5

Tuning of ADAS Functions Using Design Space Exploration

Abhishek Ravi[1], Hans Michael Koegeler[1] and Andrea Saroldi[2]

[1]AVL List Gmbh, Austria
[2]C.R.F. S.C.p.A , Italy

5.1 Introduction

An ADAS function developed within the DESERVE platform and the tuning of this function for a particular application is discussed in this chapter. Based on separating the software and tuning data, according to the standards described in detail in Chapter 2, such a function can also be used for an alternate vehicle or application use case. The opportunities as well as the potential challenges are described, using a real world example, developed within the DESERVE Project.

5.1.1 Parameter Tuning: An Overview

Tuning or calibration of vehicle components is essentially determining the optimum attributes, which fulfill the legislative standards as well as refine the car's character to meet all the expectations of the driver for drivability and comfort. Besides the comfort and legislative issues the vehicle tuning also helps in brand differentiation and helps to determine the vehicle character.

In the tuning task for a specific component (e.g.: engine), the software and the tuning data in the application layer of an Electronic Control Unit (ECU) is separated which is illustrated in Figure 5.1. The resulting code is a hex file, which can be flashed to the defined controller hardware which gives a big flexibility in powertrain development. As an example, one engine hardware can be put into more than 200 vehicle variants fitting for different countries, different vehicles and/or different transmission systems – just by flashing a different appropriate controller software.

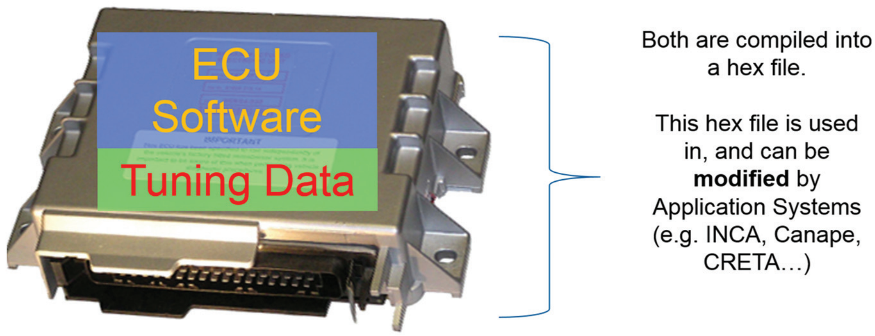

Figure 5.1 Separation of software and tuning parameters in a control unit.

5.1.2 Industrial Tuning Applications: Challenges and Opportunities

The engine – ECU has been the first mechatronic application in the automotive world. It makes sense to have a short view on the historical development of the tuning task in this field as illustrated in Figure 5.2.

In the past decades, the improving technology in the automotive sector can be seen with cars having better engine performance, less consumption, better handling and reduced emissions. But the improvement in technology has come with increased complexity, especially in the tuning task.

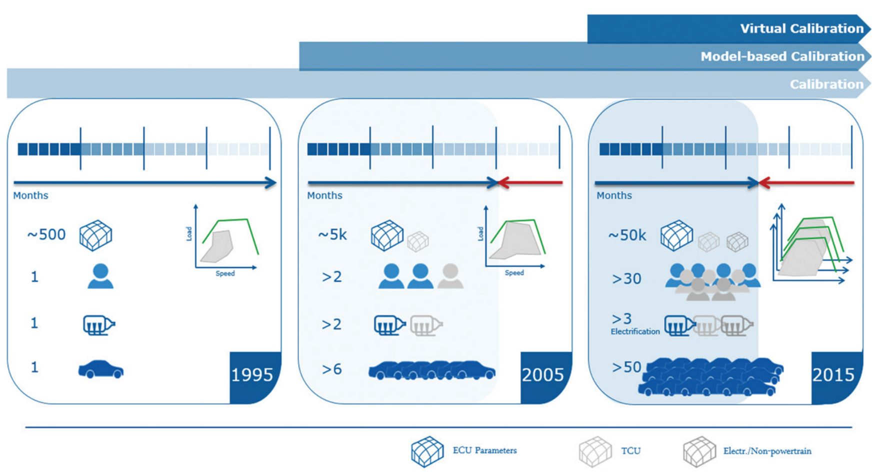

Figure 5.2 History of powertrain tuning (calibration).

As can be seen in Figure 5.2, initially there used to be around 500 parameters which needed to be tuned, which was carried out by a single engineer using the unit to be tested, which was then tested on a single test vehicle. Initially, the powertrain was quite simple and the Engine – ECU was the only one being considered.

With increasing legislative and user demands; the complexity of the technology, the number of involved interacting components (engine, gearbox and electric engine) and also the number of functions controlling the interactions between all the variable components increased dramatically. Further the tuning allowed the derivation of many more vehicle variants with the same hardware components but differing in the ECU-SW, wherein the functions in the SW stay the same, just the tuning data are specifically developed.

This effect is also seen in the number of tuning parameters to be defined in an engine calibration project, where around 50 k parameters have to be defined – clearly assigned to many functions. So it is no longer possible to have *one person*, who understands all the functions implemented and teams of specialized persons are necessary, partly working in different areas of the world. Thus the industry was confronted with several challenges and found some responses.

For example, the management of tuning data becomes an issue. It must be possible to track all the changes made to the tuning data by the different engineers involved and bring all the tuning results into a single final tuning result. The company should be able to ensure at Start of Production (SoP) that:

1. All the tuning data are calibrated.
2. All the tuning data are calibrated with the correct settings to optimally fulfill the desired, derivative use case.

These two requirements are very challenging, which explains the need of "Tuning Data Management". This topic itself is not further elaborated in this chapter, but is supported by valuable literature [1, 2].

Another challenge lies in the tuning for single use cases: For example, the emission tuning of an engine in a certain vehicle configuration for the legislation of a specific country. There are about 5 to 10 strongly interacting tuning parameters. E.g. an engine map to define the start of the combustion as function of speed and load is counted as one of these parameters, and exhaust gas recirculation rate, rail pressure, boost pressure, split patterns of the injected fuel quantity are others, all either reducing the different kinds of emissions or changing fuel consumption or noise.

So one can imagine, that it is just not possible to measure the emissions and the fuel consumption of all the feasible combinations of say 8 of such parameters on an engine. (A similar issue faced with ADAS functionality)

Such tasks are typically performed on engine test beds and chassis dynos and have to be finally validated on the road again. With the latest legislation (Real Driving Emissions, RDE) even the certification will be done on the road giving additional challenge [3–5].

Figure 5.3 illustrates the generalized development environment, which allows the engineer to reproduce maneuvers and then double check the results of tuning work. In the manual tuning method, the engineer operates the UUT with a certain setting of control parameters in certain maneuvers. The engineer observes the behavior of the UUT and performs a judgment according to his experience. Then the next setting is defined with the intention to better approach the desired behavior. This process becomes complex when there are many relevant tuning parameters [6].

In this trial and error method, the quality of tuning and the optimization results depend on whether the engineer considers all the parameters that are relevant for the desired behavior and the relevant start point. There is a strong dependence on the experience of the engineer. There are also limitation on the number of tests that can be conducted, due to the testing time, complexity and cost factors. The final results are highly subjective, as the decision making

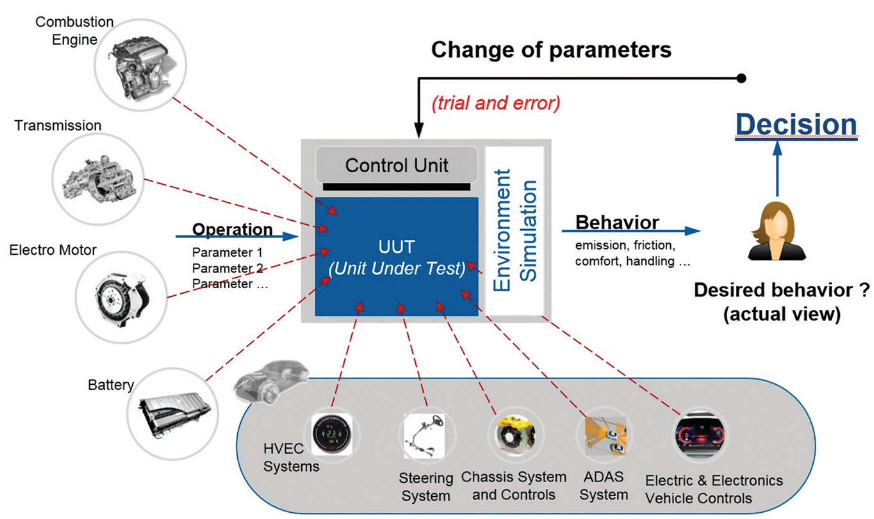

Figure 5.3 Illustration of a generalized development environment and manual tuning process.

process lacks traceability and a reuse is not possible for future projects, e.g. tuning an ADAS setup for a different drive mode. As a result, a methodology to increase the efficiency and the quality of the tuning work at the same time, the so called "Design of Experiment" method (DoE) was adapted accordingly.

