

Automated in-vehicle engine calibration to optimize emissions levels using machine learning

Alexandre Esteves

Robert Bosch Ltda

Alexandre Massayuki Kawamoto, André Pelisser, Davi Carmelutti

ETAS

Alexandre.Esteves@br.bosch.com, Alexandre.Kawamoto@etas.com, Andre.Pelisser@etas.com,
Davi.Carmelutti@etas.com

ABSTRACT

The growing concern with environmental impact is a major drive for the tighter emissions imposed on combustion engines. The compliance with those restrictions is pushing new hardware and software solutions that, nevertheless, increase system complexity leading to more iterations and hence longer development time, a risk to profitability. In this context, it is paramount to reach optimum catalytic conversion temperature earlier through calibration strategies. This is challenging though, requiring heavy efforts in terms of time and use of resources/facilities.

A promising approach to increase efficiency is the concept of virtualization via data-based engine modeling and model based calibration. This paper presents the combined automation and virtualization in vehicle for catalyst heating calibration. Firstly, a small number of test conditions were automatically measured based on a “Design of Experiment”(DoE). The acquired data was then fed into ASCMO, a Gaussian process regression algorithm implementation by ETAS, resulting in a model used for calibration optimization with respect to emission and engine stability targets. The use of this novel method anchored on automation, machine learning and virtualization aims to improve process efficiency and the robustness of the calibration data.

INTRODUCTION

Engine calibration task is divided into different working packages due to the high complexity and workload current systems require, such as base calibration, start, modeling, exhaust temperature control, drivability, diagnosis and emissions. The main goal over a calibration development is to achieve a tradeoff between customers (OEM) requirements and at the same time legislation compliance. Among the legislations required to release a combustion engine project into the market, tail pipe emissions are one of the most laborious tasks. That being said, catalyst heating strategy is a major calibration resource used to increase engine emissions efficiency mainly at cold phases. However, to optimize it there are a high number of combinations among the possible engine parameters combinations and strategies, such as:

- Ignition angle Efficiency;
- Multiple ignitions;
- Engine speed;
- Camshaft Position Control;
- Injection rail pressure;
- Injection mode operation;
- Start/End Injection angles;
- Injection distribution portion (Suction and Compression strokes).

To find out the best combination of those engine inputs extensive testing and a large number of measurements are required. In addition to that, the manual effort to analyze this data without a processing tool, can make this task even harder

and bound to misunderstandings. One of the aims of this paper would be to propose a solution to increase engine calibration efficiency throughout a databased modeling approach for the catalyst temperature behavior and related outputs based on measurements taken in-vehicle, hence decreasing exhaust emissions.

The target outputs considered in the model are: catalyst temperature itself, engine speed “roughness”, fuel mass consumption and misfire counters. They serve as metrics to determine the impact of the inputs combination on engine tailpipe emissions and the subsequent analysis and optimizations.

As a proof of concept, this study has been performed with a modern, Flex-Fuel Turbo engine with 1.0 L displacement, inline three cylinders architecture and direct injection with inlet and outlet camshaft control. All measurements conducted for the elaboration of this paper were done so automatically with a dedicated solution able to perform back-to-back measurement. The appropriate soak time was considered between measurement runs. In addition to the dedicated hardware, a combination of software solutions allows data to be measured, filtered and processed automatically. This mechatronics solution moves the calibration engine data through an efficient path, increasing not just efficiency itself but data consistency, repeatability, analysis methods and robustness.

As the present and future legislations, lead to calibration processes complexity sharp increase, its efficiency must follow the same trend to maintain its viability in terms of “manpower”, costs and infrastructure. This combined automation and virtualization in vehicle for catalyst heating is able to achieve this important goal of keeping the process viable in spite of the growing complexity by means of efficiency increase.

The usage of machine learning technics to analyze the data provides a leap in terms of process efficiency. ASCMO Gaussian process regression reduces sharply the number of data points required to evaluate the best input combination as its statistical interpolation properties allows the making of precise predictions even for combinations not explicitly measured. The ability to make reliable predictions of the catalyst temperature and engine speed roughness with much less measurement runs results in great competitive advantages in terms of engineering hours reduction and more efficient

facilities usage. In the end, this translates as less cost and therefore less compromising of project budget.

It is imperative to highlight also that the automation solution brings more than project budget savings. There are gains also in terms of robustness and consistency: regardless of the geographical location where the project may be running, it is guaranteed that the specific work package will be handled with the same execution procedures and analyses criteria. This also brings repeatability to the tests, a challenge when things are done manually by human operators. Furthermore, there is also a huge time saving when the fact that for a machine, there is no interruption with measurements and processing, no weekends, no holidays and no overnights breaks.

Only the fact of performing measurements automatically in-vehicle, overnight and over the weekend creates a data collection efficiency increase up to four times when compared to the traditional, manual process. That combined with automatic data processing and virtualized, model-based calibration proposal can, in a synergic way, increase even more calibration process quality, robustness and utmost efficiency.

METHODOLOGY

ENGINE INPUT AND OUTPUT SELECTION - Engine calibration consists in parametrizing the values commanded by ECU to actuators in terms of the input sensors values. The main actuators from an Otto engine are fuel injector, spark plug and throttle valve. To optimize catalyst heating calibration besides, the main actuators there is also the camshaft control that allows the change of engine air charge strategy for specific conditions.

However, for this proof of concept, the injector was the chosen actuator, where for a Direct Injection system through calibration is possible to control injection time, mass, angle, split mode and injection pressure by the high fuel pump pressure. Among those options three inputs were swaped to analyze the impact into catalyst heating behavior: Beginning of injection angle (SOI), end of injection angle (EOI) and split mass factor as described at Figure 1 below.

