 ACC – Controller Overview

The objective of the investigated control strategy is to allow fuel-efficient, comfortable and safe longitudinal speed profiles for the ego vehicle on a highway road, presenting the first and fundamental part of a Highway Pilot. To this end, a model predictive control (MPC) strategy is employed which adaptively controls the acceleration of the vehicle based on information from the prediction models. These models recurrently predict the motion of the ego vehicle as well as a preceding target vehicle over a prediction horizon of 20 seconds. This information is then used as inputs to a quadratic programming optimization problem, which aims to

$$
\begin{array}{ll}
\text{Minimize} & \boldsymbol{x}^T Q \boldsymbol{x} + \mathrm{q}^\mathrm{T}\mathbf{x} \\
\text{Subject to} & \text{Vehicle Constraints} \\
& \text{Traffic Speed Limits} \\
& \text{Headway Limits}
\end{array}
\tag{1}
$$

The optimization problem is solved every 500ms to obtain desired vehicle accelerations. In equation (1), $\boldsymbol{x}$ denotes the state of the system, which includes the ego vehicle's longitudinal headway distance, speed, acceleration and mechanical jerk. The cost matrix $\boldsymbol{Q}$ and cost vector $\boldsymbol{q}$ describe the weightings between the different state variables. These parameters were tuned in an initial phase for a single route with fixed traffic and environment conditions and will be discussed in detail in the next section.

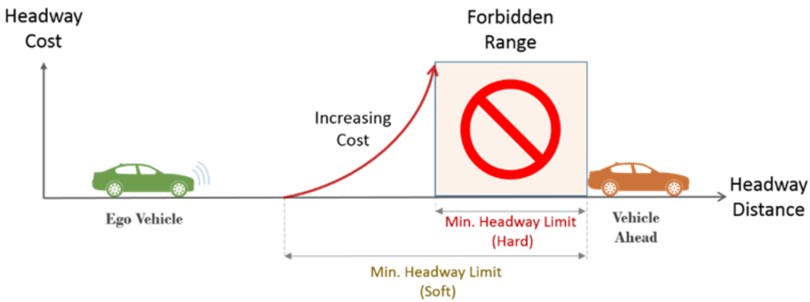

Figure 2: The minimum inter-vehicle distance is given by two headway time limits: the first soft limit may be violated at a cost and the second hard limit that must never be violated.

The vehicle is allowed to adapt its headway distance to the preceding vehicle within a predefined, velocity-dependent range. This is referred to as a flexible spacing policy. Figure 2 shows how the minimum distance is defined by two headway limits: a hard and a soft limit. The hard limit must never be violated, whereas the soft limit can be violated but at a cost. This cost grows quadratically approaching the hard limit.

The maximum distance, illustrated in Figure 3, is also composed of two components. The first component is a predefined limit, dominated either by a fixed distance or a velocity-dependent distance based on the preceding vehicle's velocity. The second component is proportional to the preceding vehicle's current speed, scaled by a 'catch-up' factor defined to be greater than one. The first component dominates when the ego vehicle is within the desired headway range, whereas the second component dominates when the vehicle is far behind. Neither of the constraints are allowed to conflict with the speed limit on the route, and lagging behind the limits is associated with a travel time cost.

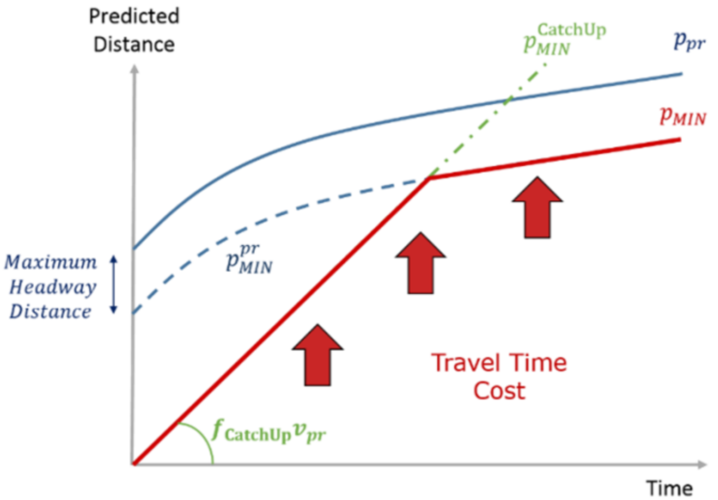

Figure 3: The maximum inter-vehicle distance $\boldsymbol{p_{MIN}}$ is defined by two limits: one limit dominates when the ego vehicle is located far behind the preceding vehicle ($\boldsymbol{p_{MIN}^{CatchUp}}$), while the other dominates after a predefined headway range has been reached ($\boldsymbol{p_{MIN}^{pr}}$).