Within the DESERVE context this methodology was applied as "Design Space Exploration" for Simulation environments, which are excellent development environments for tuning of ADAS Functions.

The model-based approach was used with two objectives:

- Firstly, to find an optimum tuning result.
- Secondly, to validate an existing tuning result under a big variety of use cases, which will happen during the lifetime of a vehicle.

5.1.3 Model-based Tuning

Model-based tuning is a statistical, model-based approach which reduces the amount of actual experiments/test runs needed to accurately describe the behavior of the UUT within the design space. This method helps to choose the position of the test data points in order to generate behavior models with an efficient low number of measurements. Such models are then utilized to develop an accurate and robust tuning according to specific optimization target(s). In Figure 5.4 the entire method is illustrated again for the generalized development environment.

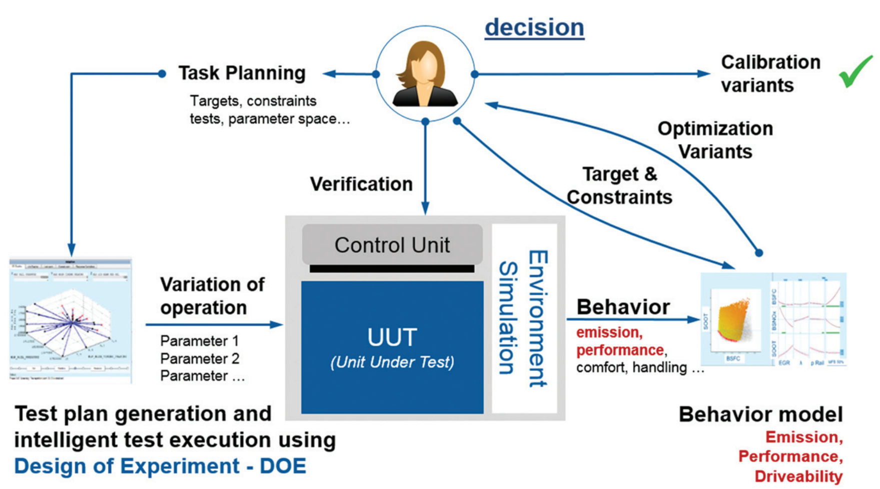

Figure 5.4 Model-based tuning task illustrated.

In a model-based tuning task the below steps are followed:

- The user begins with a task planning for the measurement series, where the targets for the tuning task are determined. Based on the targets, the relevant input parameters which are considered to influence the observed UUT response are selected. AVL CAMEO is used for the test plan generation. This is based on a one time set up process, in which CAMEO is connected to the development environment. Thus CAMEO gets access to set tuning parameters in the UUT, observe responses of the UUT and to start/stop maneuvers and to take measurements after maneuver. The development environment hosting the UUT could be in the form of a test bed, a hardware-in-the-loop (HiL) or even a vehicle simulation software like IPG Carmaker in combination with an ADAS-function prototype programmed in MATLAB.
- Once the targets have been defined the next important step is to make the test matrix. In order to get a full picture of the area to be investigated, the Design of Experiments (DoE) is used [7]. It is a systematic technique which allows varying all the parameters simultaneously while answering the two important questions of every tuning activity: Firstly, how many tests are needed to cover the entire design space? And secondly, at which locations in the design space test points are needed to effectively get modelling equations valid throughout the entire design space. There are many DoE designs available to us in AVL CAMEO, but COR DoE methodology [8] was used in the current example exercise. Besides setting up the test design, it is also important to set the limits for the test and appropriate actions when the limit is violated. These topics are addressed further on in the example discussed in Subsection 6.2.1.
- With the test plan and limits decided the tests are run, where the necessary parameter settings are uploaded to the UUT by CAMEO, and after the test, the required measurement results were stored in CAMEO. The raw measured data check is then carried out in order to check the plausibility and feasibility of measurement. It is a necessary check to get a rough idea of how the measurements compare against expected values, and also observe possible errors which could have occurred during the test execution.
- The measurements are modeled empirically to obtain behavior models of the UUT. In this content, modeling means more or less to fit a function – like a polynomial equation for example – into the measured responses in order to estimate the response function of any point in the design space.

Such a model helps understand the reaction of the UUT to the parameter tuning, and the interaction of the different tuning input parameters and the output measurements. The confidence and prediction intervals of the empirical models are observed to evaluate the model quality. Models in CAMEO also allow extrapolation in defined ranges beyond the design space covered by measurements to observe the UUT behavior at points where tests could not be run based on equipment limitations or time/cost constraints.

- Based on the optimization target, optimization algorithms can be implemented for a single objective or multiple objectives. The engineer can decide if the results meet the targets and constraints and in case of multiple objectives decide on a suitable tradeoff between the different desired targets (Pareto front).
- Before, the results from the analysis are accepted a final verification test is carried out. Tests are run at least on the point of the decided optimum, but can also be extended on parameters settings of ten or more points spread across the Pareto front. If these verification measurements match the modeled results then the empirical models are accepted and the engineer can use the optimization results as the desired tuning setting.

5.1.4 Model-based Validation

A model-based validation is a task carried out to test and evaluate the robustness of the results from the tuning task. The UUT is run at the parameters settings obtained from the tuning task, but tested for an alternate use case and the response is evaluated. For example; if say a diesel engine was tuned to operate at an economy mode and a sport mode with strong limits set on NOx emissions. Economy mode encourages the engine to conserve fuel while sacrificing power, while the Sport mode encourages the engine to provide greater power while making compromises on fuel economy, with the engine running more at the higher RPMs. The engine is initially tuned at driving conditions imitating an urban environment and lower altitudes, and from the tuning tasks the input parameters settings like the rail pressure, injection pressure, injection timing etc. are selected to operate the engine at the two targeted modes while sticking to the NOx limits. In the validation test run the engine is first run at the economic mode and then sport mode, but now the use case is in hilly road conditions and higher altitude. The engine performance is evaluated with respect to power and emissions, while the road and altitude of operation is varied. The target is to see if tuning settings could be extrapolated or extended to alternate use cases. It also gives further information on how the engine tuned for urban conditions would perform on rugged hilly conditions.

5.2 Demonstrative Example

A map-based ACC-Function (developed by the DESERVE Partner CRF) running in a commercially available MiL Environment (IPG-Carmaker + MATLAB Simulink) has been used as an example. The calibration tool of AVL CAMEO was connected to this environment in order to tune the function for a Fiat 500L.

5.2.1 Function: An Overview

A map-adaptive autonomous cruise control (ACC) was developed to:

- Control the vehicle velocity in order to enter and exit curves in a comfortable and safe manner.
- Complete the drive maneuver in the least amount of time.