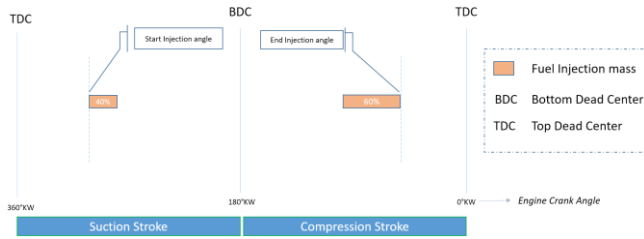


Figure 1. Injection inputs calibrations changed over the study.

Source: The Authors

Due to analysis purposes, the injection mode has been fixed into two fuel injections per cycle, one at suction stroke and the other one at compression stroke, sweeping then its angles. The split of fuel mass factor between them was the third input used for the analysis.

Afterwards, the criteria for output analysis and evaluation was based on the trade-off from two main variables: catalyst temperature and engine speed stability. By combining both outputs it is possible to infer emissions behavior: as soon as the catalyst lights off, fewer will be the emissions at tail pipe and as long as the idle engine stability is smooth, idle quality requirements will be achieved.

The steps illustrated at Figure 2 defines the entire process. Calibrator did manual steps due to physical knowledge of engine behavior for DoE planning and model analysis in order to reach reliable model.

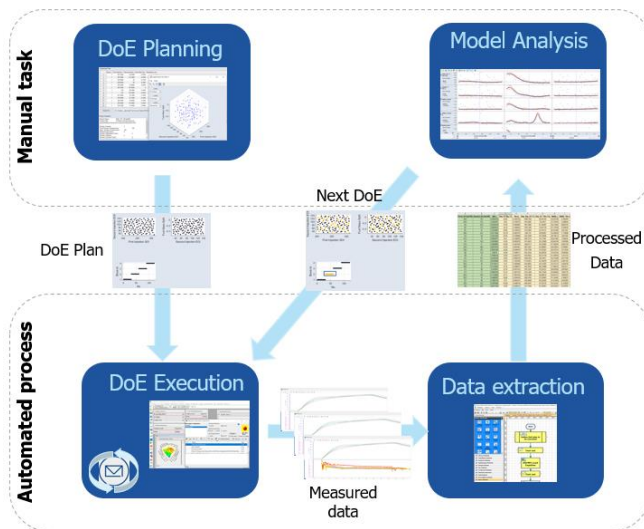


Figure 2. Overview of implemented methodology.

Source: The authors.

The automated process executed DoE in blocks, each one considered the entire range of input variables, in order to make possible generation of preliminary models. Depending on the model quality, INCA FLOW executed a new DoE to increase the amount of data.

DOE PLANNING – Catalyst heat behavior results from many different combinations: fuel mass, spark advance, engine speed are some examples. As explained in the previous sections, the DoE considered Start of First Injection, End of Second Injection and Fuel Mass split, in the ranges described in Table 1, referenced from Top Dead Center (TDC).

Table 1. Model Input ranges.

Variable	Minimum	Maximum
First Injection (SOI)	200°	310°
Second Injection (EOI)	30°	140°
Fuel Mass Split	25%	85%

For good accuracy, ASCMO indicates that for models with three inputs, at least 60 measured points were required. For better results and validation purposes, 120 points were defined, divided equally in four blocks. This approach allowed user to create models in end of each block execution.

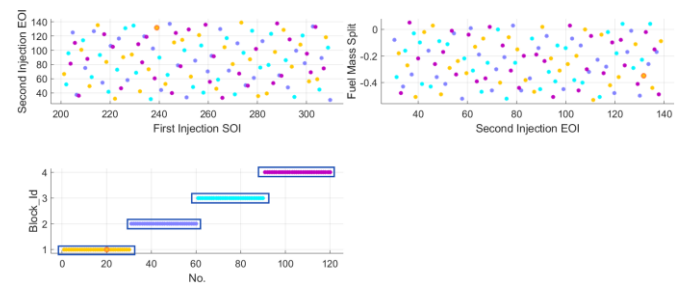


Figure 3. DoE plan.

Source: The Authors.

Figure 3 shows that each Block ID covered the entire range for Start of Injection, Second Injection and Fuel mass Split, as expected.

ROBOT – This interface simulates the driver, manipulating pedals and buttons. Compounded by electric motors commanded by PWM using INCA FLOW through ETAS ES930, multi I/O module the robot can be implemented in any vehicle. It is illustrated at Figure 4.

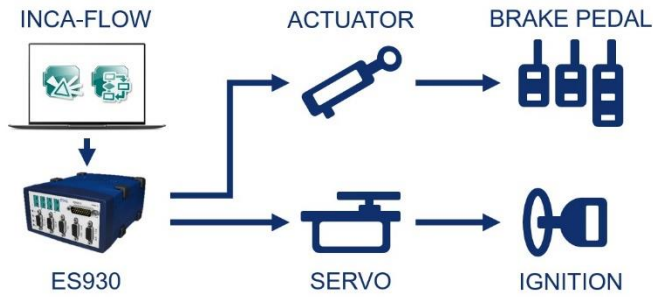


Figure 4. Robot system.
Source: The authors

A major benefit in this approach is to use the same buttons, pedals and keys designed for the Driver, without any electrical changes in the vehicle electrical system or wiring harness.

AUTOMATION TOOL – Traditional methods of measurement and calibration of ECU functions are based on manual measurement, calibration and analysis from the calibration engineer. Depending on the test scenario, environment and test conditions leads to complex criteria with hard reproduction and repetition. To optimize the process, automation methods are an alternative to monitoring all tasks and actions of a test, saving labor time and optimizing resources. One of major challenge for automotive test automation is get ECU data from ECU automotive protocols to take decision.

In order to tackle this challenge, INCA FLOW automation tool was used due to the following features:

- Measure and calibrate ECU data from INCA;
- Read DoE set point ;
- Taking decisions based on ECU data and environment criteria;
- Manipulate electrical outputs to actuators;
- Processing data from recorded files;
- Calculate signals;
- Connection to 3rd party software, like Excel or Outlook.