3 Model-based Tuning

In order to reach a target behaviour of the ACC, or in more general terms the System under Test (SUT), the relevant influencing parameters have to be determined and varied systematically. Typically the first development environment in the development process is a Model in the Loop System (MiL), where the ACC function is simulated together with the environment.

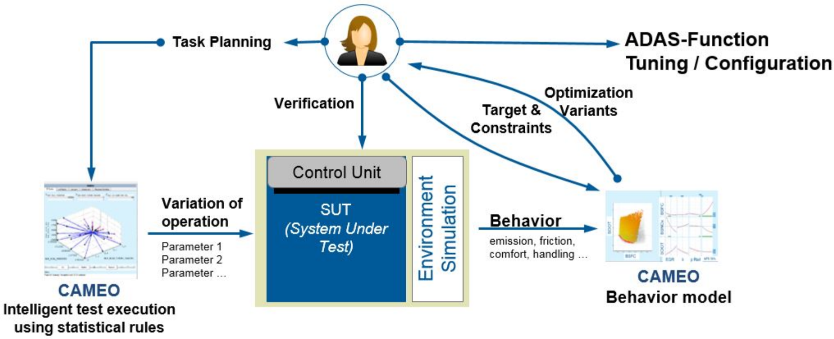

Figure 4: Model-based tuning of a System under Test (SUT)

In general, this tuning task is carried out by an expert operator, who based on his experience varies the most relevant parameters until the target behaviour is met. The results from such a subjective tuning process depends on: start values, step size, and number of considered parameters, and so on. Without a good understanding of the underlying correlations between the parameters, a perceived optimum behaviour could easily represent a local optimum. To improve that, a statistical Design of Experiment (DoE) based tuning methodology is presented in Figure 4, which tries to provide a correlation of the input parameters and the Key Performance Indicators (KPIs) using the least possible number of tests. Using an automated test procedure, all the relevant input parameters are varied simultaneously to identify the system behaviour. The observed behaviour of defined Key Performance Indicators (KPIs) is used to build behaviour models, which allows predictions of the system behaviour in the whole space defined by the variation parameters and some limited extrapolation depending on the model quality. The method assures that a global optimum is found, the parameter space is optimally covered, and the efficiency as well as traceability of the tuning-decisions is ensured. This statistical approach helps to understand the parameter influence and correlations, which is very beneficial in cases with a large number of input parameters. In such cases, the parameters with lower influence can be kept constant or excluded which reduces the tuning effort. Furthermore, the behavioural models built between the inputs and KPIs can be used for optimization tasks and trade-off analysis providing an additional tool for the adequate tuning of parameters.

The development environment for the SUT consisted of dynamic vehicle simulation program, Matlab-Simulink for the development of the ACC and AVL Cameo for the test automation, tuning and optimization.

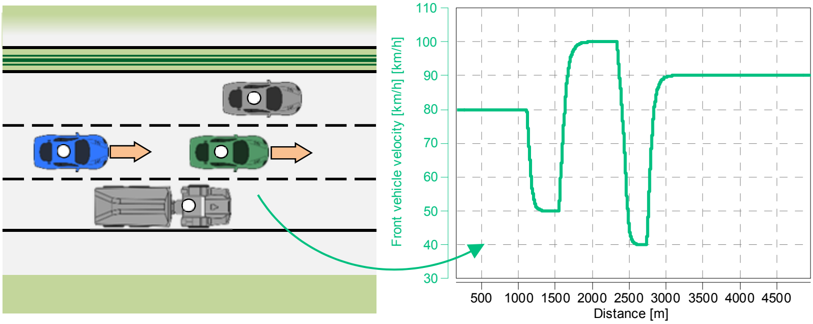

Figure 5: a) Scenario selection, b) Reference maneuver

3.1 Scenario Selection and Reference Maneuver

Figure 5 shows the scenario selected for the tuning of the ACC function. It consist of an ego vehicle (blue) following a preceding vehicle (green) using a model predictive ACC. The preceding vehicle has a predefined behaviour as shown in Figure 5 b) and the controller is tuned to satisfy several KPIs with respect to the preceding vehicle behaviour. The tuning task represents the first effort conducted for the new controller and it is placed early in the V development cycle.