The controller function controls the vehicle speed by sending jerk request (see Figure 5.7). Jerk is the rate of change of acceleration. Hence the jerk request signals from the controller function are converted into the vehicle acceleration and speed. For the reference maneuver a digitized road was used and a reference speed curve was determined, which is the maximum speed at which this road can be safely maneuvered. The function tries to ensure that, the vehicle follows this reference speed profile as closely as possible without exceeding it. The target speed was set at 130 km/h for the ACC. A demonstrative speed profile is shown in Figure 5.5 for a sample settings in

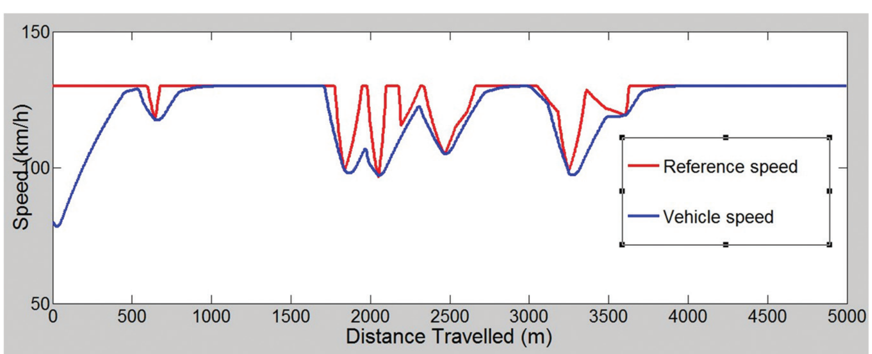

Figure 5.5 Velocity profiles for a sample test run using the control function.

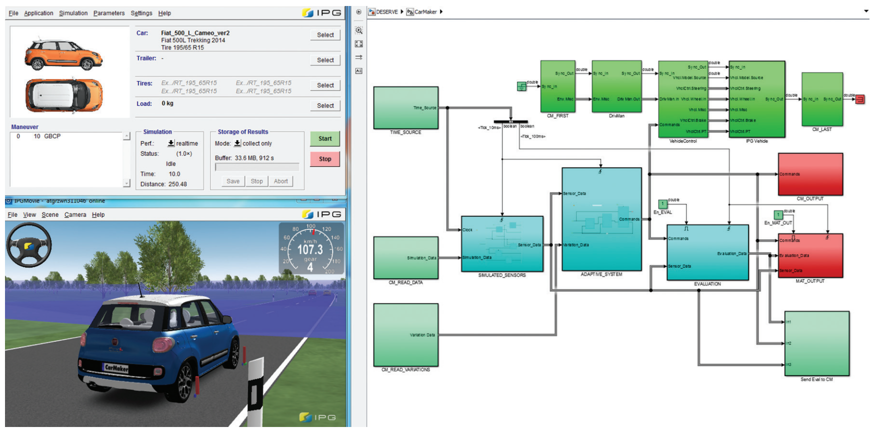

Figure 5.6 Function developed using IPG carmaker and MATLAB simulink.

the ACC function. It can be seen that the vehicle velocity tries to follow the reference velocity while never exceeding it. The vehicle velocity is not able to exactly replicate the reference velocity due the road conditions, the vehicle limitations and the control function settings.

The function was developed using IPG Carmaker for Simulink and has been illustrated in Figure 5.6. IPG Carmaker for Simulink is integrated into MATLAB/Simulink and necessary modification were done by adding the custom Simulink blocks developed for the current use case.

5.2.2 Design Variables

In order to tune the function for the reference maneuver, four input parameters or design variables were selected (see Figure 5.7). As per the terminology used in CAMEO these tunable input parameters will be referred to as the **variation parameters**. The variation parameters selected for the tuning task are:

- Acceleration Maximum (**A_MAX**) limits the maximum positive acceleration the vehicle can have while safely completing the maneuver. The negative acceleration is not limited in order for the vehicle to generate the necessary breaking force in case of obstacles.
- Jerk Maximum (**J_MAX**) limits the maximum positive jerk request from the controller function in order to meet the reference velocity curve. But only the positive jerk given by the engine and responsible for positive acceleration is limited, while there is no lower limit for the negative jerks for reasons mentioned previously.

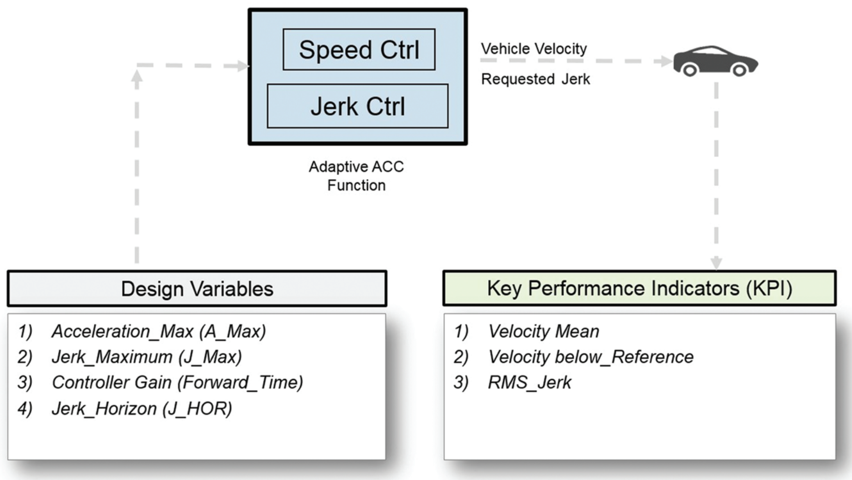

Figure 5.7 Function overview.

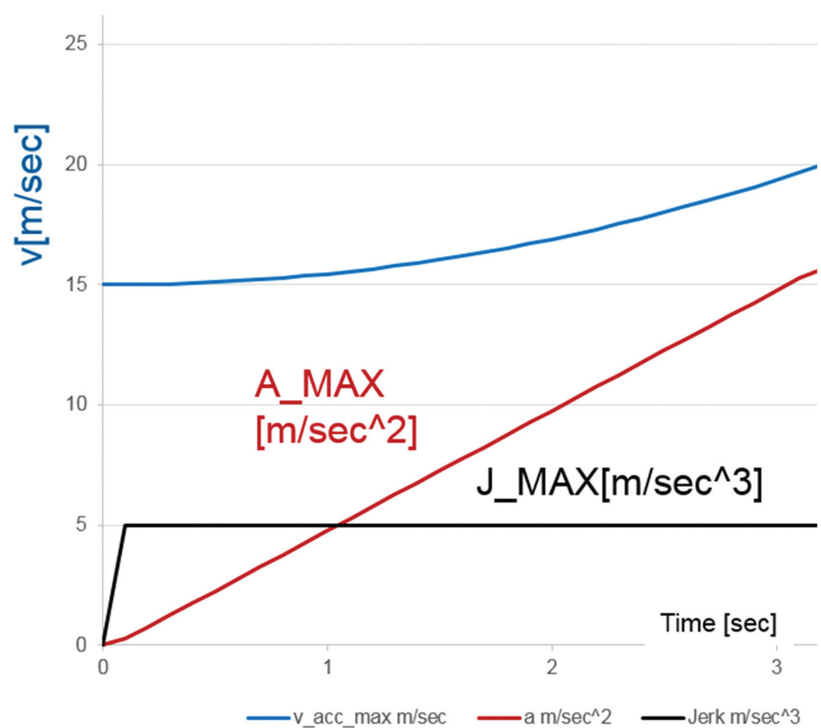

Figure 5.8 Illustration of the kinematic variables A_MAX and J_MAX.

Figure 5.8 illustrates the kinematic parameters, with acceleration being the derivative of velocity and jerk the derivative of acceleration.

- Forward Time (**FORWARD_TIME**) is a gain factor to transform the jerk request from the controller function to an acceleration request. Even though the controller function is based on jerk and sends the desired jerk requests for the vehicle, the interface to control vehicle motion is based on acceleration. Hence to control the vehicle the desired value of acceleration is required. In order to obtain the desired acceleration from the request jerk, one has to look forward for a given time which is called Forward Time. Mathematically it can be defined by the formula.