The program tasks are designed using a simple flowchart and block diagram concept as seen on Figure 5.

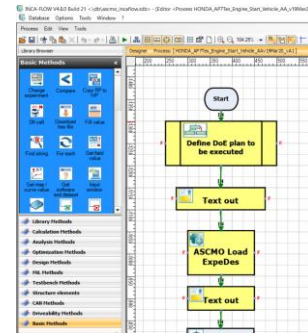


Figure 5. Inca Flow interface.
Source: The authors.

This paper used automation through INCA FLOW in two different phases: DoE execution and Data analysis.

IMPLEMENTATION OF AUTOMATED SYSTEM – The following steps implemented the automation process:

DOE Execution - To execute DoE in order to record data, the following steps were developed:

- Load DoE block;
- Monitoring oil temperature and Catalyst temperature to reach 25°C and 35°C, respectively;
- Set calibration variables according to DoE;
- Communicate the calibrator that the interaction will start by e-mail;
- Start recording;
- Start engine through robot interface;
- Wait until engine reach 50°C;
- Stop engine;
- Stop recording;
- Communicate the calibrator that the interaction ended by email;
- Wait initial criteria's to execute next DoE scenario.

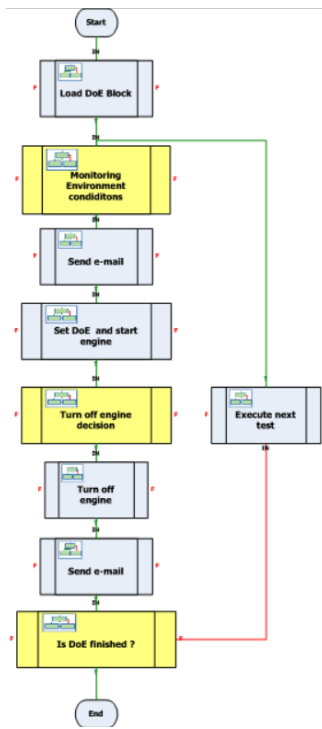


Figure 6. Developed flowchart.
Source: The authors.

Yellow blocks illustrate decisions while gray blocks defined actions by automation process.

For each interaction, the automation tool sent e-mail before and after the engine start with a test summary. It kept the calibrator informed about the status of DoE execution.

Automated system measured 120 scenarios, used to create and validate the mathematical model.

After each DoE block execution, the automation loaded the recorded data and processed the information, finding the right interval to calculate outputs that would be fed into ASCMO for modelling.

Measurement Analysis – The recorded data from ECU and measurement modules were based on time domain, while model outputs were based on static values. Then, for each DoE measured block, INCA FLOW loaded and analyzed the recorded files automatically to calculate the outputs used by modelling purposes. Based on the user criteria's, it found the right interval in the acquisition that corresponded to the catalyst heating strategy.

The following steps described measurement analysis and Figure 7 illustrate the flowchart:

- Create a template file;
- Verify non-analyzed recorded file (.dat) ;
- Extract the information from valid data interval, based on “ECU software bit's”;
- Treat the information on the variables (calculations), extracting model output variables;
- Write and organize the treated information in the template file created;
- Configure the template to the next line;
- Go back to the verify non-analyzed recorded files until all files has been analyzed.

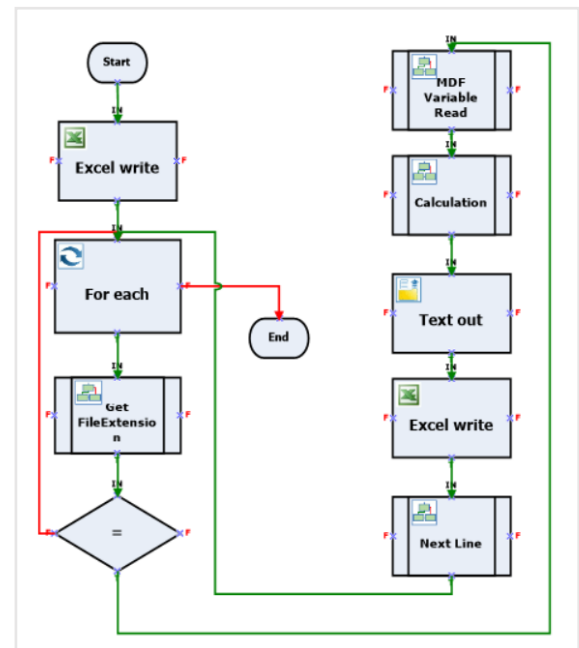


Figure 7. Measurement analysis INCA FLOW process.
Source: The Authors.

The result was a table (Figure 8) that correlate Inputs points from DoE and their respective outputs, calculated based on the recorded data files. Therefore the data information could be easily imported and interpreted at modelling phase.

First Inj. SOI (Grad KW)	Second Inj. EOI (Grad KW)	Split (-)	Temp. CAT Beg. (°C)	Time (s)	Integ. Temp. CAT (°C)
259,5	70,5	0.350255758	82,843	53.59921302	14089,75025
259,5	70,5	0.35014074	95,26	52.19959723	14325,36081
204	124,5	0.633994106	97,495	53.99848667	13901,93662
207	37,5	0.460204519	90,154	41.49989442	13250,15995
261	93	0.755330961	89,995	56.19961924	14153,95925
289,5	64,5	0.612112644	102,749	45.69873074	13840,1332
234	120	0.335214878	89,799	55.19998999	13567,22029
247,5	78	0.828567977	80,34	46.19992293	12987,37413
303	133,5	0.550529661	93,493	56.89939376	14743,94633

Figure 8. Analysis from recorded files.
Source: The Authors.

To extract the data from measured files, automated system loaded the recorded file and fulfilled internal vectors with recorded time and signals values.

Correlating the logic based in “ECU software bit’s”, automation found the properly interval at the vectors. It corresponded to the catalyst heating period.