3.2 Tuning Targets (KPIs)

For the tuning task of the ACC controller the following KPIs have been chosen:

- Safety – The controller should not violate the hard constraints regardless of the maneuver. As shown in Section 1 the controller has a hard constraint for the headway time. This constraint must not be validated in order to leave enough time for the vehicle to come to a complete stop in case of an accident.
- Fuel Performance – The controller should minimize the fuel consumption measured in liters per 100km and the influence of sudden changes in the preceding vehicle's velocity or maneuvers should be kept low. The fuel consumption when the ego vehicle perfectly follows the preceding vehicle with a fixed headway time is considered as the baseline.
- Driver Comfort – The controllers should maximize the driving comfort by reducing the integrated jerk. Similarly as above, the behavior of the preceding vehicle will be smoothed out and the controller will try to minimize braking while maintaining a safe and smooth ride.

3.3 Design Variables Sensitivity Analysis

As we are dealing with a new controller function without a priory knowledge of the behaviour, we have choosen 10 parameters and conducted a sensitivity analysis. Figure 6 shows the Relative Significance Indicator (RSI) for the 4 KPIs with respect to each of the 10 investigated tuning parameters. To calculate this the sensitivity algorithm analyzed the contribution of each individual variation channel to improve the statistical quality of the behavior model by reducing the root mean square error (RMSE). It lies between 0 and 1, denoting, how strong the remaining RMSE is reduced, when taking the corresponding variation parameter as additional input parameter to explain the observed behavior. A Robust Neural Network (RNN, [11]) is used as modeling base, which also considers nonlinear influences and interactions correspondingly. We conclude that only the first 5 parameters have a considerable influence on the overall behavior of the system. Therefore, for further tuning the rest of the parameters were kept constant.

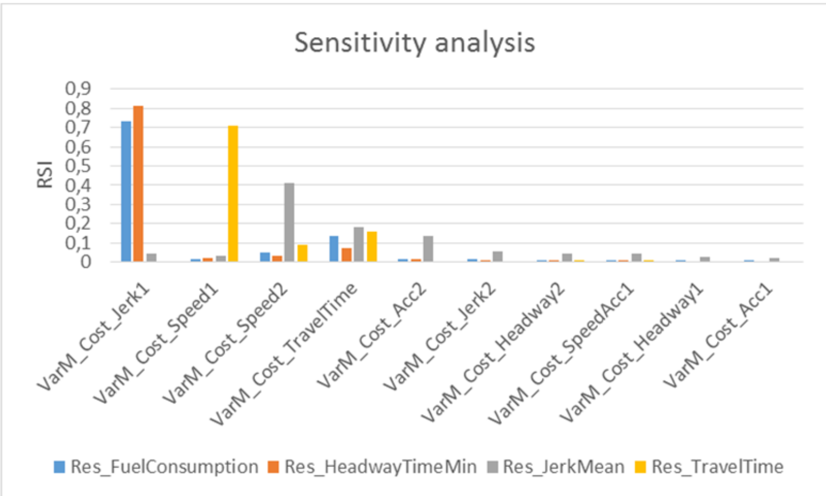

Figure 6: Parameter sensitivity analysis

3.4 Tuning Results

The tuning of the controller has been conducted with an interactive DoE procedure (COR DoE, [9]). The start design of 40 test runs for the 5 dimensions was automatically extended, in the area of interest, to 520 test runs in order to get reliable models in the area of interest, which is not known upfront. Without describing here the well-known "model based tuning process" in detail [8, 9, 10], Figure 7 shows the resulting tuned ACC-behaviour: 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 tuned 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 outside 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 are minimize the braking energy and jerk leading to more efficient and comfortable ride.

For the evaluation of the tuning, 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. In the next section, a validation of the tuned control function will be performed in order to determine the robustness of the chosen parameters with respect to different behaviours of the preceding vehicle.