  A_req = A_0 + J_req***FORWARD_TIME**
  A_req is the Acceleration request
  A_0 is the current vehicle acceleration
  J_req is the Jerk request generated by the controller function

- Jerk Horizon (**J_HOR)** is a parameter used to determine when the controller function sends the necessary jerk requests and the required jerk magnitude in response to an approaching curve. To define what is "near" and "far" (with respect to the distance from the approaching curve) for the controller function, the parameter **J_HOR** is used, where HOR stands for the horizon points (of the electronic horizon) to be considered. J_HOR is always a negative value, and values closer to zero make the controller respond to the approaching curve when it is further away with a smaller deceleration demand. Higher negative value tells the controller to respond when the approaching curve is closer in proximity but with a larger deceleration. A pictorial representation is given in Figure 5.9.

  The black line represents the target velocity set for the controller and the reference velocity curve is given in red. As explained previously the controller tries to control the vehicle speed (in blue) as close as possible to the reference speed.

  The mathematical expression "**A_MAX + J_HOR***time" determines the funnel of the vehicle velocity curve shape (shown in blue). More negative **J_HOR** give the velocity curve a sharper shape, while values closer to zero give the velocity curve a flatter shape.

The range of the variation parameters examined in the tuning task have been shown in Table 5.1.

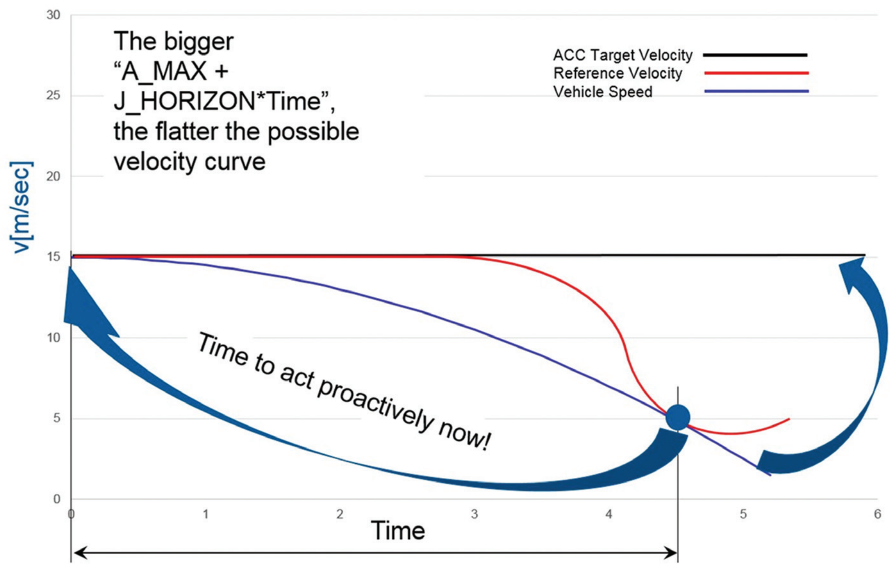

Figure 5.9 Illustration of the design variable (variation) J_HOR.

Table 5.1 Range of variation parameters used in the tuning task

Design Variable	From	To
A_MAX (m/s^2)	1	5
FORWARD_TIME (s)	0.1	2
J_HOR (m/s^3)	–5	–0.2
J_MAX (m/s^3)	1	3

5.2.3 Key Performance Indicators (KPI)

The output variables to demonstrate the effectiveness of our tuning task to meet the targets are described below and illustrated in Figure 5.10:

- Mean Speed: The mean of the vehicle speed in each test run is indicative of the sportiness of the driving experience. A higher mean speed helps finish the test maneuver in less amount of time, and makes the driving experience sportier.
- Speed below reference: The reference speed curve is the maximum speed with which the vehicle (Fiat 500L) can maneuver the digital test track without leaving the road for the reference use case. Hence to ensure vehicle safety it was ensured that the vehicle speed during the tuning task was always below the reference velocity.

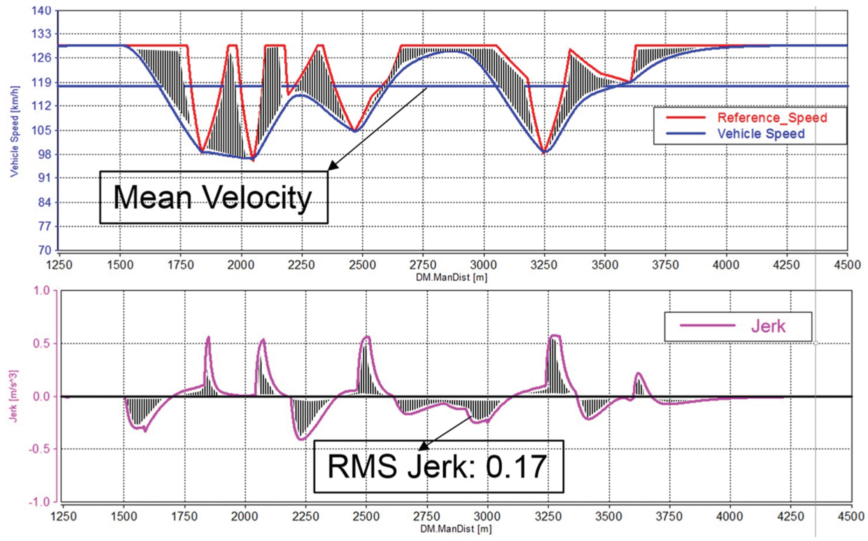

Figure 5.10 Key performance indicators.

- Jerk_RMS: Vehicle jerk which is the rate of change of vehicle acceleration, is indicative of the driving comfort. Lower rate of change of jerk gives a comfortable ride, so the root mean square of the jerk in a test run is a good indication of the driving comfort.

5.2.4 Test Maneuver

The test maneuver consisted of 5000 m test run on a digitized road imitating the road between Ceva and Savona in Italy run on IPG Carmaker for Simulink (CM4SL). IPG Carmaker environment is illustrated in Figure 5.11. The top left is the Carmaker for Simulink main GUI, showing details about the vehicle, simulation speed, time and distance of maneuver etc. The bottom left imitates the car instrumentation. The top right is time based plot of car speed and the vehicle jerk. The bottom right is the IPG Movie which illustrates the overall test run in a movie.

5.2.5 Test Run Overview

The test run overview is illustrated in Figure 5.12. The test parametrization was done in AVL CAMEO, where a space filling DoE design with the four variations was used. The variations were then uploaded to CM4SL through

Figure 5.11 IPG Carmaker test environment.

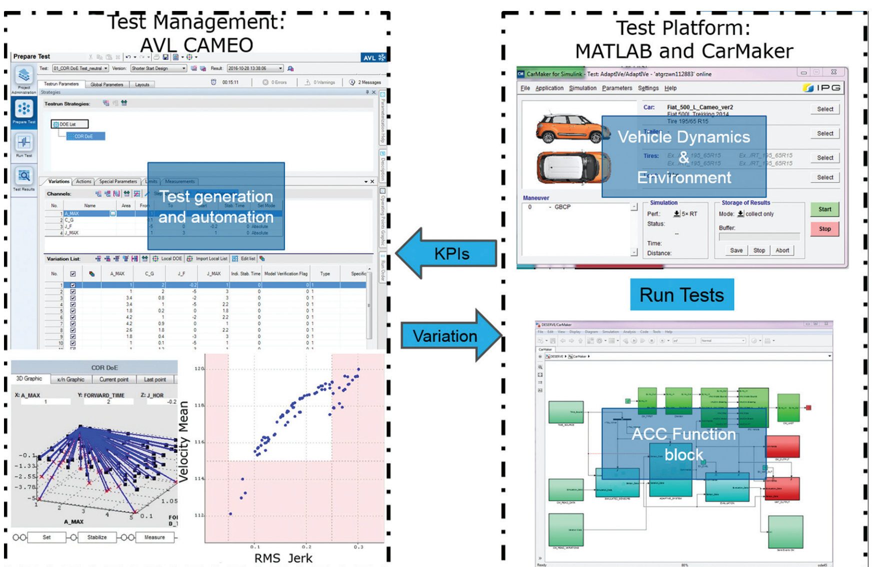

Figure 5.12 Test run overview illustrating the work flow.

the CAMEO-Carmaker Interface, where the test maneuver was run for each variations setting. AVL CAMEO then stores the measurement parameters observed as the KPIs for further evaluation.