With the relevant data, the following calculations were done in order to calculate the properly information for the Gaussian models:

- Inputs variables “First Injection”, “Second Injection” and “Split” needed to be constants during the “Catalyst Heating” period. To ensure that, a “Mean Value Calculations” was executed to each variable.;
- In order to evaluate the data recorded in variation aspect a “Standard Deviation” was calculated by the variable “Split”;
- An off-set in the time variable was made so it’s represented the catalyst heating period. The heating initial time was subtract from the general time.

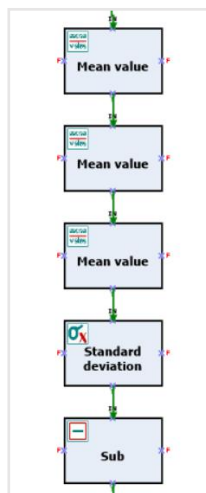


Figure 9. Inca flow “input’s” calculation.
Source: The Authors.

- To represent the catalyst temperature as an accumulated an integration of catalyst temperature and the maximum value were extracted;
- The catalyst heat time division performed the time correlation;
- In order to have more than one type of representation on the catalyst temperature was used an average calculation.

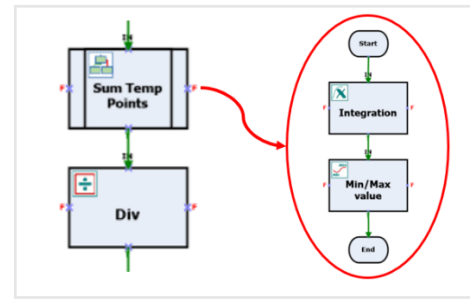


Figure 10. INCA FLOW calculation.
Source: The Authors.

MACHINE LEARNING MODELLING – Machine learning technics may be divided in 3 main branches: “Supervised Learning”, “Unsupervised Learning” and “Reinforcement Learning”:

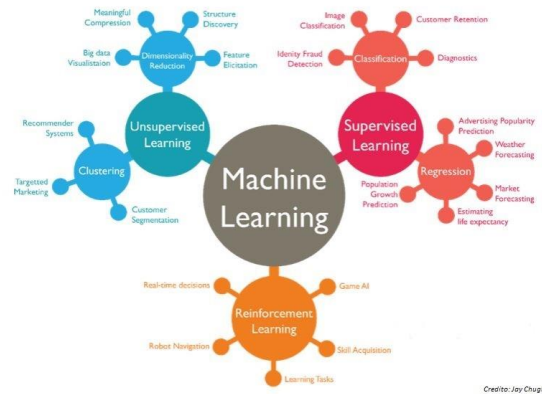


Figure 11. Types of Machine Learning Algorithms.
Source: <https://www.linkedin.com/pulse/business-intelligence-its-relationship-big-data-geekstyle>

The technic chosen for the present paper, “Gaussian Process Regression” is within the realm of “Supervised Learning”: the algorithm is fed with both the input data and the respective measured output. As both inputs and outputs are known quantities for the training data and used in training, the term “Supervised” is used.

ASCMO runs a Gaussian Process Regression Algorithm (Kriging) in order to establish relations and find underlying patterns between the model inputs and its respective outputs. The core idea is very similar to the more common polynomial regression, where the outputs are represented as a sum of a number of functions of the inputs. Nevertheless, in this case these kernel functions are of the exponential quadratic form (Gaussian Process), leading to a statistical model.

As in any machine learning technic, the first step is the definition of a training dataset (X, y) containing n measured inputs and its respective outputs:

$$\begin{aligned} X &= \{(x_i) | i = 1, \dots, n\} \\ y &= \{(y_i) | i = 1, \dots, n\} \end{aligned} \quad (1)$$

It is defined as well a test dataset, for which the trained model should make its predictions. The inputs X_* are the desired inputs for which a predictor function f_* shall be constructed:

$$X_* = \{(x_{i*}) | i = 1, \dots, n_*\} \quad (1)$$

The first assumption is to define this predictor function in the form of a Gaussian Process, a type of multivariate stochastic process for which every collection of random variables contained in it have a joint multivariate normal distribution. This kind of mathematical object is particularly suitable for the application as it is completely defined by its second order statistics. In other words, it is possible to synthesize a Gaussian Process by defining only its mean function $m(x)$ and a covariance function between its elements $k(x, x')$:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x'))$$

More precisely the definitions are:

$$\begin{aligned} m(x) &= E[f(x)] \\ k(x, x') &= E[(f(x) - m(x))(f(x') - m(x')))] \end{aligned}$$

The problem of Gaussian Process Regression is now redefined as the process of defining a proper covariance function $k(x, x')$, which is done in two steps:

Firstly and without looking at the training data, general assumptions are made on the expected behavior of the input/output relation being modeled. In this paper, the classic exponential quadratic framework was chosen. More details on it may be found on Rasmussen and Williams, Chapter 2:

$$\text{cov}(f(x_p), f(x_q)) = \sigma_f^2 \exp\left(-\frac{1}{2} \frac{|x_p - x_q|^2}{L^2}\right) + \sigma_n^2 \delta_{pq}$$

This covariance function between the outputs in terms of the inputs has some degrees of freedom as it is written with 3

free “hyperparameters” σ_f^2 , L^2 and σ_n^2 . The first one, σ_f^2 , is a scale factor between the inputs and outputs. L^2 , on its turn, is called a Length Scale, representing the extent and reach of the influence of one training data point on its neighbors. Finally, as real world data naturally contains “noise” as a result of the measuring processes, this is incorporated in the model as an additive “Noise Level” σ_n^2 acting on the input.