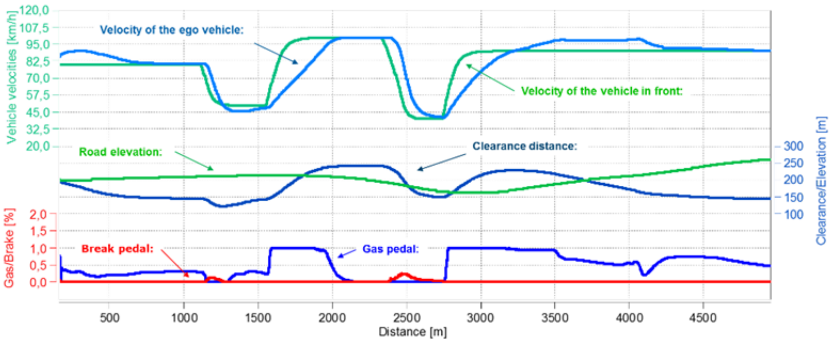

Figure 7: Tuning results for the reference scenario

4 Model-based Validation

After successfully tuning the SUT, a validation procedure is needed in order to prove that satisfactory behaviour and robustness is assured in a wider range of maneuvers. Figure 8 shows an overview of the validation task and we notice a considerable similarity to the tuning task discussed previously. In this paper, we aim to prove that the same principles of model-based tuning can be transferred to the model-based validation of Automated Driving Functions. The main difference between the tuning and validation is that now the external or environmental parameters are varied in order to validate the internal or controller parameters. Similar to the tuning task, a parameter space coverage is determined using DOE after which behaviour models are built for all KPIs. The most important benefits of this approach are [9]:

1) Compared to “Full factorial grid approach” the behaviour of safety-critical KPIs can be predicted in with an order of magnitude less simulation runs, giving complete results in the investigated ranges.
2) Interactions and significance of the parameter changes are fully considered when searching for safety-critical situations.
3) The highly automated process is much faster than a manual one and the human interaction happens on the level where decisions are made. In addition, a new interactive DOE procedure is available [11], which fills in points automatically after an initial sampling in order to improve the accuracy of the models in regions of high interest.

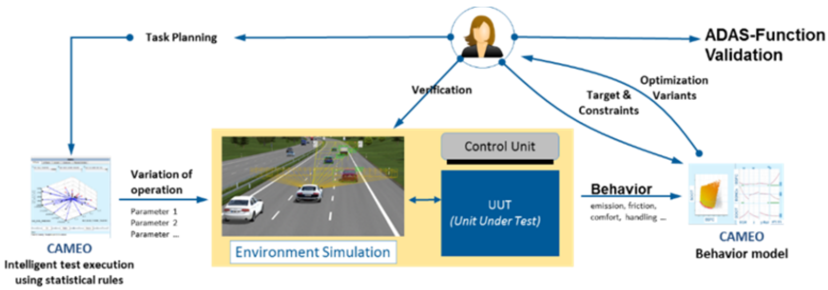

Figure 8: Model-based validation of a System under Test (SUT)

By selecting appropriate KPIs, it is possible to locate safety-critical behaviour regions in-side the parameter space by optimizing the models built in the previous step. Faulty behaviour of an ADF in a test run can be defined as the behaviour occurring when a critical criteria is not satisfied, e.g. the minimum clearance distance was breached. For a function developer it is of high interest to investigate the region in more detail and to get a better understanding of the faulty behaviour. This can be done automatically using the interactive DOE procedure.

4.1 Scenario Selection for Validation

The scenario selected for the validation of the Highway Pilot can be seen in Figure 9. In this scenario the ego vehicle (blue) is driving with the desired cruising velocity and a vehicle (green) is cutting in from the right side with a much lower speed. After cutting in, the preceding vehicle's velocity is changing following a sinusoidal trajectory where the mean velocity, amplitude, and period are varied as validation parameters. The length of the test run is fixed to 3km. In addition, this type of scenario represents the second most likely accident type defined by the "German In-Depth Accident Study" (GIDAS) [5].