During parametrization there were limits set on the minimum (–2 m/s^3) and maximum (2 m/s^3) acceptable vehicle jerk values. Whenever the vehicle jerk value violated the limits the test run at that test point was halted and no measurements were recorded. This affected the overall DoE design effectiveness with a reduced design space and as a result reduced measurement points. To overcome this challenge a COR DoE (Customized Output Range) method was utilized, which is an iterative method where first alternate test points were added by CAMEO to maintain the DoE design. Then based on these preliminary measurements the design space was further modified and additional test points were added in the relevant variation space to improve the final information from the measurements. Design space modification. The AVL CAMEO interface is illustrated in Figure 5.13, where the image to the left illustrates the overall test parametrization while the image to the right shows the test run window.

5.2.6 Raw Data Plausibility Check

Before the mathematical modeling of the selected output measured variables, the raw measurements were checked for plausibility. Firstly, the measured variables were checked for any outliers as shown in Figure 5.14 for mean

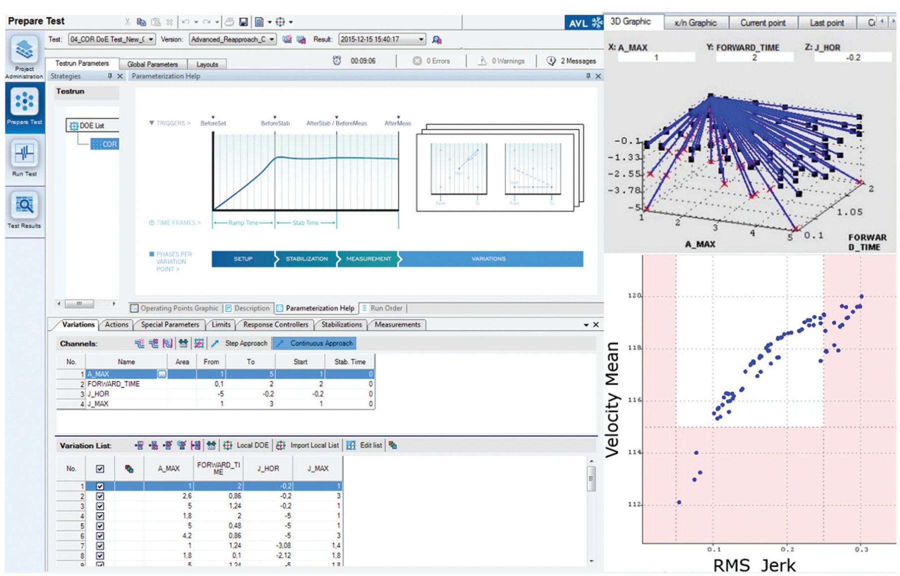

Figure 5.13 Left image illustrates the test preparation window while the right image illustrates the test run window.

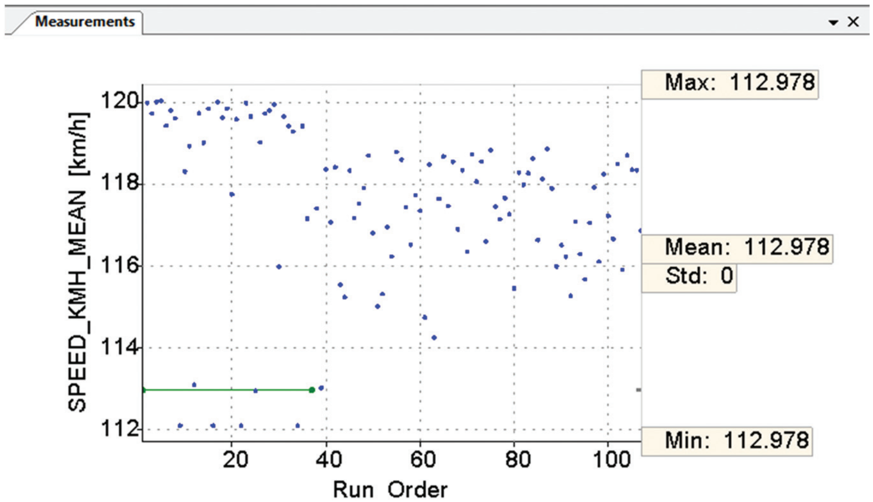

Figure 5.14 Checking for outliers in the measured variables.

speed. The measured values were within the acceptable range. The figure also shows that the repetition points (a select number of test conditions, usually the start condition which are repeated to check the reproducibility of test results) shown in green were perfectly reproduced.

The effect of design space modification, due to limit violations and the design correction by COR DoE method can be seen in Figure 5.15. In a certain range of variations for A_MAX, J_HOR and FORWARD_TIME there are no test points. Limit violations encountered when tests were carried out at these range of points are the reason why they were skipped by AVL CAMEO. Conversely a greater density of test points in certain ranges of variations show where the COR DoE added alternate or additional test points.

5.2.7 Meta Modelling

The raw data plausibility check was followed by empirical modeling of the output variables. The automatic modeling in CAMEO gave reasonable results with a neural networks model with local model order 2, as can be seen in Figure 5.16 which is the Measured (Predicted) plot which shows the fit of the model to the measurement points. If there is a perfect match all points will lie along the black line, but in our case the measurement points are reasonably close to the black line.

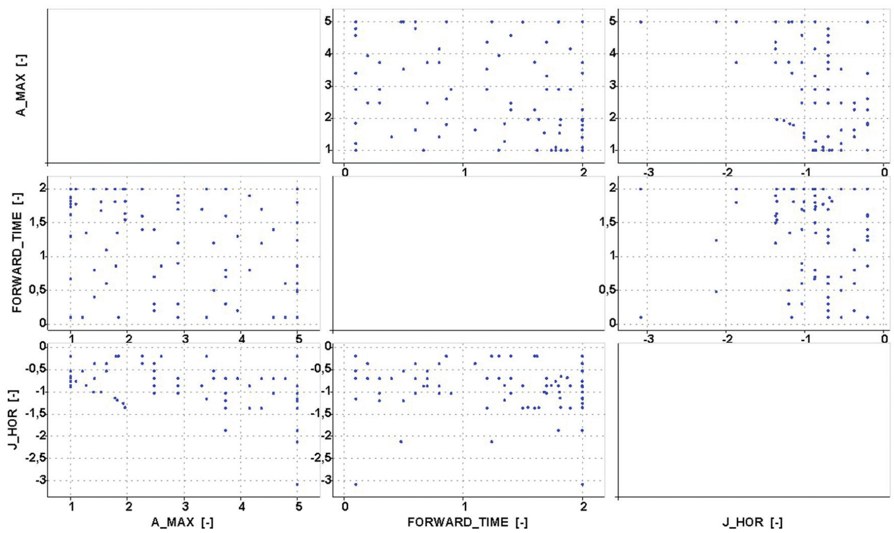

Figure 5.15 Check of DoE design and the boundaries of variation parameters.