It is clear now that “training” the model corresponds to finding the best combinations of hyperparameters by learning through the data. This is done by a marginal likelihood optimization algorithm in ASCMO. Naturally, the marginal likelihood optimization runs iteratively over a series of generated models making the model synthesizing a relevant part of the process:

Applying the parametrized covariation function elementwise over the training dataset, a covariance matrix $K(X, X)$ is created:

$$\text{cov}(y) = K(X, X) + \sigma_n^2 I,$$

The same process is repeated for the hypothetical test dataset X_* . It is possible now to define a joint distribution of training and test data. It is a Gaussian Process with mean zero and therefore governed by the covariance matrixes of training data and test data:

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right)$$

The idea here is to isolate the unknown predictor function f_* from the known quantities. This is done by conditioning it to the observed data using Bayes Theorem, resulting in the following “Posteriori” probability distribution:

$$f_* | X_*, X, y \sim \mathcal{N}(\bar{f}_*, \text{cov}(f_*))$$

The predictor function \bar{f}_* may be defined as the most probable value of the posteriori probability distribution that its average. The details on the mathematical process may be found on Rasmussen and Williams [2].

$$\bar{f}_* \triangleq E[f_* | X_*, X, y] = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} y$$

In analogous manner, it is possible to determine the confidence of the prediction by calculating its covariance.

This is a major advantage of the usage of Gaussian process modelling, as this serves as an internal model quality metrics.

$$\text{cov}(f_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*)$$

Model Creation With ASCMO – In order to generate the model, data created by automated system was fed in ASCMO considering:

Inputs:

- Start of First Injection Angle(SOI);
- End of Second Injection Angle(EOI);
- Injection Split.

Outputs:

- Final catalyst converter temperature;
- Catalyst converter average temperature;
- Engine speed stability.

Figure 112 highlights ASCMO's user interface: Inputs are listed in the lower bar while outputs are represented in left bar. The points represents measured data and the black line represents the model prediction in terms of the inputs. Cursors can be used by the user to navigate the inputs and visualize the predicted output.

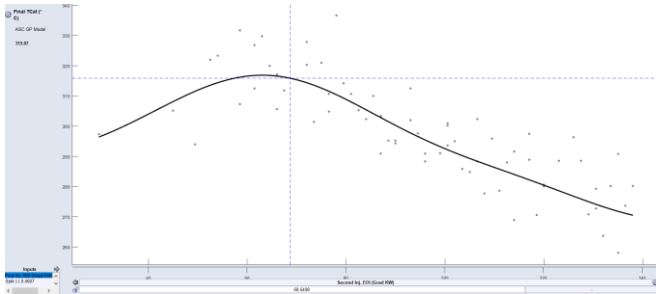


Figure 112. Gaussian model by ASCMO.

Source: The authors.

Alternately, for studying the interaction between multiple inputs, 3D view can be used, as illustrated by Figure 123. The points in 3D map represents the measured points while the surface represents the model.

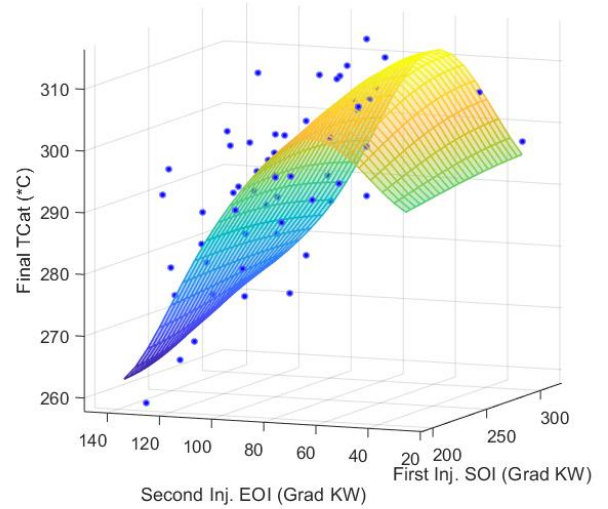


Figure 123. 3D model example.

Source: The authors.

To evaluate the model quality, some metrics were chosen. The definitions of R^2 and RMSE are given below:

$$\text{RMSE} = \sqrt{\frac{\text{SSR}}{n}} \quad R^2 = 1 - \frac{\text{SSR}}{\text{SST}}, \text{ where:}$$

$$\text{SSR} = \sum_{i=1}^n (x_{i,\text{pred}} - x_{i,\text{meas}})^2$$

$$\text{SST} = \sum_{i=1}^n (x_{i,\text{meas}} - \bar{x}_{\text{meas}})^2$$

n = number of data points (training or test) under analysis

A cross validation by “Leave-One-Out” method was executed. A instance of the model is generated trained by each point on the dataset except one. A prediction for this point “left out” is then made. The process is repeated successively for all the training data. The results are then plotted on a diagram with the predicted value on bottom axes and the real measured value on left axes. Finally, RMSE and R^2 may be applied on this data to assess model quality. An example is shown in Figure 14:

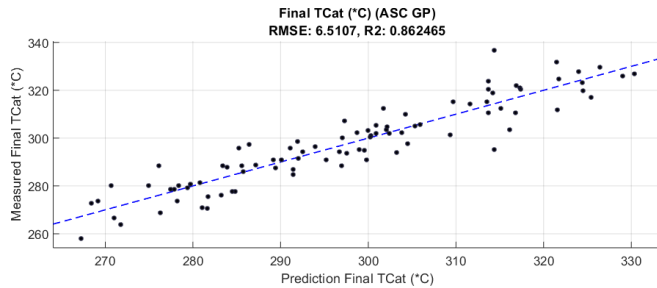


Figure 14. Leave One Out.
Source: The authors.

RESULTS AND ANALYSIS

AUTOMATED SCRIPT RESULTS – For each point of DoE, a soak time was considered in order to guarantee the same initial test conditions in each test instance. In our process, each soak cycle took around three hours. In total, automation system was executed for around 360 hours without human interference.