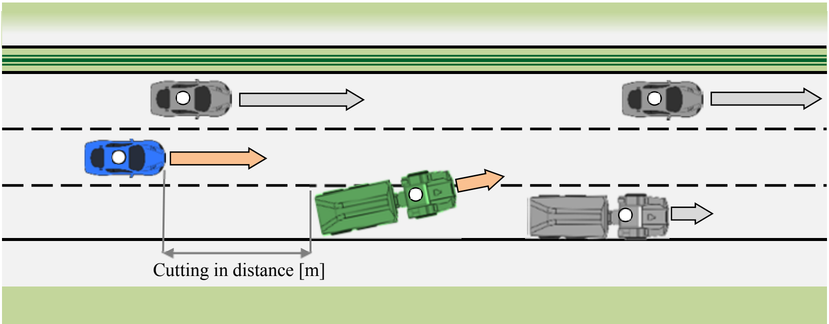

Figure 9: Selected scenario for validation

4.2 Validation Parameters and KPIs

The parameters of the scenario can be split in two parts. First, the parameter set by the driver:

- ACC target velocity (target cruising velocity) – If no preceding vehicle is detected, the ACC will accelerate to the target velocity.
- Maximum headway time to next vehicle – If a preceding vehicle is detected the ACC is going to keep the headway time below the maximum value, i.e. keeping a steady but safe distance from the vehicle in front.

Second, the environment parameters shaping the test case to a defined test run:

- Distance to the preceding vehicle – Distance between the ego and preceding vehicle at the cutting-in time instant.
- Road friction coefficient

- Amplitude, frequency, and mean velocity of preceding vehicle – In order to validate the ACC under a wide range of behaviors the preceding vehicles´ velocity was changed following a sinusoidal trajectory where the amplitude, frequency and mean are defined as input parameters for the validation.

Table 1 gives an overview of the validation parameters and their corresponding ranges for the selected example. The scenario was simplified for a better understanding in the following way: The cutting-in distance was fixed to a reasonable borderline value (for shorter distances the truck driver would be responsible for an accident). We did not consider variations in the road friction and the road gradients turned out to have no influence as long as the road friction is sufficient.

Table 1: Validation parameters

	Start velocity of the maneuver	fixed values	150 km/h
Selected by the Driver	ACC target velocity		150 km/h
	Maximum time gap		2 s

		min	max	Start value
Maneuver - variation parameters	Distance to the incoming vehicle	fixed values		80 m
	Road friction			1
	Road gradient			0%
	Average traffic velocity	40 km/h	140 km/h	80 km/h
	Sine velocity amplitude	- 20 km/h	20 km/h	0 km/h
	Sine velocity time period	10 km/h	40 km/h	10 km/h

Figure 10 shows one example taken from the test runs. The ACC set velocity was 150km/h and at time instance of 40s the ego vehicle's sensor detects a vehicle cutting in with a velocity of 65km/h. Finally, the velocity of the preceding vehicle is changing following a sinusoidal trajectory with an amplitude of 20km/h and a period of 25 seconds.

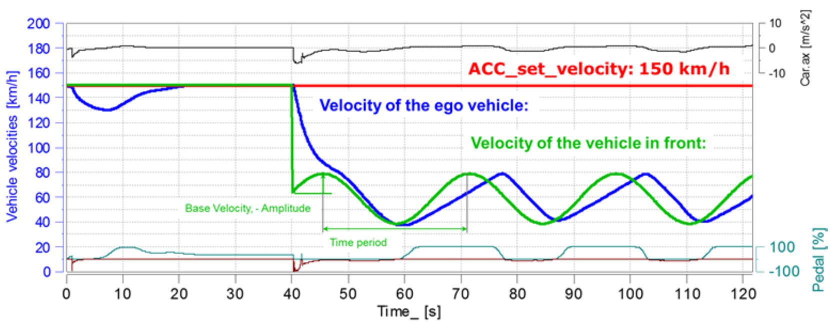

Figure 10: Example validation test run

For the validation, the following KPIs were considered:

- Minimum clearance distance between ego and preceding vehicle.
- Minimum headway time – representing the minimum time that the ego vehicle would need to reach the preceding vehicle's current position.

The KPI values for the example test run are shown in Figure 11.