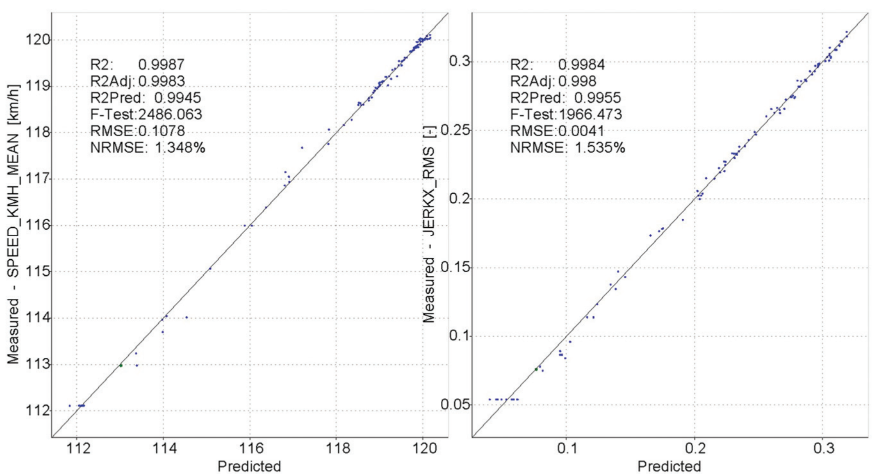

Figure 5.16 Figure depicting the quality of empirical modeling.

After checking the quality of modeling, the intersection plots were used which represent a cut through the multidimensional model, showing the influence of each variation depending on the values of the other variations. In Figure 5.17 the influence of the variation parameters on Speed_Mean and

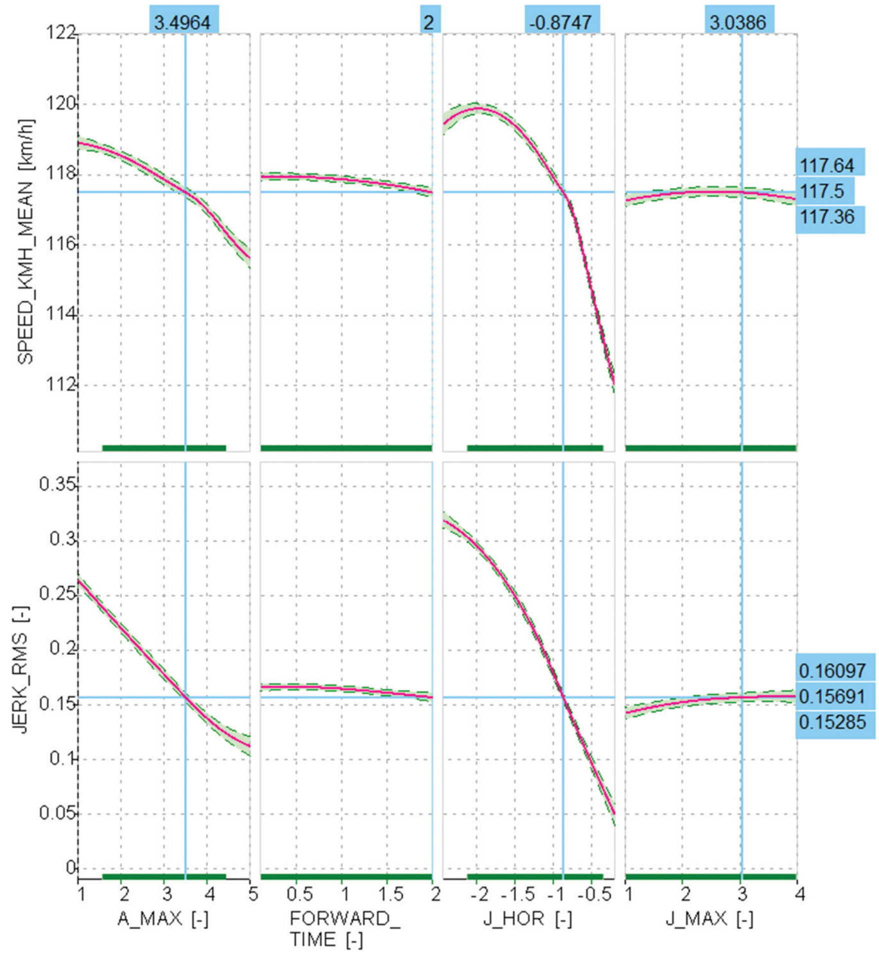

Figure 5.17 Intersection plot highlighting the influence of each variation on the output variables and their interaction.

Jerk_RMS can be observed. The confidence interval of the model is displayed in the green dotted line and colored section. The narrow confidence interval shows a high quality fit. The green bar on the x axis for each variation shows the total design space, and as the confidence interval of the model in the extrapolated region is also narrow, it shows good extrapolation capability of the model. Now looking at the intersection plots, it can be noticed that J_HOR and A_MAX have a strong influence on the output parameters. The more negative the J_HOR, the later the vehicle reacts to an approaching curve. Hence it is still travelling at a high speed before decelerating to approach the curve safely. Hence a higher mean speed is observed, but the resulting braking produces higher vehicle jerk reducing the driving comfort. Influence of A_MAX can be a bit counter intuitive but it can be seen that A_MAX is used to calculate J_HOR. The higher A_MAX, the less negative is J_HOR. Hence for higher A_MAX values J_HOR is closer to zero hence a smoother and slower ride. It can also be observed that higher FORWARD_TIME allows for a smoother and slower ride, which is because the controller can take more time to achieve the desired acceleration.

5.2.8 Optimization

From the intersection plot, it is possible to manually find values of the variations which give a comfortable ride or sporty ride or an acceptable compromise. But it is quite easy to miss the optimum or an acceptable compromise when working with multiple input variations, hence the optimization tool in CAMEO was used. In the current tuning scenario, the target was to be able to isolate two modes of operation, comfort mode and sporty mode. Hence a multi objective optimization was chosen with limits set on the minimum desired mean speed of 115 Km/h and maximum acceptable JERK_RMS of 0.28 (Figure 5.18).

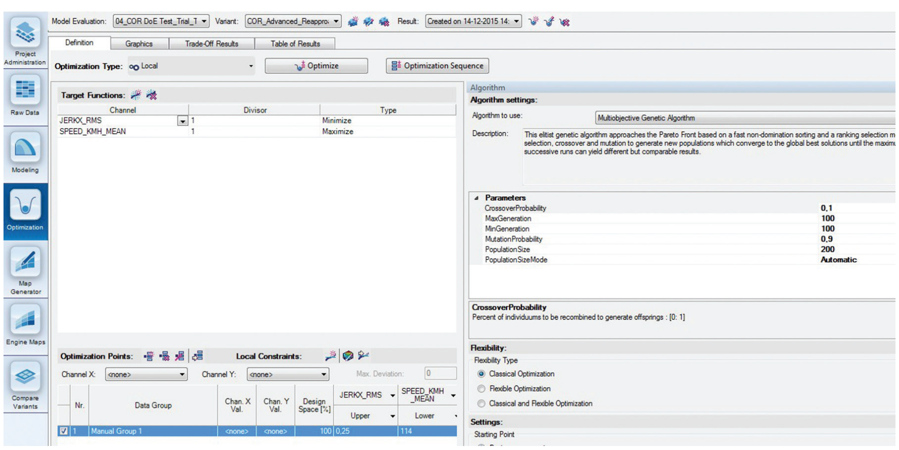

Figure 5.18 Optimization setting window in AVL CAMEO.

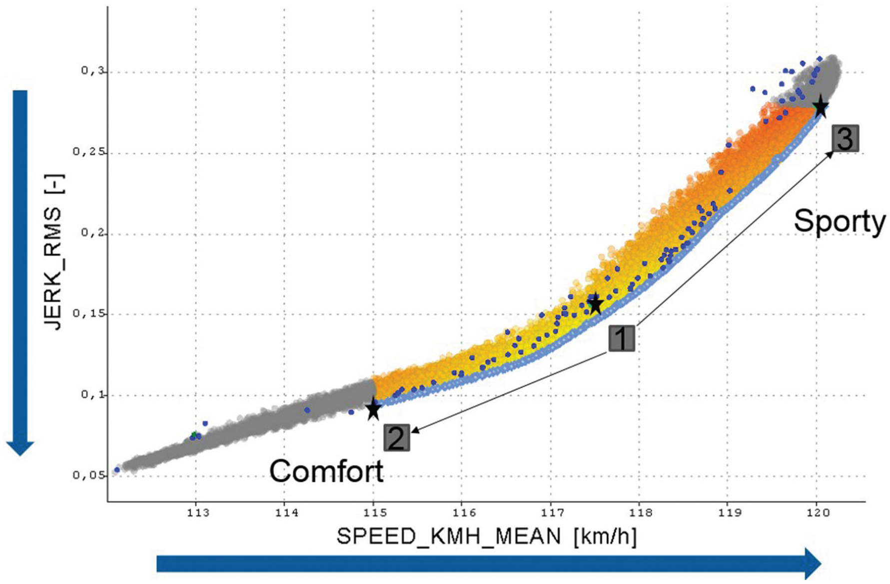

Figure 5.19 Trade-off plot between comfort and speed.