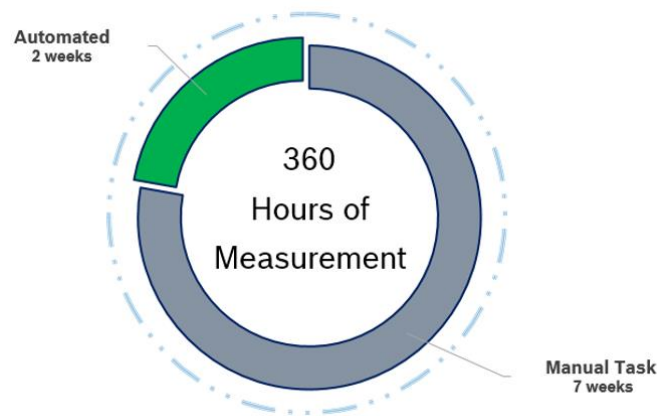


Figure 15. Efficiency gain during the measurement process.
Source: The authors.

Considering a workweek of 42 hours, an equivalent measurement process done manually would take at least seven workweeks to be completed. In comparison, once the proposed measurement system has been automatized, it took only two straight weeks to complete the 360 hours of measurements. Therefore, it is possible to affirm that for this measurement process the efficiency increased by roughly 71% when compared to a manual measurement system.

When each measurement block was finished and the data archives generated, a calibrator executed the automated data analysis script, to “extract” the relevant information for model training. The process took mere minutes to load recorded data, calculate the parameters and save relevant information in a Excel file, orders of magnitude faster than manual analysis.

MODEL ANALYSIS – Better quality models were observed from 68 measured points, close to ASCMO DoE development indication.

Model With 68 Measured Points – Firstly a Gaussian model was generated using 68 measured points from 3 of the 4 blocks. Figures 16, 17 and 18 illustrates the input and output relationship obtained.

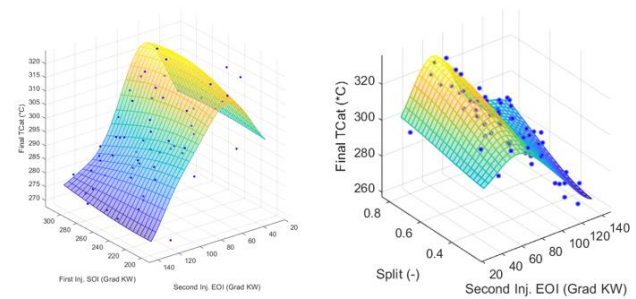


Figure 16. 3D model of Inputs and Outputs for Final TCat.
Source: The authors

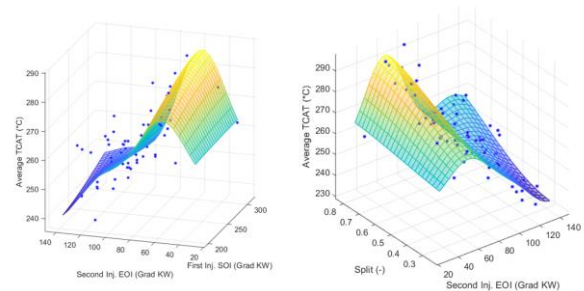


Figure 13. 3D view for Average TCat model.
Source: The authors.

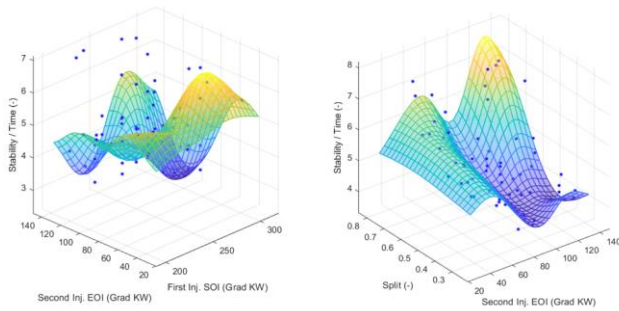


Figure 18. 3D view form Stability/Time model.
Source: The authors.

This preliminary model was validated against 24 testing data points measured from the 4th block, not used in model training. The SOI, EOI and split values were input on the model and the respective predicted outputs were compared to measured outputs. The results and its associate metrics are shown in Table 2:

Table 2. Global R^2 and RMSE for training and test data.

	Training Data		Validation Data	
	R^2	RMSE	R^2	RMSE
TCAT Average Temperature	0,80	6,60	0,90	4,92
Final TCAT	0,83	7,14	0,91	5,37
Stability/Time	0,66	0,68	0,58	0,84

For Final TCAT and Average TCAT models, training Leave One Out global and validation against Validation Data have real good fit, with R^2 over 0,8. ETAS indicates this number as a threshold for the model ability of making quantitative predictions. The Stability/Time criteria results in less reliable predictions, with R^2 between 0,6 and 0,7. Nevertheless, this range indicates the consistence for making qualitative predictions, which is precisely the idea for this specific criteria, as it is more important here to predict instability trends with input variation than finding its exact value. Therefore, the 68 point model was considered successful in its objectives.

With the model quality assured, the validation data points were incorporated in the training process for even better accuracy, resulting in a model with 90 points. The scatter plot from Figure 14 to Figure 15 indicates the measured data used to train the model and the observed results for outputs. The colors indicate the same regions between the scatter plots.

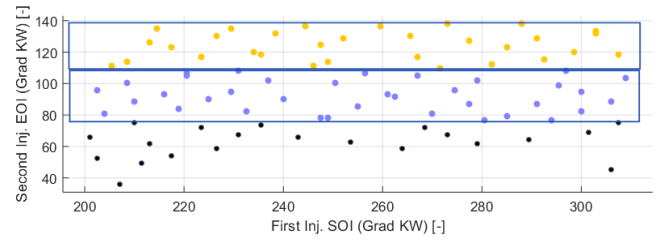


Figure 14. Scatter plot of measured data SOI x EOI.
Source: The authors.