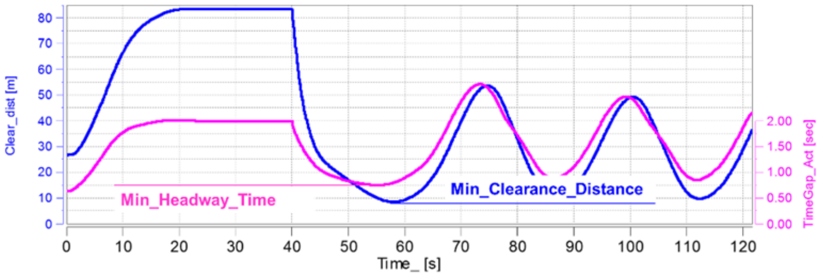

Figure 11: Clearance distance and time gap KPIs for the example test run

4.3 Validation Results

In order to validate the ACC for this specific scenario we first need to define a criterion that denotes a critical situation. The *"Safe distance between vehicles"* published by the *Conference of European Directors of Roads* [12] gives an overview of existing laws regarding the minimum clearance distance and headway time. In the publication we find that the minimum headway time defined by the Austrian law is 0.4 seconds regardless of the scenario. Following the regulations as guidelines we define that a dangerous test run ranges between 0.4s – 0.8s headway time and clearance distance ranging from 3 m to 15m, where scenarios with lower values represent crashes.

In this regard, we performed a multi objective optimization on the KPI models, in the dangerous ranges defined above, in order to find the most critical environmental parameters for the specified scenario. Figure 12 a) shows the results of the optimization. The optimization variables are the minimum clearance distance and the maximum breaking energy. The optimization leads to a test run with the strongest braking effort and the minimum clearance distance still considered safe. From Figure 12 a) we can see that the selected Pareto optimum solution dominates all other dangerous test runs and that a more critical solution does not exist except those with a lower predicted minimum clearance distance of 3 m, which were rated as "crashes". The reason for this borderline lies in a standard deviation of 1 m for the behavior model for the "minimum clearance distance". From Figure 12 b), where a verification simulation for this setting is shown, we determine that the minimum clearance distance and time for the worst test run are 2.2 m and 0.45s respectively.

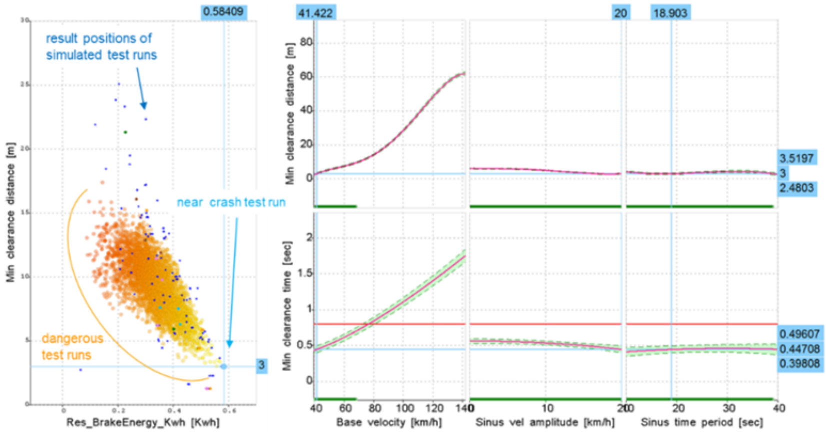

Figure 12: a) Pareto front for the critical scenario, b) Parameter influence on safety

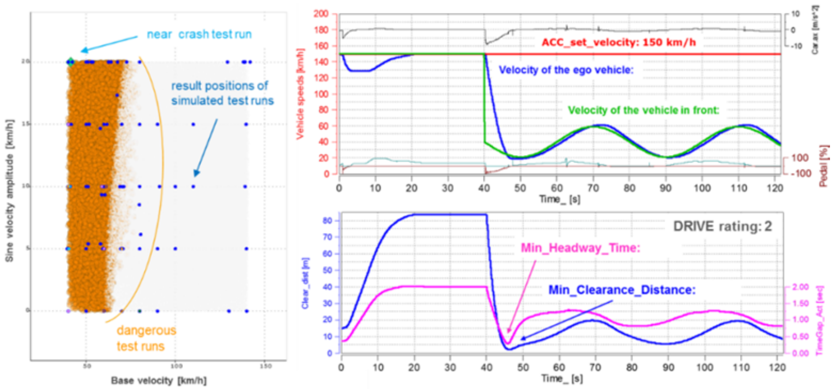

Figure 13: a) Dangerous test runs regarding the Sine parameters b) Most critical test run

In addition, Figure 13 a) shows all the dangerous test runs with respect to the sinusoidal velocity amplitude and the base velocity. We can see that these two parameters have the main influence on the minimum clearance distance especially when the amplitude of the sine is negative leading to lower velocities. This effect can be seen in Figure 13 b) where the most critical “near crash test run” is presented.