The result is plotted in a trade-off plot as shown in Figure 5.19, where the steel blue is the pareto front, the blue points indicates the measurement values and the other yellows points are random space filling points. The pareto front shows the possible optimum trade-off solutions which can be considered equally good as the only way to improve on objective would be to compromise on the second objective. So by observing the pareto front it is possible to define an optimum for comfort mode and an optimum for sporty mode of operation Table 5.2.

In Figure 5.20: Sporty mode vs comfort mode: the vehicle performance when operating at the two modes can be observed. The red velocity curve is the reference velocity and blue velocity curve is the actual vehicle velocity. It can be observed that the actual velocity is always below reference velocity which was the safety requirement. Also the velocity changes in comfort mode

Table 5.2 Variations values for comfort and sporty mode

	A_MAX	FORWARD_TIME	J_HOR	J_MAX	SPEED_Mean	JERK_RMS
Comfort	4.99	1.94	–0.84	1.0	115	0.09
Sporty	3.88	1.37	–1.84	3.36	120	0.28

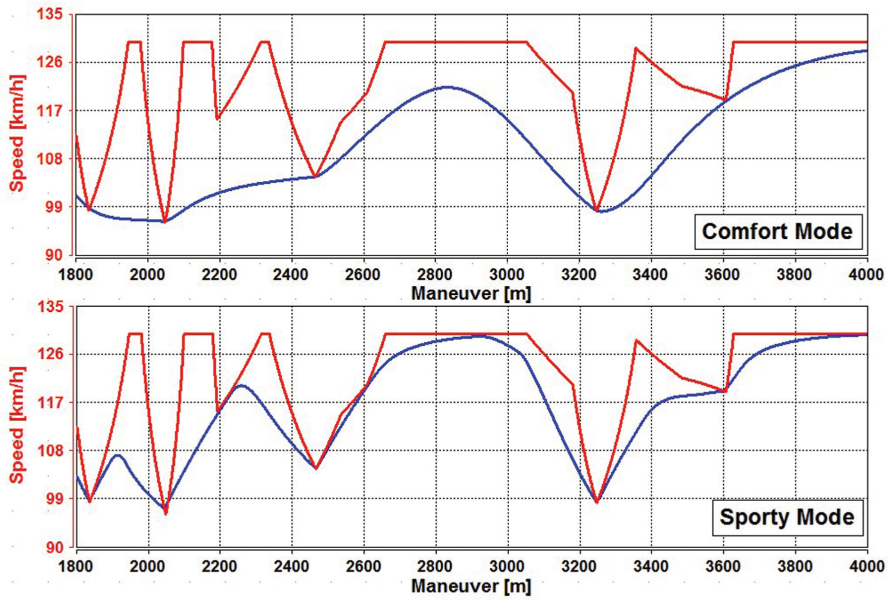

Figure 5.20 Sporty mode vs comfort mode.

is more gradual with no sharp peaks unlike in sporty mode where there are rapid fluctuations in vehicle velocity. This behavior is also mirrored in the acceleration values in both operation modes. The vehicle jerk curves (red plot is the jerk request generated by the controller and blue the actual vehicle jerk response) show much lower values in vehicle jerk for comfort mode while the sporty mode show sharp and frequent peaks in jerk value.

5.2.9 Verification

The pareto front consists of points a majority of which are from the model extrapolation. In order to verify the robustness of the model to accurately extrapolate, ten random points were selected from the pareto front and for the corresponding variation values the test runs were rerun. The results from these test runs were evaluated as verification points in CAMEO. The Figure 5.21 shows the extrapolated model (in red) and its prediction interval (in blue), and the measured verification points and its modeling (in green). The measured verification points lie within the prediction interval of the model, showing the extrapolation accuracy of the model.

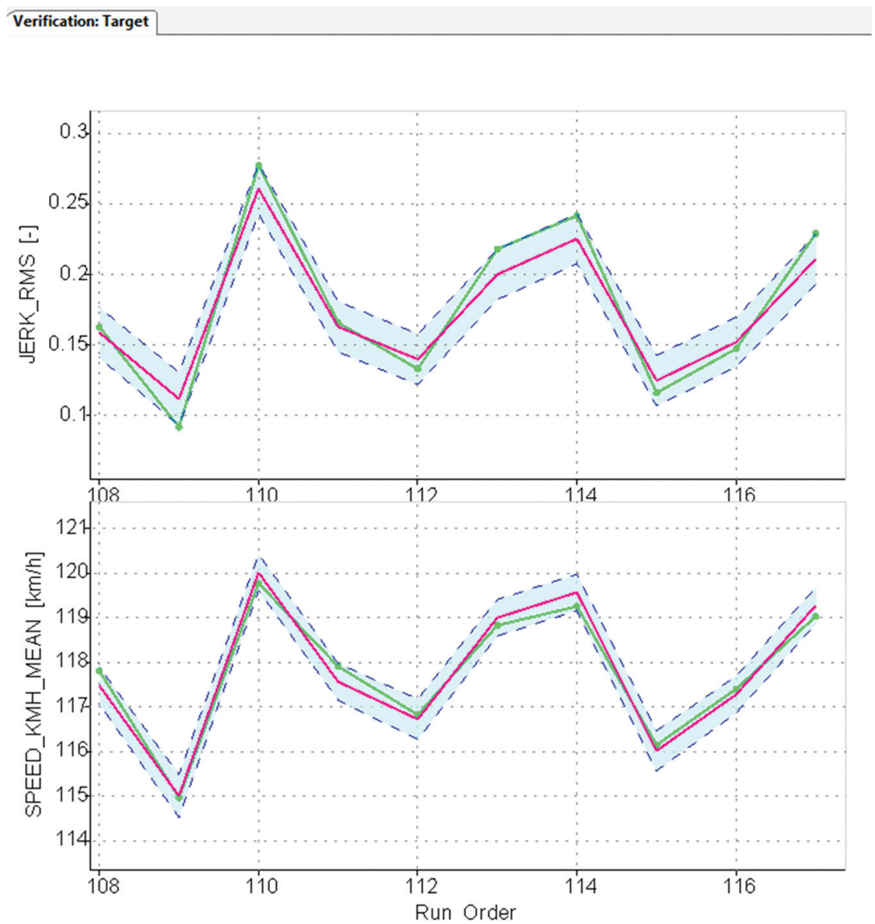

Figure 5.21 Verification plot to see how well the measured results from the verification run fit the model results.

5.3 Model-based Validation

Once the reference tuning task is completed, it has to be tested, if the tuning results are still acceptable, when not running the reference use case but for varying road characteristics. Will the comfort mode still allow for a comfortable drive also for different road situations? It would be unfeasible to run simulations on thousands of different roads, besides making it difficult to realize the influence of a specific road. In the current method the two tuning modes are fixed and a system variation of a digitized road is performed using the model based approach to validate our tuning results.

The digitized road is shown in Figure 5.22, where the lengths of the straight sections (L1, L2, L3, and L4) and curvatures (R1, R2, R3) were varied while keeping the total maneuver length to 5000 m. The controller settings were fixed to run at first comfort mode and then sporty mode, and the resulting measurement output variables are shown in Figure 5.23.