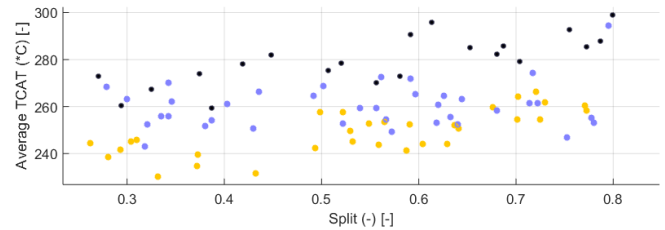


Figure 20. Scatter Plot for Average TCAT and Fuel Split.
Source: The authors.

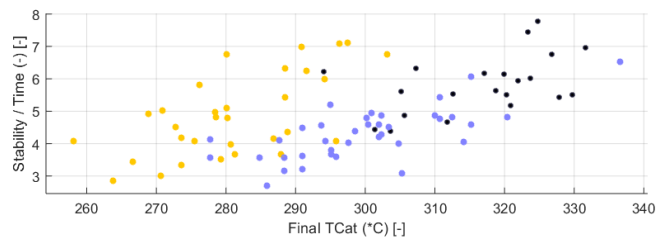


Figure 15. Scatter Plot for Final TCAT x Stability/Time.
Source: The authors.

The scatter plot indicates that catalytic temperature achieved higher values for EOI angle range between 36 and 75 degrees to a wide range of SOI angle. However, the Stability / Time has high values on this same range, indicating engine idle instability and therefore conflicting solutions to find the best calibration.

Figures 22, 23 and 24 shows the trained model by ASCMO and their correlation with the three inputs used:

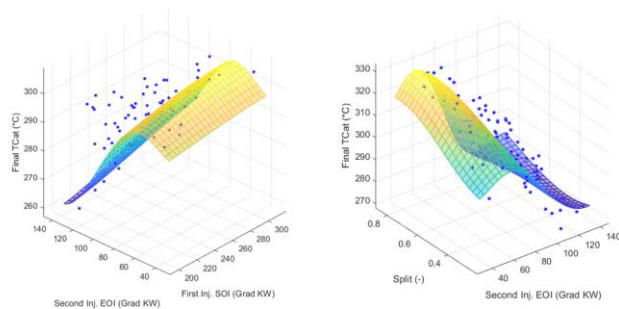


Figure 16. Final catalytic temperature model.
Source: The authors

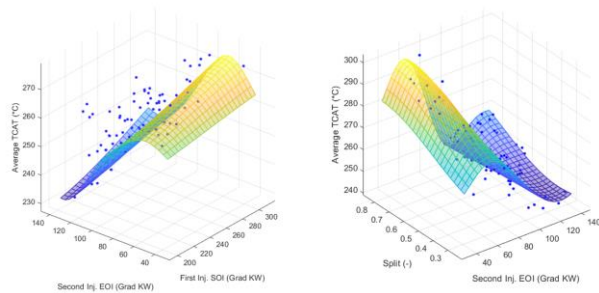


Figure 17. Average catalytic temperature model.
Source: The authors

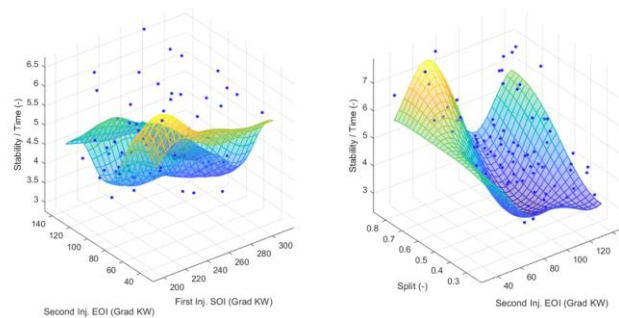


Figure 18. Stability/Time model final model.
Source: The authors

As expected, the model generated with 90 points has the same behavior as its predecessor of 68 points.

The Leave One Out analysis was also applied to the final model. Its results are shown in (Figure 19 to Figure 21).

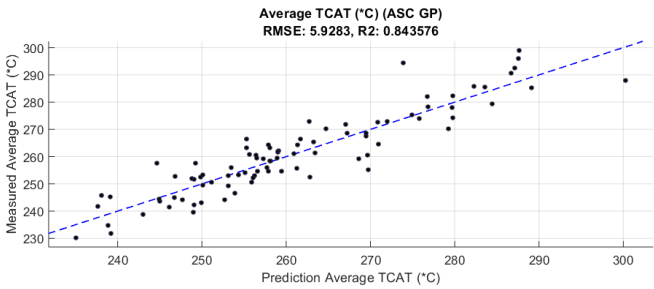


Figure 19. Predicted vs Measured Leave One Out analysis for average catalytic temperature.
Source: The Authors.

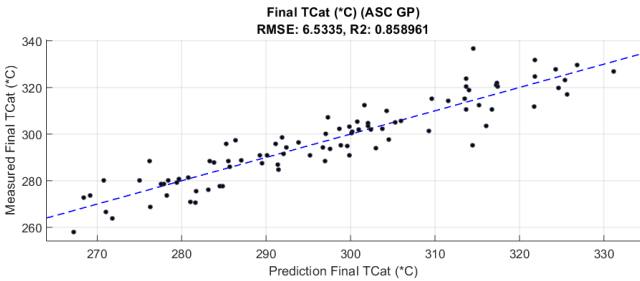


Figure 20. Predicted vs Measured Leave One Out analysis for Final Catalytic temperature.
Source: The Authors.

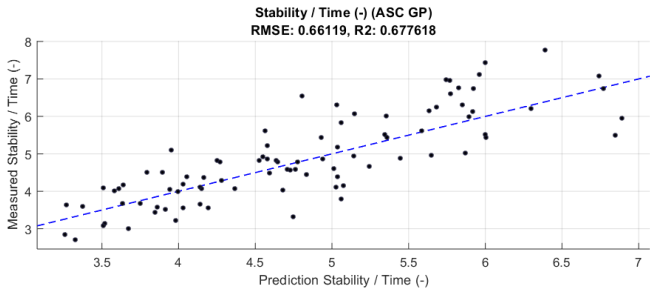


Figure 21. Predicted vs Measured Leave One Out analysis for Stability/Time model.
Source: The Authors.