Thus we can state, that for the test run considering parameters in table 1 with an ACC target velocity of 150 km/h under

- dry conditions, on flat road, with a cutting in distance of 80 m,
- a sinus time period between 10 and 40 seconds
- with an velocity amplitude within +/-20 km/h
- a base velocity of the “cutting in vehicle” of more than 41 km/h,

for any combination of these parameters, the system will react safe in terms of “crash avoidance” (minimum clearance distance less than 3 +/-1 m is not reached).

Furthermore, as long as the base velocity of the cutting in vehicle is above 78 km/h (no dangerous test run results exists with higher velocity values) the legal limits of at least 0.8 seconds headway time is kept under all conditions.

One difference to a conventional testing is that we can draw these conclusions based on 210 simulation runs while considering 5 influencing parameters. Additionally, in this test case we did not take in consideration friction and road gradient; however, conventional testing methods would still needed 693 simulation runs, when changing each of the remaining 3 parameters in steps of 11 Average traffic velocity values, 9 Velocity amplitude values and 7 Velocity time period values.

The information of these borderline test runs, gathered out of simulations, are useful in later testing and validation stapes which can be performed in development environments like a “Driving cube” or on a “Proving ground” where a high integration between the function and vehicle is possible.

4.4 Discussion and future work

The “near crash test run” shown in Figure 13 b) gives also another aspect, which will be part of near future work: Passengers sitting in such a car will have a perceived safety evaluation of the test run.

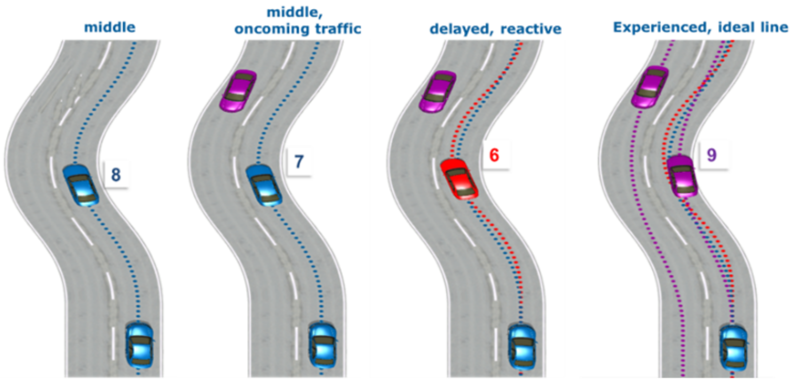

Figure 14: Rated passenger feeling, when driven in automated way, taking the other traffic participants into consideration [13]

In order to quantify the effect of the ADF to such subjective evaluation “perceived safety” and comfort metrics are introduced [13]. Figure 14 shows the perceived safety and comfort of passengers in a vehicle rated from 1 to 10 when driving on a rural road. This means that at any time in all possible situations the distances between the traffic partners must not be undershoot. But also for one and the same trace of an ego vehicle – keeping sufficient distances to other traffic participants from safety point of view – the perceived safety will depend strongly on the oncoming traffic.

Another important issue is that the most relevant scenarios and their respective test cases including KPI´s and realistically relevant parameter ranges for the function under consideration have to be selected.

The current tuning and validation example for the model predictive ACC was carried out on a single highway scenario and not all the critical parameters were considered. For a more extensive validation all possible scenarios and all relevant test cases need to be taken into account. In the end, all traffic situations have to be handled in proper way by the ADF. This immediately leads to the question of what it means “all” and what “to be handled in a proper way”.

First, the most relevant scenarios and their respective parameters for this function have to be selected. In order to do so, probability distributions for the scenarios and their parameter value ranges have to be collected. Figure 15 shows the high level architectural framework elaborated within the ENABLE-S3 project. It provides a high level overview on the technology bricks required to validate an automated system and shows how the proposed approach of this publication (marked as “Test definition and control” + “Evaluation”) is embedded in the overall big picture of automated system validation.