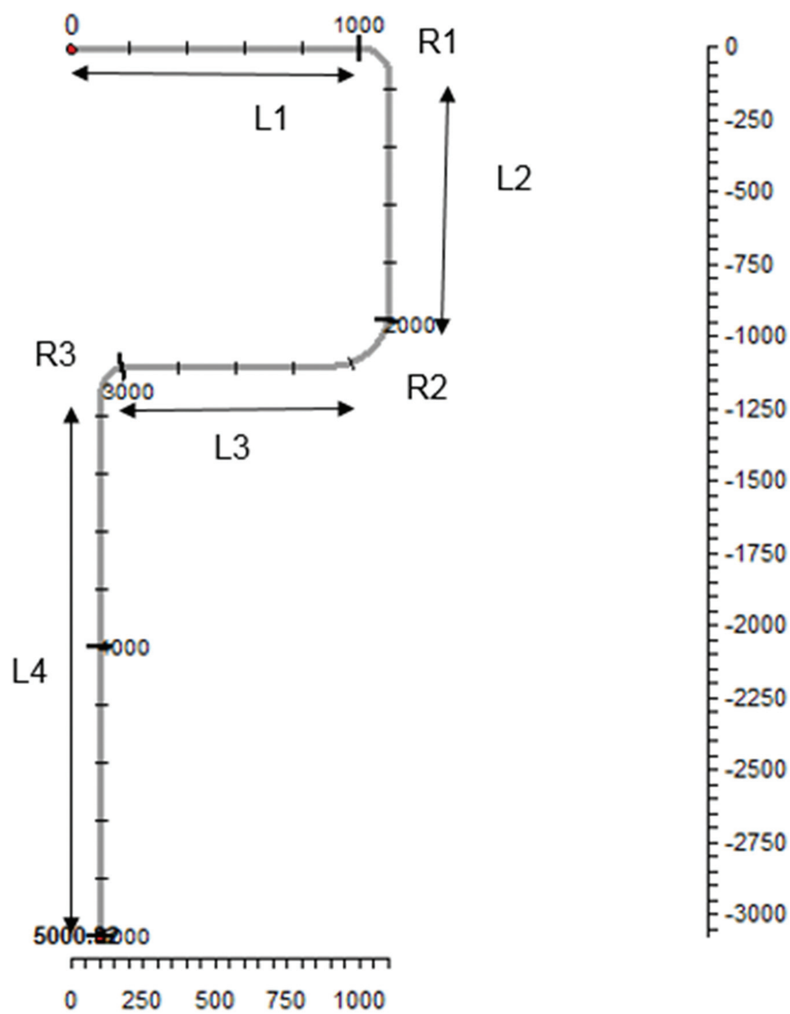

Figure 5.22 Digitized road used for the validation run.

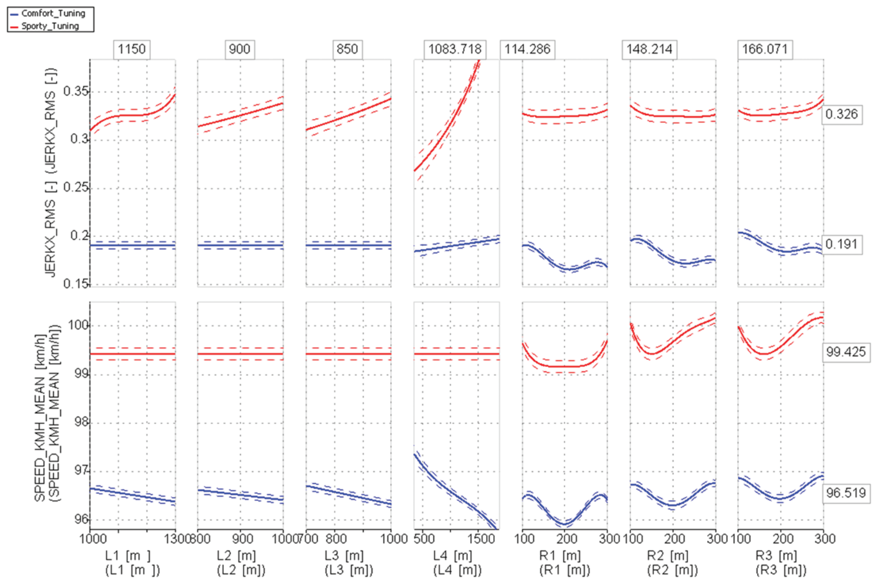

Figure 5.23 Measurements comparison when run on comfort mode (in blue) and sporty mode (in red).

It can be seen in Figure 5.23: Measurements comparison when run on comfort mode (in blue) and sporty Mode (in red) that for the sporty mode the resulting drive comfort is lower as indicated by the higher JERK_RMS. The length of the straight portions do not influence the JERK_RMS for comfort mode as strongly as in the sporty mode. The curvature of the turns seem to influence the output in both the operation modes. A JERK_RMS limit of at least 0.35 is expected, and it can be seen that the limit is maintained in both the modes of operation for majority of the design space. In the sporty mode the controller is set to maintain a higher vehicle speed and responds to the oncoming curve only when it is close, hence the longer the straight sections, the larger the jerk experienced when it decelerates rapidly to approach the curve followed by a strong acceleration on leaving the curve. For the comfort mode, the controller is set to focus on keeping the vehicle jerk close to minimum. The validation task showed that, if the function (our UUT) is kept constant and the simulation environment is changed, the function still manages to meet the expected vehicle jerk targets. The influence of 'L4' on the jerk behavior needs to be further investigated as it strongly increases the vehicle jerk fluctuations at higher values especially for the sporty mode. To further explore and investigate the influence of test track characteristics on the function response, it can be tested on a variety of road types and test tracks. This assists in the further improving the function performance.

5.4 Conclusions

Virtual tuning of an ADAS function developed on a MiL environment using an optimization tool can be a powerful combination for the development of a brands driver assistance system. The classical approach relies on a subjective tuning of the ADAS function on a proving ground and public roads, which can be supported and accelerated by using a virtual tuning environment. Using DoE methods supported by AVL CAMEO, it was possible to increase the number of tuning tests compared to a manual tuning, and also the number of target parameters and tests needed to match them. The possibility to use the developed function for alternate use cases by separating the software and the tuning data is precondition for tuning works in general.

Independent of that also in the validation process a model-based approach can be very helpful, as the test coverage for a certain use case can be extended to a wide range of possibly occurring variants of that use case. The robustness of the key performance indicators considered as relevant can be estimated.

Acknowledgement

We would like to thank Mr. Andreas Saroldi from CRF for providing the ADAS function.

References

[1] M. Paulweber and K. Lebert: *Instrumentation and Test Systems: Power Train development, Hybridization and Electrification.* Chapter 5.4.2 *Application Data management,* Springer View, 2016.
[2] AVL Tuning Data Management Software: http://www.avl.com/creta (called at 2016 01 20).
[3] H.-M. Koegeler; A. Fürhapter; M. Mayer; K. Gschweitl: *DGI-Engine Calibration, Using New Methodology with CAMEO*. In SAE NA, Capri –Italy, 23–27. September 2001.
[4] E. Castagna; M. Biondo; J. Cottrell; H. Altenstrasser; Ch. Beidl; H.-M. Koegeler; N. Schuch: *Multiple Tier 3 Engine Applications based on global modelling.* In MTZ 6/2007.

[5] T. Fortuna; H.-M. Koegeler; M. Kordon; V. Gianluca: *DoE and Beyond-Evolution of the Model based Development Approach*, in ATZ worldwide, Springer, 2015.
[6] H. M. Koegeler; B. Schick; P. E. Pfeffer; A. Contini; M. Lugert; T. Schöning: *Model Based Steering ECU Calibration on a Steering in the Loop Test Bench*, in Chassis Tech 2015.
[7] D. C. Montgomery: *Design and Analysis of Experiments,* John Wiley and Sons.
[8] A. Rainer; H. M Koegeler; D. Rogers: *Iterative DoE – Improved emission models and better optimization results within a shortened measurement time,* in PMC, 2014.