Table 3 compare 90 points model with the previous the 68 point models in terms of R² and RMSE metrics. As expected, more data resulted in more model accuracy.

Table 3. Global R^2 and RMSE comparison between models.

	Model form 68points		Model form 90 points	
	R^2	RMSE	R^2	RMSE
Tcat Average Temperature	0,80	6,60	0,84	5,92
Final Tcat	0,83	7,14	0,85	6,53
Stability/Time	0,66	0,68	0,67	0,66

The main beneficial point found in the use of virtual models for catalyst heating calibration process was that with considerably few measurements it was possible to obtain a high quality model prediction for the system. This, added to the calibration engineer expertise, allows a much better understanding of the system behavior and its individual dependencies on each of the inputs. Calibration strategy opportunities are more easily seen and many complex scenarios may be tested without the need for additional experiments. The virtual model also opens up the possibility of optimization algorithms being used as a calibrator's tool.

SINGLE AND MULTI RESULT OPTIMIZATION –

After the model behavior is assessed, it is up to the calibration engineer to navigate the data and define the best possible trade-offs. ASCMO may help with that as well with its inbuilt single target and multi-result optimization packages. They were employed on our model to illustrate the additional efficiency gain possible through its usage:

Single Result – A specific input combinations is calculated to better suffice the criteria defined for each one of the outputs:

Table 4: Single Result parametrization for desired output.

Single Result Optimization		
Output	Criteria	Value
Average TCAT	Target	300°C
Final TCAT	Weak Upper Bound	300°C
Stability/Time	Weak Upper Bound	4

This method of optimization works better when the outputs are limited to a narrow value range. For future, fully automated analysis, that require more deterministic steps, this method is more suitable than a multi result, for example.

Multi Result – In this methodology of optimization, ASCMO finds not one, but all the optimal solutions in terms of the inputs combination. They are then presented to the user

as a series of “Pareto Frontiers”. The user may then highlight each solution and see its impact on the outputs, defining manually the desired calibration tradeoff.

This technic is very efficient when dealing with big models comprised of many inputs and desired output targets and constrains. Its holistic presentation of the variable interactions and the assurance that all the displayed/proposed solutions are "Pareto optimal" enables the quick visualization of the tradeoffs, its impacts and, finally, the obtention of the desired calibration dataset.

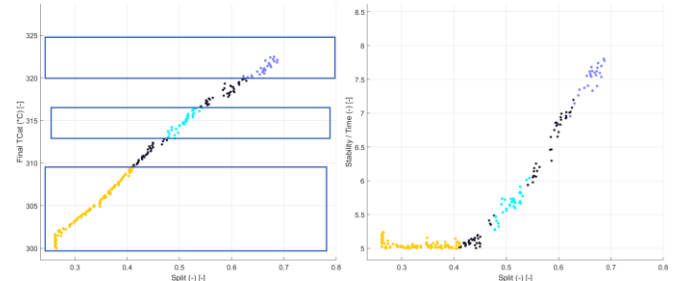


Figure 22. Fuel mass split factor input, Catalyst Temp and Engine Stability Outputs- Pareto output graphs with area selection.

Source: The Authors

For future implementation, calibration proposal and report are expected to be generated automatically as well, the diagram illustrated in Figure 239 shows how the automated system it is supposed to works.

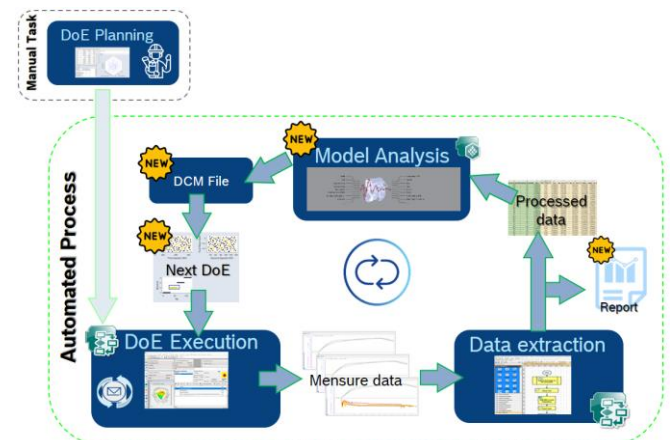


Figure 239. System overview with expected automated process included

Source: The Authors.

CONCLUSION

The proposed combinations of solutions has shown clearly a promising path to increase in-vehicle calibration efficiency. This area, particularly, lacks integrated solutions, as most of the standard calibration automation tools are aimed at engine dyno. The process efficiency was increased through the entire chain: data measurement, processing and analysis, data based model virtualization and subsequent optimization.

Combining automation process and mechatronic systems, tests could be executed uninterruptedly for many days or even weeks increasing prototype usage. Also, data analysis could be done automatically, saving much engineering hours.

By using virtualization within calibration tasks, more scenarios could be foreseen and then analyzed even with a reduced number of tests. Gaussian process modelling was applied successfully to the purposed system. As a benefit, general behavior of the system could be visualized and optimization scenarios were calculated. Both aspects are relevant considering tighter emissions and new technology implementations.

The next step to this project will be the integration of automated calibration proposals and automatic compiled data report generation.

REFERENCES

- [1] André Pelisser, Alexandre Kawamoto, Wilhelm Vatanabe, Victor Namba. “A Gaussian process model for tires in combined slip case” – SIMEA 2019. São Paulo.
- [2] C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006, (Available at <http://www.gaussianprocess.org/gpml/chapters/RW.pdf>)
- [3] ETAS GmbH. INCA-FLOW Tutorial III - Measurement Data Analysis, 2013.
- [4] ETAS GmbH. ETAS ASCMO Static V5.2 - User's Guide, 2018.