Together, the systematic scenario selection and the proposed validation method are going to provide a comprehensive and extensive validation. As a future work, we will investigate the scenario selection aspects in more detail and will focus also on additional aspects of evaluation (e.g. perceived safety for the driver/passenger to rate also the acceptance for these systems).

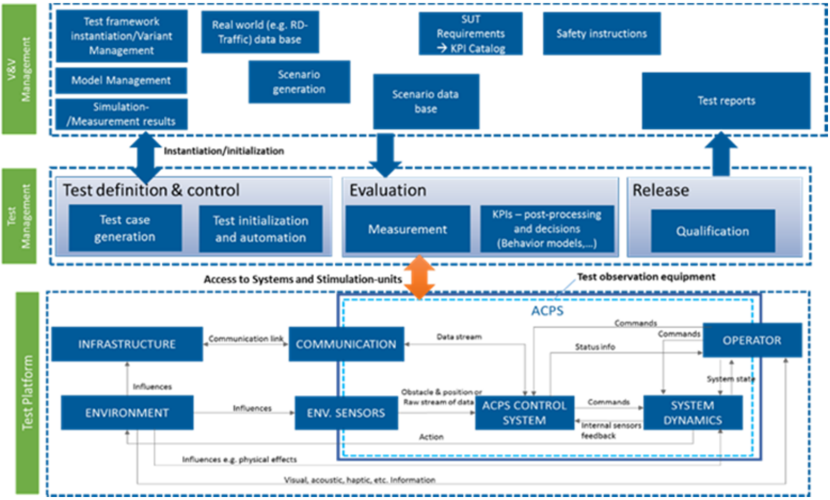

Figure 15: Validation and Verification framework of the ENABLE S3 Project

5 Conclusion

In this paper we presented a comprehensive introduction for the development, tuning and validation of a simple Automated Driving Function – ACC of a Highway pilot. The emphasis was put on the validation methodologies using as an example one test scenario. We have shown how principles of the well understood tuning task could be carried onto the validation. An ACC function was developed from the ground up presenting the first part of a Highway pilot. The tuning of the controller was done on a fixed scenario and the ACC was then validated on a test case, which was shaped in its appearance in 210 test runs. The resulting behaviour models allow us to specify areas for

- robustness and satisfactory response of the controller as well as
- areas, where legal limits are already violated for short time and further
- its borderline performance for the near crash situation.

We have shown that the ACC is able to stay above the absolute minimum safety margins for nearly all test runs in the specific scenario and in future work an additional perceived safety rating KPI will be taken into account.

It is planned to further develop this approach within the ENABLE S3 project. Thus a first methodical step towards a valid base for quantifying the remaining risk of a certain AD-driving function in its later lifetime is done. Such a quantification finally is needed in order to decide whether to accept the remaining risks or to request further testing to improving the functionality before allowing a usage in real traffic.

Literature

[1] Watzenig, Daniel, and Martin Horn, eds. *Automated Driving: Safer and More Efficient Future Driving*. Springer, 2016.

[2] Winner, Hermann, and Walther Wachenfeld. "Absicherung automatischen Fahrens, 6." *FAS-Tagung München, Munich* 2013.

[3] Wachenfeld, Walther, and Hermann Winner. "Die freigabe des autonomen fahrens." *Autonomes Fahren*. Springer Berlin Heidelberg, 2015.

[4] www.enable-s3.eu – European Initiative to Enable Validation for Highly Automated Safe and Secure Systems.

[5] www.bast.de – Bundesanstalt für Straßenwesen (BASt).

[6] Dr.-Ing. Armin Engstle, Dipl.-Ing. Helmut Assmayr, Dr. Georg Schwab, Softwareentwicklung für autonome Fahrfunktionen,

[7] www.sae.org – Society of Automotive Engineers

[8] 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, Bradford, UK*. 2014.

[9] 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,

[10] 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.

[11] 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.

[12] Conference of European Directors of Roads – CEDR, *"Safe distance between vehicles"*, 2010.

[13] Holzinger, Juergen; Schloemicher, Thomas; Bogner, Erik, *Objective assessment of comfort and safety of advanced driver assistance systems*, FISITA 2016 World Automotive Congress, Busan, Korea